Deep Invariant Feature Learning for Remote Sensing Scene Classification

Shidong Wang

School of Computing Sciences

University of East Anglia

A thesis is submitted for the degree of

Doctor of Philosophy

January 2021

I dedicate this thesis to my beloved parents, my lovely wife and my son.

Acknowledgements

This thesis is the culmination of my Ph.D. journey, accompanied by encouragement, hardship, trust and frustration. Although only my name appears on the cover of this dissertation, I realise that many people, including my supervisors, family members, friends, colleagues and examiners have contributed to accomplish this challenging task.

At this moment of achievement, I am greatly indebted to my research guide, Prof. Ling Shao, who accepted me as his Ph.D. student and provided me with his guidance and care. Under his guidance, I successfully overcame many difficulties and learned a lot. His enthusiasm and unflinching conviction in research have always inspired me to do more. For all these, I will be truly indebted to him throughout my lifetime.

My sincere appreciation goes to all the staffs of my supervisor team, the PGR administration team and the IT support team, including Dr. Sarah Taylor, Dr. Edwin Ren, Prof. Gerard Parr, Prof. Andy Day, Dr. Katharina Huber and Matthew Ladd, for their advices, encouragement, support, requisite facilities and resources in my Ph.D. research.

I am extremely thankful to my thesis examiners, Dr. Michal Mackiewicz and Prof. Shengfeng Qin, for their time to carefully read my thesis and give me the valuable suggestions, which can significantly improve the quality of the thesis. Special thanks to Professor Richard Harvey for being the chair of my viva and encouraging me to stay relaxed during the oral examination.

My earnest thanks to Dr. Yu Guan, the leader of machine learning at the Open Lab of Newcastle University, U.K., for providing an internship opportunity to visit his team during my third year of Ph.D. and allowing me to access all research

facilities. I am also very grateful that he offered me my first research assistant job, which greatly broadened my research direction and horizons.

I greatly appreciate and acknowledge the support received from my fellow colleagues and friends, Dr. Li Liu, and Dr. Yang Long, for enlightening me the first glance of the research in computer vision and machine learning. Special thanks to Dr. Bingzhang Hu, Dr. Yi Zhou, Dr. Yuming Shen, Dr. Yang Liu and Dr. Jin Li, for supporting me when I encountered with difficulties. Big thanks to Dr. Haofeng Zhang and Dr. Zan Chen for their co-operation and support. There are many people I would like to acknowledge, and I hope you all going well.

I am sincerely grateful to the people who are important to me, my parents, for showing faith in me and giving me the liberty to choose what I desired. I thank you all for the selfless love and care, and salute you all for the dedication to shaping my life. I will never be able to repay the love and affection my parents confided.

Finally, I would like to thank a very special person, my wife, Dr. Tong Xin, for her unremitting efforts, firm support and understanding during my pursuit of Ph.D. degree. I attach great attention to her contribution and deeply appreciate her trust in me. I appreciate my son, Orlando, for abiding my ignorance and the patience he showed during my thesis writing. Words would never say how grateful I am to both of you.

Declaration

I hereby declare that this thesis has been composed by myself and that it has not been previously submitted to any other institution for a degree or qualification. Parts of the thesis have been published in authoritative conferences or journals. I can confirm that all the papers listed below, as the research results of my Ph.D., were written or co-authored by me-Shidong Wang.

- Z. Chen., **S. Wang.**, X. Hou. and L. Shao., "Recurrent Transformer Network for Remote Sensing Scene Categorisation", in *British Machine Vision Conference (BMVC)*, 2018. (Chapter 3)
- Y. Gu., **S. Wang.**, H. Zhang., Y. Yao., W. Yang. and L. Liu., "Clustering-driven unsupervised deep hashing for image retrieval", in *Neurocomputing*, 2019.
- H. Mao., H. Zhang., **S. Wang.**, Y. Long. and L. Yang., "A General Transductive Regularizer for Zero-Shot Learning", in *British Machine Vision Conference (BMVC)*, 2019.
- Y. Wang., H. Zhang., **S. Wang.**, Y. Long. and L Yang., "Semantic Combined Network for Zero Shot Scene Parsing", *IET Image Processing*, 2019.
- X. Hou., Z. Chen., L. Shao. and **S. Wang.**, "Revising Regularization with Linear Approximation Term for Compressive Sensing Improvement", in *Electronics Letters*, 2019.
- Z. Chen., X. Hou., L. Shao., C. Gong., X. Qian., Y. Huang. and **S. Wang.**, "Compressive Sensing Multi-layer Residual Coefficients for Image Coding", in *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2019.

- **S. Wang.**, Y. Guan. and L. Shao., "Multi-Granularity Canonical Appearance Pooling for Remote Sensing Scene Classification," in *IEEE Transactions on Image Processing (TIP)*, 2020. (Chapter 4)
- **S. Wang.**, Y. Long., Y. Guan. and L. Shao., "Covariance Feature Embedding for Remote Sensing Scene Classification," in submission of *IEEE Transactions on Geoscience and Remote Sensing (TGRS), 2020.* (Chapter 5)
- **S. Wang.**, Y. Ren., G. Parr., Y. Guan. and L. Shao., Invariant Deep Compact Covariance Pooling for Aerial Image Classification. in *IEEE Transactions on Geoscience and Remote Sensing (TGRS), 2020.* (Chapter 6)
- H. Duan., **S. Wang.** and Y. Guan., "SOFA-Net: Second-Order and First-order Attention Network for Crowd Counting," in *British Machine Vision Conference (BMVC), 2020.*

Signature: Shidong Wang

Date: January 2021

Abstract

With the advent of deep learning and the continuous innovation of neural networks, the field of computer vision has proceeded at a breakneck pace in the past few years. The two main driving factors behind the growth of computer vision are the tremendous amount of visual data generated daily and the increasing computing power, which makes deep learning-based algorithms blow conventional statistical methods on a plethora of benchmarks and even surpass humans in certain image recognition tasks. Although deep learning has achieved unprecedented success, it has apparent shortcomings, including the demand for a considerable number of well-annotated data to circumvent the problems of model over-fitting and lack of prior knowledge. However, manual labelling is an expensive and time-consuming process, and it is difficult to incorporate adequate variations of samples. These issues become more critical when deep learning is deployed to specific domains (such as the classification of remote sensing scene images), because the annotation process normally requires the participation of domain experts.

Since the emergence of remote sensing image classification, it has been one of the most active research field and has become increasingly attractive due to the rapid development of remote sensing acquisition facilities and deep learning technologies. The classification of remote sensing scene images aims to assign correct semantic labels to the given remotely sensed images by analysing the extracted discriminative features. This task is closely associated with a broad range of practical applications, such as urban planning, natural hazard detection, vegetation mapping, environmental monitoring, land use and land cover determination. However, compared with prevailing image classification tasks, large visual semantic ambiguities, nuisance variations, clutter backgrounds and limited number of training samples make the off-the-shelf deep learning frameworks perform defectively in the task of classifying remote sensing scene images.

In my thesis, I will devote myself to exploring ways to extend the incorporating capabilities of the prior knowledge of deep learning models and then alleviate the impact of aforementioned problems in remote sensing image classification. For this purpose, four deep learning models are proposed from different perspectives to effectively learn the second-order transformation-invariant features of remote sensing images. Firstly, a multi-stream recurrent transformer network (RTN (Z. Chen, Wang, Hou, & Shao, 2018) in Chapter 3) is proposed to gradually determine the discriminative regions of the input images and extract the corresponding bilinear features. The optimisation of RTN (Z. Chen et al., 2018) is constrained by the pairwise ranking objective function, which guarantees that adjacent network streams can converge in a mutually reinforcing manner. Secondly, the multi-granularity canonical appearance pooling (MG-CAP model (S. Wang, Guan, & Shao, 2020) in Chapter 4) is designed to autonomously learn the covariance features corresponding to the hierarchical ontology structures implicit in the datasets, with the provision of effective methods to support the calculations of the square-root and logarithmic gradients of the covariance matrix on TensorFlow-GPU. Thirdly, the covariance feature embedding (CFE model in Chapter 5 (S. Wang, Long, Guan, & Shao, -)) is devised to accurately measure the distances of vectorised high-dimensional covariance features by leveraging a novel low-norm cosine similarity loss function. Finally, a unified paradigm model - invariant deep compressible covariance pooling (IDCCP in Chapter 6 (S. Wang, Ren, Parr, Guan, & Shao, 2020)) is presented to boost the performance of remote sensing scene image classification with the highly compressed number of model parameters. The generalisation ability of IDCCP model is proved from the perspective of group theory and manifold optimisation. All proposed models can be well-supported by GPU acceleration and allow for training in an end-to-end manner. Extensive experiments have been conducted on publicly available remote sensing image datasets to demonstrate the great improvements of proposed algorithms in comparisons with the state-of-the-art methods.

Keywords— Computer Vision, Remote Sensing Scene Image Classification, Siamese-style Convolution Neural Networks, Second-order Statistics, Group Theory

Table of Contents

List of Figures

List of Tables

List of Abbreviations

RS ..... Remote Sensing

RSSC ..... Remote Sensing Scene Classification

EOS ..... Earth Observing System

LULC ..... Land Use and Land Cover

UAV ..... Unmanned Aerial Vehicle

SAR ..... Synthetic Aperture Radar

RTN ..... Recurrent Transformer Network

MG-CAP ..... Multi-Granularity based Canonical Appearance Pooling

CFE ..... Covariance Feature Embedding

IDCCP ..... Invariant Deep Compressible Covariance Pooling

EIG ..... Eigenvalue Decomposition Function

SVD ..... Singular Value Decomposition

SPD ..... Symmetric Positive Definite

SGD ..... Stochastic Gradient Descent

GPU ..... Graphics Processing Unit

SIFT ..... Scale-Invariant Feature Transform

SURF ..... Speed up Robust Features

HOG ..... Histogram of Gradient

LBP ..... Local Binary Pattern

ACC . . . . . Auto Colour Correlogram

BIC . . . . . Border/Interior pixel Classification

BoVW . . . . . Bag-of-Visual-Words

VLAD . . . . . Vector of Locally Aggregated Descriptors

FV . . . . . Fisher Vector

PCA . . . . . Principal Component Analysis

SVM . . . . . Support Vector Machine

KNN . . . . . K-Nearest Neighbour

NN . . . . . Neural Network

ANN . . . . . Artificial Neural Network

CNN . . . . . Convolutional Neural Network

GAP . . . . . Global Average Pooling

GMP . . . . . Global Max Pooling

FC . . . . . Fully-connected Layer

1 | Introduction

1.1 Research Background & Motivation

Over the past decade, the accelerated evolution of Earth observation system (EBS) has rendered the value, variety and volume of remote sensing (RS) images to expand at a staggering rate. As reported in (McCabe et al., 2017), an advanced satellite can gather terabytes data on a daily basis and it is easy to acquire petabytes data over its regular lifetime. Apart from the tremendous increase in the quantity of RS images, the use of advanced Earth observation sensors can additionally bring the quality of RS images to an unprecedented level. As the spatial resolution increases from low to high, the relationship between pixels and image objects changes accordingly. Specifically, in the early years, pixel-level (i.e., per-pixel and sub-pixel) analysis methods are widely employed in low-resolution satellite images where image objects were significantly smaller than pixels or remained at a similar level (Blaschke, 2010). However, these techniques are inefficient when examining high-resolution satellite images. In particular, they need to regionalise pixels into pixel groups so that distinguishable contextual information can be captured at the object-level. The object-level delineation of satellite imagery has dominated the classification task for a long time, but it rarely contains semantics. Consequently, the semantic-level remote sensing scene classification (RSSC) (Cheng, Han, & Lu, 2017) was proposed to mitigate the impact of lack of semantic information and it has become one of the most active studies in the field of understanding remote sensing images.

With the rapidly increasing in the number of diverse RS images, methods to fully exploit such valuable data become vitally important. Effective approaches will have a profound impact on numerous applications related to remote sens-

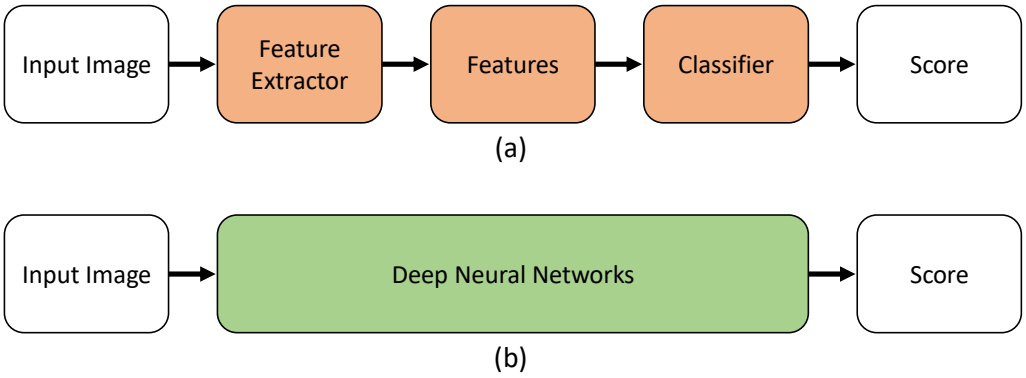


Figure 1.1: Comparison of two standard machine learning workflows in the RSSC task. (A) Traditional machine learning workflow. (B) Deep learning workflow.

ing, including land use and land cover (LULC) determination, natural hazards detection, vegetation mapping, urban planning, environment monitoring and geographic space object detection. During the past decades, machine learning algorithms have contributed highly in the advancement of classification systems because they enable the automatic analysis of massive quantities of data and improve from experience without involving extensive manpower. Due to the vast benefits and potential of machine learning, its popularity has dramatically increased among image classification and it undoubtedly becomes the first choice for solving problems in RSSC tasks. The research fields of machine learning algorithms in RSSC tasks can be roughly classified into two categories: traditional machine learning algorithms and deep learning algorithms. In Figure 1.1, two standard workflows of machine learning algorithms are presented. It can be clearly seen that the goal of these two algorithms is to be consistent, that is, to return an accurate classification confidence score for a given input image. The disparity between them lies in extracting feature representations and training classifiers.

Traditional machine learning methods have been profoundly explored and applied to various scenarios related to remote sensing image processing. In traditional machine learning techniques, most applicable characteristics need to be determined

by domain experts in order to reduce the complexity of the data and make the pattern more effective for the learning algorithm. As shown in Figure 1.1 (a), traditional machine learning techniques decompose the problem statement into several parts to be solved, and then merge the results in the final stage. Commonly utilised feature extraction methods include but not limited to scale-invariant feature transform (SIFT) (Lowe, 2004), speed up robust features (SURF) (Bay, Tuytelaars, & Van Gool, 2006), histogram of gradient (HOG) (Dalal & Triggs, 2005), local binary pattern (LBP) (Ahonen, Hadid, & Pietikainen, 2006), auto-colour correlogram (ACC) (J. Huang, Kumar, Mitra, Zhu, & Zabih, 1997), border/interior pixel classification (BIC) (Stehling, Nascimento, & Falcão, 2002), colour histogram (Swain & Ballard, 1991), GIST descriptor (Oliva & Torralba, 2001). These low-level characteristics are occasionally transformed into mid-level features (e.g., bag-of-visual-words (BoVW) (Csurka, Dance, Fan, Willamowski, & Bray, 2004), vector of locally aggregated descriptors (VLAD) (Jegou et al., 2011) and fisher vector (FV) (Sánchez, Perronnin, Mensink, & Verbeek, 2013)) with a certain degree of semantic information by clustering methods such as principal component analysis (PCA) (Jolliffe, 2011) and K-means (MacQueen et al., 1967). The received features can be trained with one or more proper classifiers to predict the classification results. Since the considerable amount of manual intervention involved in feature extraction and classifier training, the quality of the prediction results largely depends on human prior knowledge and experience.

Unlike traditional machine learning, where the workflow is broken down into separate components, deep learning techniques tend to solve problems end-to-end as shown in Figure 1.1 (b). This exceedingly eliminates the need for domain expertise and the complicated process of feature selection. Furthermore, the reasoning time of deep learning algorithms is much less when comparing with traditional machine learning techniques. Due to the large number of parameters,

deep learning algorithms take a long time to train, but this process can be accomplished offline in a reasonable time with high-end infrastructures. In addition, without needing to understand feature introspection, deep learning algorithms bring tremendous benefits in terms of accuracy and test time, notably eclipsing traditional machine learning and even beyond humans. However, apart from the supremacy of deep learning algorithms in terms of accuracy, its shortcomings are also exposed, namely, the requirement for large amounts of data and the lack of interpretability. These reasons have also act as my motivation to explore the successful deployment of deep learning to solve the challenges of remote sensing image classification tasks. More detailed analysis of the challenges in the RSSC tasks will be presented in the next subsection.

1.2 Challenges

The latest development of remote sensing technology has led to the accumulation of very high spatial resolution images (e.g., about 1-4 m/pixel), which takes out remote sensing image characteristics to a new level of illustrating the geometry structure and texture peculiarities in a more distinct way. The increasing spatial resolution of aerial images not only allows the peculiarities of the image to be depicted in a smaller space, but also makes classification more ambiguous and challenging. For a more intuitive understanding of the RSSC task, I randomly selected two images from each category of the experimental datasets for display. The collected samples are diverse in weather, seasons, lighting conditions and imaging conditions, which gives rise to extremely challenges to the RSSC task. For the sake of clarity, we summarise the challenges of the RSSC task into the following aspects.

Challenge 1 — Visual-semantic discrepancy: The main reason for aris-

ing the discrepancy problem is that the pixel-level feature representation lacks high-level semantic information as the corresponding label. Specifically, remote sensing scene images usually cover a large geographic area, in which contains a variety of unstructured information and a complicated arrangement of multiple objects (existence or coexistence), therefore detailed annotations are required as supervision information.

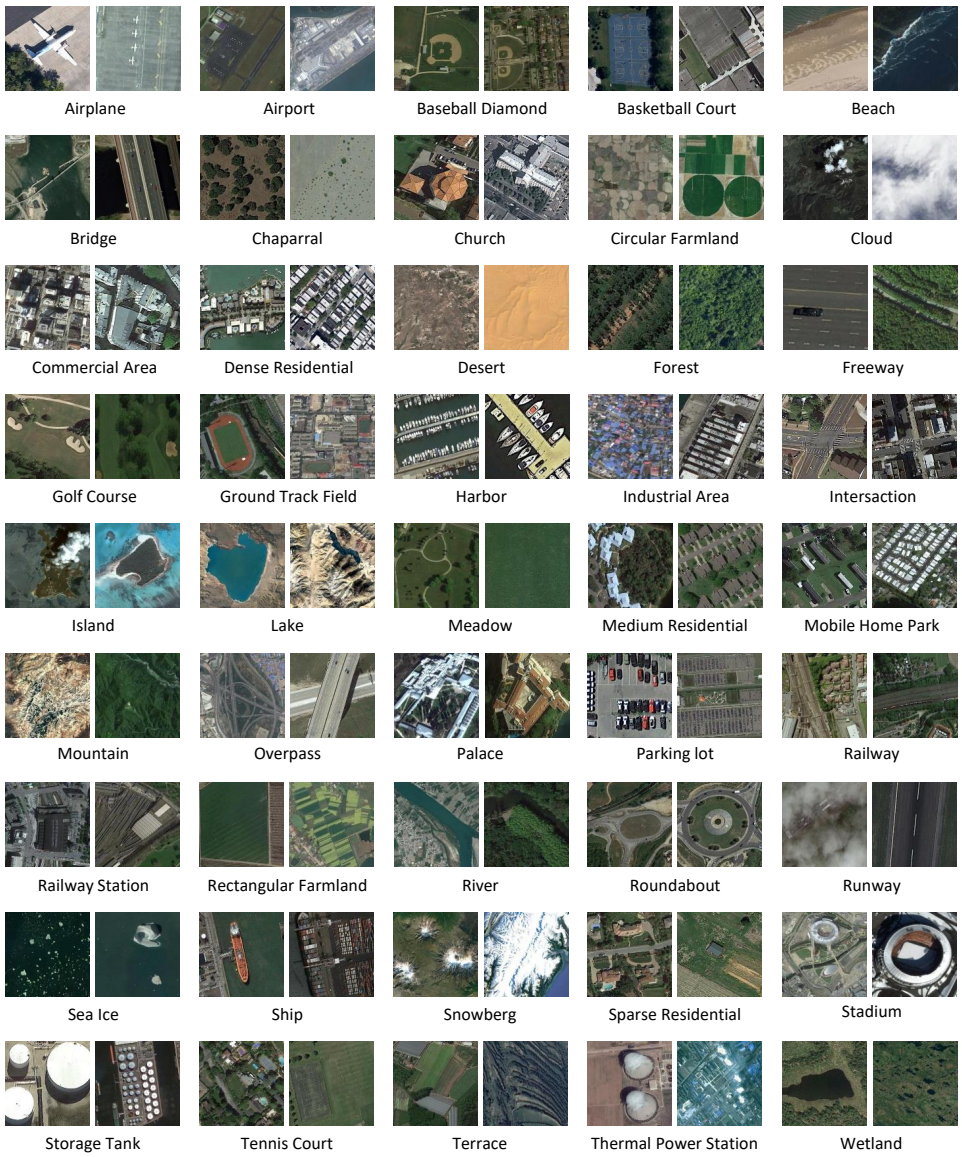


Figure 1.2: Example images from NWPU-RESISC45 dataset.

Challenge 2 — Nuisance variations: Variation has always been a common problem in large-scale datasets, but it is especially apparent in RS datasets. The reason is that remote sensing images have abundant changes in translation, rotation, scaling, viewpoint, object appearance, spatial resolution, lighting and occlusion, etc (Example images can be found in Figure 1.2, 1.3, 1.4 and 1.5). From a computer vision perspective, these disturbing variations can be summarised as intra-class diversity and inter-class similarity. Concretely, the intra-class variations are mainly caused by affine transformations or appearance changes of samples in the same category, whereas the inter-class variations are produced by subtle visual discriminations between different categories.

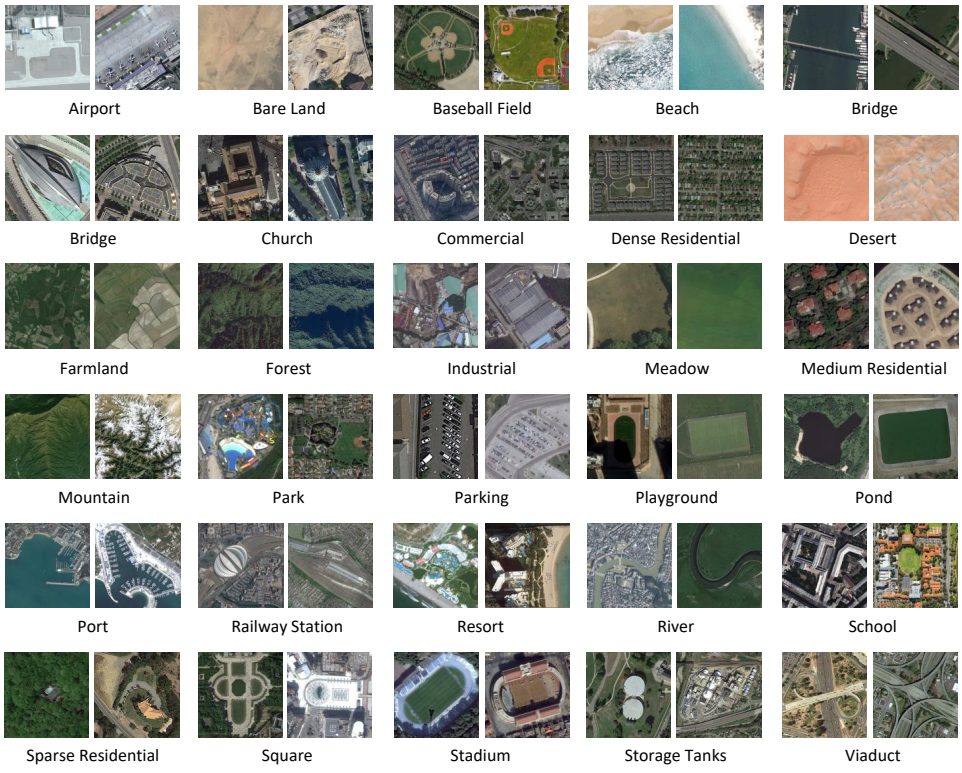


Figure 1.3: Example images from AID dataset.

Challenge 3 — Clutter background: Due to the non-ideal imaging environment, remote sensing images are usually contaminated by natural clutter.

The presence of noise and clutter inevitably degrades the quality of image, especially weakens the detailed structure of region of interest. Therefore, it is difficult to find particular objects or regions that can be used to represent the semantics of the image.

Figure 1.4: Example images from UC Merced Land-Use dataset.

Challenge 4 — Overfitting: High-quality, well-annotated satellite images are expensive to acquire. Taking an experimental scenario of NWPU-RESISC45 dataset as an example (Cheng, Han, & Lu, 2017), it is likely to cause overfitting when there are only 3,150 images available for training the deep learning model.

Challenge 5 — Unsatisfactory Performance: Existing methods can only achieve the promising results with using a high proportion of training samples, but the performance on relatively challenging datasets (e.g., (Cheng, Han, & Lu, 2017; Xia, Hu, et al., 2017)) is dramatically decreased, far from reaching the ideal level.

Figure 1.5: Example images from Optimal-31 dataset.

1.3 Contributions & Thesis Outline

In my theis, I devoted myself to discovering effective second-order features that are discriminative and transformation-invariant to nuisance variations, meanwhile, regarded it as the core that can be extended in multiple ways to solve the aforementioned challenges. I also tried to gradually optimise proposed algorithms by reducing the model complexity and the number of model parameters, while ensuring that the classification accuracy is retained at the state-of-the-art level. The remainder of my thesis is organised as follows:

Chapter 2 — Literature Review: This chapter contains a comprehensive review of previous works. As the spatial resolution of remote sensing images shifts from low to high, the development of RS image classification tasks can be roughly grouped into three main stages in chronological order, including classic pixel and sub-pixel analysis, object-based image analysis and semantic-level scene classification. When it comes to semantic-level scene classification, the detailed description of the experimental datasets and evaluation methods will be presented first, and then examine the existing algorithms, covering methods based on handcrafted features, mid-level features and deep learning features especially focusing on the delineation of methods related to second-order deep statistical features and Siamese-style networks. However, the problems of intra-class diversity caused by various affine transformations (i.e., scaling, translation and rotation) and inter-class similarity produced by the co-occurrence of similar targets have not been specifically solved. In this thesis, four models will be shown to solve the above problems by simultaneously introducing transformations that are conducive to model classification and enhancing the discriminative ability of deep learning features.

Chapter 3 — Recurrent Transformer Network (RTN) (Z. Chen et al., 2018): The visual-semantic discrepancy caused by the mismatch between the pixel-level representation and the semantic label has always been the main problem that plagues the RSSC task. Aiming to alleviate the impact of visual-semantic discrepancies, a novel attention mechanism based on parameterised transformation is proposed, which uses the positioning network repeatedly to find multiple distinct regions of the input image from coarse to fine. In order to enhance the expression ability of features, the traditional first-order pooling CNN features are discarded and replaced by the

latest second-order pooling method. In addition, the pairwise ranking loss function is ingeniously imposed on each pair of adjacent streams in order to capture the dependence of different streams and ensure that the localising of the distinguishing parts and the multi-stream feature learning are correlated and can be mutually reinforced.

Chapter 4 — Multi-Granularity Canonical Appearance Pooling (MG-CAP) (S. Wang, Guan, & Shao, 2020): The success of RTN (Z. Chen et al., 2018) introduced in Chapter 3 proves that gradually paying attention to the discriminatory areas of the image is beneficial to improve the accuracy of the RSSC task. Namely, accurately annotating the discriminative parts of the image is also the primary factor affecting the accuracy of classification, but the detailed labeling process involves labour-intensive, subjective and time-consuming. Hence, a novel MG-CAP method is proposed to automatically learn hierarchical features that match the latent ontology structures of remote sensing datasets, and then realise the alignment of high-level semantic annotations with pixel-level feature representations. This fine-grained feature learning network is derived from gradually cropping the input image three times. For each specific granularity, the input image will derive multiple instances according to a predefined set of transformations, and then learn the features of the canonical appearance through a max-out Siamese style network. Furthermore, the Gaussian covariance matrix is employed to substitute ordinary CNN features to be flattened to enhance the ability of feature discrimination. In addition, a numerically stable method is implemented so that the normalisation of covariance matrix based on the eigenvalue decomposition function can be stably trained under the GPU, and the corresponding back-propagation can be calculated using matrix calculus.

Chapter 5 — Covariance Feature Embedding (CFE) (S. Wang et al., -

): The second-order statistical features containing favourable prior knowledge can effectively improve the model classification ability (investigated by RTN (Z. Chen et al., 2018) in Chapter 3 and MG-CAP (S. Wang, Guan, & Shao, 2020) in Chapter 4). However, the existing methods neglect the fact that the vectorised second-order statistical feature lies in a high-dimensional space, in which appropriate measurements need to be adopted. To cope with this problem, a novel Low-norm Cosine Similarity (LnCS) loss is introduced, which measures the similarity of images by penalising the angles between the vectorised second-order features and their corresponding weights in the high-dimensional embedding space. Furthermore, after obtaining the covariance matrix of the CNN feature with the greatest response to the objective function, two complementary matrix Frobenius norms will be inserted before and after the square-root normalisation of the covariance matrix to enhance the discriminative power of the feature while ensuring numerical stability during training.

Chapter 6 — Deep Invariant Compressible Covariance Pooling (IDCCP) (S. Wang, Ren, et al., 2020): The common intention of all the models mentioned above (i.e., RTN (Z. Chen et al., 2018) in Chapter 3, MG-CAP (S. Wang, Guan, & Shao, 2020) in Chapter 4 and CFE (S. Wang et al., -) in Chapter 5) is to learn discriminative and invariant second-order features, which is the key to solving nuisance variations in RS scene image categorisation. However, these models are not only troubled by vectorised high-dimensional second-order features but also lack theoretical analysis to support their success. To this end, it will first consider transforming the input image according to a finite transformation group (such as the D4 group) composed of multiple confounding orthogonal matrices. Then, a Siamese-style network is adopted to transfer the group structure to the representation

space, in which a trivial representation that is invariant under the group actions can be derived. The linear classifier trained with trivial representation will also possess the properties of invariance. To further improve the discriminative power of representation, the obtained CNN feature representations are extended to a tensor representation space, in which orthogonal constraints are imposed on the transformation matrix to effectively reduce the dimension of high-dimensional tensor features.

Chapter 7 — Conclusion and Future Work: The last chapter will summarise the contributions of this thesis, together with an outlook of my future research plan.

2 | Literature Review

The continuous vigorous development of remote sensing image analysis technology is inseparable from the evolution of RS image spatial resolution. In this section, a comprehensive literature review of RS image classification technology will be given. The methods involved will be classified into three categories, including pixel and sub-pixel analysis methods, object-based image analysis methods and semantic-level image analysis methods. Semantic-level image classification methods will be particularly emphasised and analysed because it is also the main subject of this thesis.

The emergence of traditional RS image classification techniques can be traced back to the 1980s (M. Li, Zang, Zhang, Li, & Wu, 2014). Between the 1980s and 1990s, researchers principally devoted themselves to analysing RS images from the pixel-level or sub-pixel perspectives. The assumption of the pixel-based analysis method is that each pixel typically has and corresponds to only one LULC type. However, the resolution of the images provided by early Landsats is customarily low, resulting in the target object or area in the image being significantly smaller than the pixel size, which also immediately stimulated the growth of sub-pixel-based analysis methods. The mainstream methods of pixel-level image analysis consist of supervised learning methods and unsupervised learning methods. In the supervised learning scenario, the label prediction of each pixel is performed by comparing the representation of the test image and the supervised training samples (Lillesand, Kiefer, & Chipman, 2015). Examples include methods based on the Gaussian maximum likelihood classifier (Settle & Briggs, 1987), artificial neural network (ANN) classifier (Dwivedi, Kandrika, & Ramana, 2004), KNN classifier (H. Zhu & Basir, 2005), decision tree algorithm (Friedl & Brodley, 1997; McIver & Friedl, 2002), random forest (Gislason, Benediktsson, & Sveins-

son, 2006) and SVM-based methods (C. Huang, Davis, & Townshend, 2002; Pal & Mather, 2005; Marconcini, Camps-Valls, & Bruzzone, 2009). Unsupervised learning methods predict the labels of different pixels based on examining the correlation between features and natural cluster representations instead of supervised information. Representative work includes those algorithms based on partition clustering (Rollet, Benie, Li, Wang, & Boucher, 1998), iterative clustering learning (Dhodhi, Saghri, Ahmad, & Ul-Mustafa, 1999) and agglomerative hierarchical clustering (Goncalves, Netto, Costa, & Zullo Junior, 2008). The image classification method based on sub-pixel level can not only be applied to interpret the pixel information at its own level, but it is also preferable to the pixel-level image analysis method when solving the problem of mixed pixels. Because geographic phenomena are naturally fuzzy, fuzzy classification algorithms (J. Zhang & Foody, 1998; Tang, Wang, & Myint, 2007) have received extensive attention in numerous sub-pixel analysis methods. In addition, logical classification and regression models, and spectral hybrid analysis models are also applied for remote sensing data classification (C.-C. Yang et al., 2003; Yuan, Sawaya, Loeffelholz, & Bauer, 2005), urban composition monitoring (C. Wu, 2004), and impervious surface estimation (C. Wu & Murray, 2003), respectively.

Since the late 1990s, dissatisfaction with pixel-based or sub-pixel-based image analysis methods has continued to increase, because when the entity is significantly larger than pixels, a single pixel is insufficient to capture the spatial heterogeneity of the spectral information displayed in the RS image. In 2001, Thomas raised a principal question "What's wrong with pixels" (Blaschke, 2001), and conducted thorough discussions and strong statements in subsequent works (Burnett & Blaschke, 2003; Blaschke, Burnett, & Pekkarinen, 2004; Blaschke, 2010). Since then, remote sensing image analysis has gradually developed from the pixel level to the object level, in which the object is defined as the basic entity per-

ceptually sensed from high-resolution pixel groups that have similar data values, intrinsic sizes, shapes, and geographical relationships (G. Hay, Marceau, Dube, & Bouchard, 2001). Considering the unique high-spatial and hyperspectral characteristics of RS images, (G. J. Hay & Castilla, 2008) defined a new framework named Geographic Object-Based Image Analysis (GEOBIA) and stated that it needs to incorporate some fundamental principles, including Earth-centric data, geo-object-based delineation, multi-source analysis permission, contextual and adaptable to include human semantics. For GEOBIA, the heterogeneity of fragments should be smaller than the heterogeneity of adjacent fragments, therefore, it is difficult to determine appropriate segmentation parameters for the varying size, shape and spatial locations of image segments. Typical works include the multi-resolution segmentation scheme (Benz, Hofmann, Willhauck, Lingenfelder, & Heynen, 2004; Lang & Tiede, 2007; Cheng & Han, 2016; Feitosa, da Costa, Mota, & Feijó, 2011), variance map-based methods (Kim, Madden, & Warner, 2008; Draguţ, Tiede, & Levick, 2010) and proper OBIA assessment-based methods (Blaschke, 2010; Drăguţ, Csillik, Eisank, & Tiede, 2014; Congalton & Green, 2002; Clinton et al., 2010; Congalton & Green, 2002; Lizarazo, 2014; MacLean & Congalton, 2012; Radoux, Bogaert, Fasbender, & Defourny, 2011)

2.1 Semantic-level Image Analysis Methods

In low- and medium-resolution remote sensing images, neither pixel- nor object-based image analysis methods can flawlessly handle the intricate and diverse correspondence between the object and the spectral response curve (i.e., the same object may have different spectral response curves while different objects may share the identical spectral response curve) (Gu, Wang, & Li, 2019). In recent years, this problem has become increasingly critical because the advancement of

remote sensing technology has made it possible to obtain high-resolution images that distinctly display rich detailed information of local areas and usually do not have a high spectral resolution. The image analysis method based on the scene as the sampling unit came into being, and quickly occupied the leading position of remote sensing image interpretation task.

The scene is usually composed of a set of irregularly structured objects located in a varied and complex environment, which contains rich abstract semantics. Effectively identifying different objects in remote sensing images with the scene as the sampling unit and perceiving their spatial topological distribution can eliminate interpretation ambiguities existed in remote sensing images and then serve humans to better understand remote sensing scene images. The urgent need has given birth to a series of tasks centred on remote sensing scene image understanding, which can be summarised into three categories: RS image scene classification, RS image scene retrieval and RS image object detection. These three tasks all focus on analysing the characteristics of the RS scene image, but the technologies and ultimate goals involved are completely different. Specifically, the classification of RS scene images pursues high-precision classification results by perceiving the spatial context and ontology of objects (Cheng, Han, & Lu, 2017). The objective of classification is to pursue accuracy, while image retrieval pays more attention to efficiency. For this reason, the visual features of RS scene image retrieval need to be projected into a relatively low-dimensional vector space (Xia, Tong, et al., 2017). RS image object detection not only needs to learn the context features of objects to obtain the corresponding classification results, but also requires to know the orientations and positions of the objects (Han, Zhang, Cheng, Guo, & Ren, 2014). The core of these three tasks is to learn feature representations that benefit the objective function, and the most prestigious and challenging task is the RS scene image classification based on learning discriminative

Table 2.1: The statistics of experimental datasets for remote sensing scene image classification.

Datasets	No. Images	No. Class	No. Images (Per-class)	Resolution (m)	Image Size
UC Merced Land-Use (Y. Yang & Newsam, 2010)	2,100	21	100	0.3	256×256
AID (Xia, Hu, et al., 2017)	10,000	30	220∼420	0.5-8	600×600
NWPU-RESISC45 (Cheng, Han, & Lu, 2017)	31,500	45	700	0.2-30	256×256
OPTIMAL-31 (Q. Wang, Liu, Chanussot, & Li, 2018)	1,861	31	60	-	256×256

semantic information of global features.

2.2 Datasets Description & Evaluation

In the past few years, in order to adapt to the rapid development of remote sensing image classification technology, various remote sensing image datasets have appeared one after another, especially after the advent of deep learning, the volume and diversity of remote sensing image datasets have reached unprecedented levels. Here, it will briefly review the high-resolution remote sensing image datasets widely adopted in the classification task of RS scene images. The statistics of four well-known datasets can be found in Table. 2.1.

UC Merced Land-Use Dataset (Y. Yang & Newsam, 2010) is the first remote sensing scene dataset with high spatial resolution and the category-level labels. The images are in RGB colour space and are downloaded from the United States Geological Survey (USGS) National Map. The images reveal the land use types of the US. Especially, the regions that include Birmingham, Boston, Buffalo, Columbus, Dallas, Harrisburg, Houston, Jacksonville, Las Vegas, Los Angeles, Miami, Napa, New York, Reno, San Diego, Santa Barbara, Seattle, Tampa, Tucson, and Ventura. The dataset contains 21 categories, including agricultural, airplane, base-

ball diamond, beach, buildings, chaparral, dense residential, forest, freeway, golf course, harbor, intersection, medium density residential, mobile home park, overpass, parking lot, river, runway, sparse residential, storage tanks, and tennis courts.

Aerial Image Dataset (AID) (Xia, Hu, et al., 2017) is a large-scale aerial image dataset. The images are manually collected from Google Earth imagery. Furthermore, all overhead images are chosen from different areas around the world. The majority of these images are the covers of China, United States, England, France, Germany, Japan, Italy, etc. Moreover, the images are collected at different seasons under different imaging conditions, which will make a contribution to increasing the intra-class variance. The datasets includes the following scene types: airport, bare land, baseball field, beach, bridge, center, church, commercial, dense residential, desert, farmland, forest, industrial, meadow, medium residential, mountain, park, parking, playground, pond, port, railway station, resort, river, school, sparse residential, square, stadium, storage tanks and viaduct. Note: the image of **railwaystation_7** in AID (Xia, Hu, et al., 2017) is a damaged image and will be removed during training.

NWPU-RESISC45 Dataset (Cheng, Han, & Lu, 2017) is the most challenging large-scale dataset in existence. The difficulty is mainly caused by a large number of classes and the variety of spatial resolution. Except for the 30 widely-used land-use categories, there are more 15 meaningful scene categories created and incorporated in NWPU-RESISC45 dataset. Consequently, it has 45 scene classes in total, includes airplane, airport, baseball diamond, basketball court, beach, bridge, chaparral, church, circular farmland, cloud, commercial area, dense residential, desert, forest, freeway, golf course, ground track field, harbor, industrial area, intersection, island, lake, meadow, medium residential, mobile home park, mountain, overpass, palace, parking lot, railway, railway station, rectangular farmland, river, roundabout, runway, sea ice, ship, snowberg, sparse residential, stadium,

storage tank, tennis court, terrace, thermal power station, and wetland. The sample images are extracted from Google Earth by domain-experts.

OPTIMAL-31 Dataset (Q. Wang et al., 2018) has been released in 2018. The image source comes from Google Earth. The dataset is made up of the following 31 scene categories: aeroplane, airport, baseball field, basketball court, beach, bridge, bushes, church, round farmland, business district, dense houses, desert, forest, freeway, golf field, playground, harbour, factory, crossroads, island, lake, meadow, medium houses, mobile house area, mountain, overpass, parking lot, railway, square farmland, roundabout, and runway.

The above four datasets are widely-adopted for RSSC task. Recalling the example images in the Introduction Chapter, remote sensing scene images are usually complex and contain various interference information. To comprehensively evaluate our proposed algorithms, we select four challenging high-resolution scene image datasets from all public remotely sensed datasets. Other remote sensing datasets are relatively simple since they only contain few categories or a small number of samples in each category.

Experimental Setup

For a fair comparison, the training-test ratios of the datasets are strictly in accordance with the description in original papers of the published datasets (Cheng, Han, & Lu, 2017; Q. Wang et al., 2018; Xia, Hu, et al., 2017; Y. Yang & Newsam, 2010). For NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017), two split schemes are considered: randomly split the dataset into 10% for training and 90% for testing (i.e., 70 training samples and 630 test samples per class); randomly take 20% of the dataset for training and the rest 80% is used for testing (i.e., 140 training samples and 560 test samples per class). For AID dataset (Xia, Hu, et al., 2017), there are two splitting scenarios: the proportion of training data is set to 20% and

50%, and the rest is used for testing. For the other two datasets, UC Merced land use dataset and OPTIMAL-31 dataset, the training ratio is 80% according to the original papers (Y. Yang & Newsam, 2010; Q. Wang et al., 2018), and the rest is used for testing.

Evaluation Metrics

The overall accuracy is one of the most widely used evaluation metrics in image classification tasks. Specifically, it is usually expressed as a percent, with 100% accuracy being a perfect classification where all test samples were classified correctly. The computation of overall accuracy (OA) can be obtained by following:

$$OA = \frac{TP + TN}{TP + TN + FN + FP} \tag{2.1}$$

where *TP*, *TN*, *FP* and *FN* denote the number of true samples correctly classified, the amount of negative samples correctly classified, the number of negative samples incorrectly classified and the amount of positive samples incorrectly classified, respectively.

Furthermore, the average accuracy (AA), which means averaging the prediction accuracy of every class, is also used in the evaluation sections of this thesis.

The confusion matrix, also known as the error matrix, is a specific table used to reveal the classification performance of the proposed model at the category-level. Generally, each row of the confusion matrix represents the predicted result while each column indicates the actual category, and vice versa. Therefore, it is one of the most intuitive expressions of category-level classification results in terms of correct classification and misclassification.

The K-fold cross-validation will be used in this thesis to reduce the influence of the randomness and obtain reliable results. As suggested by an approximate statistical

testing paper (Dietterich, 1998), cross-validated K times is the most powerful testing method among supervised learning methods. Since the original CFE model (S. Wang et al., -) only shows the result of cross-validation once, the algorithm will be replicated first, and then cross-validated multiple times. Note: the results of the CFE model will be reported in Chapter 3, where K = 5. Furthermore, the value of K in IDCCP model (S. Wang, Ren, et al., 2020) (Chapter 6) is set to 5 in order to be consistent with the original paper of the OPTIMAL-31 Dataset (Q. Wang et al., 2018). The K is set to 10 for both CFE model (S. Wang et al., -) in Chapter 5 and MG-CAP model (S. Wang, Guan, & Shao, 2020) in Chapter 4.

2.3 Handcrafted Feature based Methods

Handcrafted features aim to gather natural and characteristic information from the input image and have been widely adopted in tasks related to RS image understanding, including RS image classification, RS image retrieval and RS image object detection. Over time, a number of manually designed features are reported in the literature to overcome specific problems such as occlusion, scale, and illumination variations to better adapt to the various tasks being tackled by researchers. In order to explicitly compare the differences between methods based on handcrafted features, it will revisit the existing algorithms from three perspectives: local handcrafted features, global handcrafted features and texture descriptors.

Local handcrafted features mainly focus on accumulating distinct regional features and treating them as preferred classification clues. The well-known SIFT feature (Lowe, 2004) realises the invariance of re-scaling, translation and rotation to local objects by searching for similar circular areas in multiple scales and positions. For example, Yang et al. (Y. Yang & Newsam, 2010) reported that the performance of employing SIFT feature is better than the texture feature based on

Gabor filter (Jain, Ratha, & Lakshmanan, 1997) in classifying RS images. Then, (Risojević & Babić, 2012) integrated global texture feature and local SIFT feature to further improve the performance of RSSC task. The SIFT feature is robust to object rotation and scale variations, but this robustness requires a high computational cost. In this case, the surf feature (Bay et al., 2006) is proposed to approximate the image difference of Gaussian (DoG) using a box filter, which makes it faster and more stable than the SIFT feature (Lowe, 2004) when using integral images. In addition, the HOG feature (Dalal & Triggs, 2005) is another sought-after handcrafted feature because it can count occurrences of gradient orientation in a dense grid with uniformly spaced cells. Several works based on the HOG feature have been successfully applied to tasks related to RS and can be introduced as representative examples, including RS image object detection model based on multi-scale HOG features (Cheng et al., 2013), ship detection model based on HOG features (Shi, Yu, Jiang, & Li, 2013), RS image classification model based on coarse-to-fine HOG features (Cheng, Han, Guo, Liu, Bu, & Ren, 2015; Cheng, Han, Guo, & Liu, 2015).

Global handcrafted features aim to delineate the overall statistical information from the perspective of the entire image. Compared with local handcrafted features, the most prominent advantage of global handcrafted features is that the extracted features can be directly thrown into the classifier for classification. As one of the simplest global functions, the colour indexing feature only relies on dividing the colour histogram into tiles and independently calculating each histogram for the final concatenation (Swain & Ballard, 1991), which can easily accomplish the effect of translation and rotation invariance of the input image. (dos Santos & Penatti, 2010) specifically evaluated the performance of combining colour and texture descriptors in remote sensing image retrieval and classification tasks. More advanced colour-centric algorithms are successively proposed,

including Colour Auto-Correlogram (ACC) (J. Huang et al., 1997) that utilises the probability between two pixels to encode spatial colour information and the Border-Interior pixel Classification (BIC) (Stehling et al., 2002; Penatti, Valle, & Torres, 2012) that can be used to calculate the colour histograms of both border pixels and interior pixels. Although methods based on colour features have been manipulated in the field of remote sensing (H. Li, Gu, Han, & Yang, 2010; Penatti, Nogueira, & dos Santos, 2015), they are often insufficient to convey spatial information and are sensitive to the small illumination changes or quantisation errors. The GIST feature (Oliva & Torralba, 2001) is another global descriptor that represents the principal spatial structure of the image. (Z. Li & Itti, 2010) performed a representative work combining GIST features and saliency-based attention features to effectively detect the statistical features of objects in RS images.

Texture features are known for learning the similarity of low-level texture peculiarities of images. Because RS images usually cover larger homogeneous areas, such as forests, woodlands, grasslands, etc., extracting texture information happens to be a relatively simple yet effective method. Many methods based on different texture descriptors have appeared in the remote sensing field (Bhagavathy & Manjunath, 2006; Marceau, Howarth, Dubois, Gratton, et al., 1990; Musci, Feitosa, Costa, & Velloso, 2013). As one of the well-known texture descriptors, the grey-level co-occurrence matrix (GLCM) has been widely used in satellite image classification in the early days. For example, (Marceau et al., 1990) created the texture bands using GLCM and added them to the spectral bands in order to improve the classification accuracy. (Gebejes & Huertas, 2013) studied the dependence of GLCM based on different texture features such as contrast, homogeneity, dissimilarity, energy and entropy. The Gabor feature (Jain et al., 1997) is another simple method to extract features using a set of Gabor filters in different frequencies and orientations. In 2011, (Risojević, Momić, & Babić, 2011) was dedicated

to searching a appropriate kernel function for the Gabor filter and employed the SVM classifier for evaluation. Then, (W. Li & Du, 2014) employed Gabor filters for feature extraction, and then operated a classifier based on the nearest regularised subspace (NRS) for classification. Lately, (C. Chen et al., 2015) not only employed the multi-orientation Gabor filters to extract the global texture information but also adopted the local binary patterns based method to capture the local texture information. Since Ojala et al. (Ahonen et al., 2006) proposed the LBP feature, it has speedily become a popular descriptor, mainly because of its low computational complexity and the ability to encode fine details. Specifically, (Musci et al., 2013) adopted the local variance estimation combined with LBP or local phase quantisation (LPQ) descriptors to assess the classification performance of RS images, and they reported that the results achieved using LBP or LPQ descriptors can be noteworthy better than the GLCM feature. Furthermore, (C. Chen, Zhang, Su, Li, & Wang, 2016; L. Huang, Chen, Li, & Du, 2016) introduced multi-scale completed local binary patterns (CLBP), which was equipped with a kernel-based extreme learning machine to improve the land-use scene classification. More recently, (Anwer, Khan, van de Weijer, Molinier, & Laaksonen, 2018) devised TEX-Nets to encode the deep learning feature by LBP and displayed substantial improvements compared with conventional RGB networks.

Since different handcrafted features tend to collect a certain amount of explicit features in the image, this has greatly stimulated numerous researchers to attempt to merge the advantages of different handcrafted features to further improve the classification performance. Typical examples of remote sensing image classification include the method to blend local and global features at the histogram level (Q. Zhu, Zhong, Zhao, Xia, & Zhang, 2016), and the method to fuse global Gabor features and local SIFT features in a hierarchical manner (Risojević & Babić, 2012). In addition, (Zou, Li, Chen, & Du, 2016) adopted a locality-constrained

linear coding (LLC) for the K-means based visual codebook, and combined multi-scale completed local binary patterns (MS-CLBP) for the kernel collaborative representation-based classification.

2.4 Unsupervised Learning based Methods

Although methods based on handcrafted features can conclusively reveal a certain degree of dominant features in RS images, they rely heavily on human prior knowledge of the dataset. The choice of features depends on a trial-and-error strategy, which is excessively expensive to acquire and time-consuming. At this stage, even for experienced feature designers, the demands for obtaining better accuracy remain quite high. Furthermore, the manually designed features involve numerous hard-to-reproduce hyperparameters and tricky strategies. With the increase of variations in publicly available remote sensing datasets, manually designed functions become less and less reliable when capturing discriminatory information, resulting in less and less revenue. This prompted researchers to start developing methods that can learn implicit features derived from raw data.

In this context, unsupervised learning was quickly introduced to remote sensing image classification tasks and produced various variants to effectively solve the unique problems in remote sensing scene images. Unsupervised learning-based features have several advantages. First, it proves the feasibility of learning representations from raw pixels or low-level handcrafted features. Second, it allows learning features from unlabelled data, thereby greatly reducing the need for human resources and the risk of manual intervention. Third, unsupervised learning methods take into account the necessity of capturing all unknown data patterns that are beneficial to classification. In the following, it will comprehensively review several mainstream unsupervised learning methods, including

K-means (MacQueen et al., 1967), PCA (Jolliffe, 2011), Autoencoder (Hinton & Salakhutdinov, 2006) and Sparse coding (Olshausen & Field, 1997).

The PCA (Jolliffe, 2011) is one of the most fundamental unsupervised learning methods, known for its ability to effectively retain the most valuable components while deleting the least important elements of input data through dimensionality reduction. Early work mainly applied PCA or kernel PCA to learn the compact feature representation of hyperspectral remote sensing image data. After proving that PCA can effectively process remote sensing data, some variants of PCA are proposed, among which the more famous methods such as PCANet (Chan et al., 2015). In addition, (Chaib, Gu, & Yao, 2016) proposed Sparse PCA, which constructs visual dictionaries for high-resolution satellite image classification by extracting information from local features, namely, SIFT and SURF features.

The sparse coding (Olshausen & Field, 1997) normally learns the sparse representation of the input data by simultaneously optimising the L1 norm of the reconstruction loss and the sparse representation loss. Sparse coding is extremely effective in highlighting essential features and eliminating noise, which also makes it attractive in scene image classification tasks. For instance, (Cheriyadat, 2014) encoded unlabeled low-level features with a set of basic functions and then generated corresponding sparse feature representations in a sparse coding manner. (Zheng, Sun, Fu, & Wang, 2012) designed an annotation framework by using a multi-feature joint sparse coding method based on spatial relationship constraints. Sparse coding has also been widely applied to tasks related to RS image classification. (Sheng, Yang, Xu, & Sun, 2012) and (Mekhalfi, Melgani, Bazi, & Alajlan, 2015) presented multi-feature fusion methods based on sparse coding to classify scene images. (Zou et al., 2016) studied the local features based on sparse coding, and combined with the global multi-scale completed local binary patterns for RS scene classification. A method of generating sparse coding-based correlograms

for visual codewords was proposed by (Qi, Xiaochun, Baiyan, & Wu, 2016). Besides, (Han, Zhou, et al., 2014) integrated visual saliency modelling and sparse coding coefficients to improve the performance of detecting objects in RS images.

The autoencoder (Hinton & Salakhutdinov, 2006) is a powerful asymmetric artificial neural network (ANN) that allows to learn compact low-dimensional representations in an unsupervised learning manner. Zhang et al. adopted an autoencoder architecture to learn compressible features from the salient regions generated by the salient detection algorithm (F. Zhang, Du, & Zhang, 2015a). Then, (Ma, Wang, & Geng, 2016) improved the structure of the autoencoder by imposing an supplementary regularisation term on the energy function, and then combined it with the collaborative representation for hyperspectral image classification. (W. Li et al., 2016) introduced a case study of an autoencoder technology for remote sensing image classification in Africa. There was a work (Othman, Bazi, Alajlan, Alhichri, & Melgani, 2016) that combined the merits of sparse coding and autoencoder, and then proposed a novel sparse autoencoder.

The K-means algorithm (MacQueen et al., 1967) (a.k.a. K-means clustering) aims to aggregate given data points into a set of groups using appropriate similarity measures. Specifically, the K-means algorithm needs to identify the number of K centroids, and then allocate each data point to the nearest cluster while maintaining the centroid as small as possible. In this way, the algorithm can effectively remove noisy data by minimising within-cluster variances. The popularity of K-means is largely attributed to the Bag-of-visual-word (BoVW) (Csurka et al., 2004), which generates intermediate image descriptors to narrow the gap between low-level features and high-level semantics. The BoVW algorithm learns sparse vector representation by counting the occurrence of visual words in the word dictionary. The whole algorithm includes two main processes: feature encoding and codebook generation (the common method is K-means clustering). As a powerful

middle-level feature, the BoVW feature has been widely employed in RS image classification (Bahmanyar, Cui, & Datcu, 2015; S. Chen & Tian, 2014; Cheng, Li, Yao, Guo, & Wei, 2017; Y. Zhang, Sun, Wang, & Fu, 2013; J. Zhang, Li, Lu, & Cheng, 2016; L. Zhao, Tang, & Huo, 2014; L.-J. Zhao, Tang, & Huo, 2014; L. Zhao, Tang, & Huo, 2016; B. Zhao, Zhong, & Zhang, 2016; Q. Zhu et al., 2016; Zou et al., 2016; Shahriari & Bergevin, 2017). In 2010, (Y. Yang & Newsam, 2010) first tried to use the standard BoVW framework to classify land-use scene images with high spatial resolution, and the BoVW algorithm performs robustly in certain categories. Immediately thereafter, many efforts have been made to incorporate more accurate spatial and contextual information when extracting local features. These methods include a concentric circle-based rotation-invariant feature representation (L.-J. Zhao et al., 2014), a spatial relationship based pyramid feature (S. Chen & Tian, 2014) and a combined representation based on the mid-level feature of the object (J. Zhang et al., 2016). In addition, (Q. Zhu et al., 2016; Zou et al., 2016) studied the effectiveness of fusing local and global features for RS image scene classification. The main difference between the two approaches is that (Zou et al., 2016) exploited the shape-based invariant texture index to capture the global information, while (Q. Zhu et al., 2016) adopted multi-scale LBP features. In recent years, with the rapid development of deep learning, many methods have been proposed to replace traditional machine learning features with deep learning features to create bags of words to further improve the classification performance (Cheng, Li, et al., 2017).

2.5 Deep Learning based Methods

The emergence of deep learning is mainly attributed to the integration of artificial neural network (ANN) and modern machine learning technology (Hinton

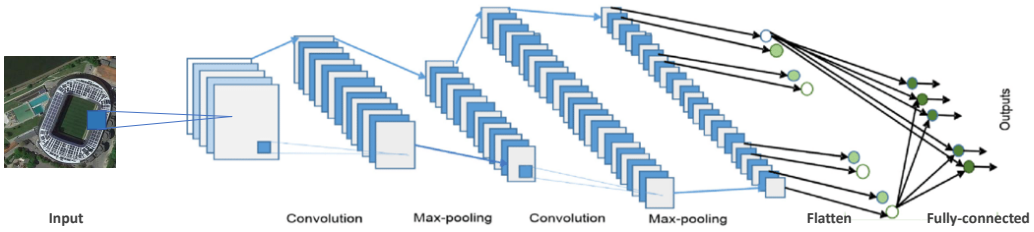


Figure 2.1: An illustration of the standard CNN structure.

& Salakhutdinov, 2006). The most significant difference between a deep learning model and other traditional machine learning models is that it can form a high-level hierarchy by superimposing multiple layers of neurons. At present, deep learning models based on multi-layer convolutional neural networks have made massive breakthroughs in the field of image recognition. Deep learning has many significant advantages, including the elimination of feature engineering requirements, the excellent large-scale dataset processing capabilities, the successful delivery of high-quality results, and etc. Especially, deep learning models will neither rely profoundly on human prior knowledge and experience like traditional machine learning algorithms, nor will they appear weak or impractical in large-scale data clustering like unsupervised learning.

2.5.1 Convolutional Neural Network (CNN)

Before discussing deep learning algorithms in remote sensing image classification tasks, it is necessary to understand the primary components and functions of deep learning. The overall architecture of CNN is analogous to the connection pattern of neurons in the human brain, in which the original input can be projected to the desired output through stacked multiple perceptrons. As shown in 2.1, the standard CNN structure consists of convolutional layers, pooling layers, fully-connected layers, output layers, and activation functions or normalisation functions embedded in the previous and current convolutional layers. Example

architectures include AlexNet (Krizhevsky, Sutskever, & Hinton, 2012), VGGNet (Simonyan & Zisserman, 2014) GoogleNet (Szegedy et al., 2015), ResNet (K. He et al., 2016) and so on. For more detailed information about deep learning and CNN architectures, I refer readers to (Goodfellow, Bengio, & Courville, 2016).

The convolutional layer is the core layer of the CNN architecture. The purpose of the convolution layer, or more precisely the convolution operator, is to extract high-level features from the input signal. In the convolution operation, the convolution kernel is applied to work on a specific area of the image in a "sliding window" manner. The convolution kernel is a weight matrix, which can be updated during back propagation. The most rare thing is that the kernel can significantly reduce the number of parameters by sharing weight operations and make it easier to learn a deeper CNN architecture. In this way, the upper convolutional layer in the CNN architecture can capture low-level features, such as colour, edge, gradient direction, etc., and as the number of stacked convolutional layers increases, high-level features can also be gradually gained. The output of the convolution operation is called the feature map, and the dimension of the feature map is determined by the padding function used. Specifically, the dimensionality of the feature map will decrease with using the valid padding function while it can also increase or remain the same as the input with using the same padding function.

The pooling layer is principally responsible for reducing the spatial size of the feature map, but it does not require a matrix with updatable weights like convolution operation. Common pooling layers include max-pooling, average pooling and random pooling. More specifically, max-pooling brings about the maximum value from the portion of the image covered by the convolution kernel. The max-pooling layer is not only beneficial to reduce the size of features, but also effectively performs noise suppression. Likewise, both average pooling and random pooling can be used to reduce the spatial size of the feature map, but random pooling is able

to return random values from the weighted feature map with a certain probability. More importantly, the pooling layer is highly robust for learning transformation invariant features, especially for the specific shift of the input data.

The fully connected layer is an effective method for learning nonlinear combinations of high-level features, which are generated by flattening the output of the convolutional layer. It is important to know that most of the parameters in the CNN framework are stored in fully-connected layers with a large number of neurons, which are composed of learnable weights and biases. Each neuron is thoroughly connected to all activated neurons in the previous layer. Since each neuron in the fully connected layer is fully connected with all activated neurons in the previous layer, the tremendous vector space contained therein not only effectively eliminates the spatial information of the feature map, but also guarantees high-precision classification results.

The aforementioned operations serve as the main procedures for mapping the high-order feature space to the vector space in the standard CNN structure. Apart from these operations, many other functions are also proposed to assist or consummate the CNN architecture. Main examples include activation functions, normalisation methods, regularisation terms, and classification functions. The activation function, especially in the case of non-linearity, avoids the problem of gradient vanishing during training by determining whether to activate the node (mapped to a specific interval). The most common types of activation functions include sigmoid, hyperbolic tangent (tanh) and rectified linear unit (ReLu) (Glorot, Bordes, & Bengio, 2011). The normalisation function is proposed to alleviate the effect of the covariance shift problem (i.e., if the input distribution changes, the behaviour of the algorithm will change), and accelerate the learning process by allowing the use of a higher learning rate. Well-known normalisation methods include batch normalisation (Ioffe & Szegedy, 2015), layer normalisation (Ba, Kiros, & Hinton,

2016), instance normalisation (Ulyanov, Vedaldi, & Lempitsky, 2016) and group normalisation (Y. Wu & He, 2018). All these feature normalisation methods perform the calculation $\hat{x}_i = \frac{x_i - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}}$ for every coefficient $\hat{x}_i$ of the input feature x, where μ_i and σ_i^2 denote the mean and variance over a set S_i of coefficients. ϵ is a small value added for numerical stability. The main difference of these normalisation methods is the value of S_i. Specifically, in batch normalisation (Ioffe & Szegedy, 2015), the μ_i and σ_i^2 are computed along the batch, height and width dimensions of the feature. The S_i is defined as a set of coefficients in the same channel as x_i. Similarly, layer normalisation (Ba et al., 2016) computes μ_i and σ_i^2 along the feature's channel, height and width, with S_i is defined as all coefficients belonging to the same input feature as x_i. Instance normalisation (Ulyanov et al., 2016) only calculates the μ_i and σ_i^2 along the feature height and width, with S_i is defined as a set of coefficients in the same input feature and in the same channel as x_i. The group normalisation (Y. Wu & He, 2018) is more advanced, which organises the feature channels into different groups and computes their μ_i and σ_i^2 along the channel, height and width dimensions of the grouped feature. The value of S_i is set to be the same as instance normalisation but subject to the constraints of the *group*. In addition, deep learning models are also affected by regularisation terms (e.g, L1 or L2 regularisation). More precisely, the regularisation function discourages the learning of more complex or flexible models, thereby avoiding the risk of overfitting. Finally, the main role of the classification function like the Softmax function is to convert the output of the fully-connected layer into a form of probability to facilitate calculation.

2.5.2 CNN-based Methods

As CNN announced its success in various large-scale visual classification tasks, it finally began to penetrate into the field of remote sensing image analysis around

2015, and has been making breakthroughs ever since (L. Zhang, Zhang, & Du, 2016; X. X. Zhu et al., 2017). Benefiting from the unprecedented feature representation capabilities of CNN, many works have deployed it to RSSC tasks with different strategies, including the use of pre-trained CNNs for feature extraction, fine-tuning the pre-trained CNNs to the target datasets, and training task-specific CNNs from scratch.

Using pre-trained CNNs to extract features from remote sensing scene images is the most prevalent and dependable way in the early stage. In 2015, Penatti et al. investigated the effectiveness of off-the-shelf CNNs in the classification of RS images and reported CNN-based features are generally superior to traditional low-level descriptors (Penatti et al., 2015). In the same year, (Hu, Xia, Hu, & Zhang, 2015) studied how to take advantage of the pre-trained CNNs as a feature extractors for high-resolution remote sensing imagery classification. (Marmanis, Datcu, Esch, & Stilla, 2015) designed a two-stage CNN framework for feature extraction and scene classification. Later, (Chaib, Liu, Gu, & Yao, 2017) examined the effectiveness and necessity of the fusion of pre-trained CNN features to improve the classification result. Cheng et al. (Cheng, Li, et al., 2017) proposed the bag-of-convolutional-feature (BoCF) as the replacement of conventional local descriptors. In (Lu, Sun, & Zheng, 2019), authors aimed to comprehensively explore semantic label information, a feature aggregation CNN (FACNN) scheme for scene classification is then introduced.

Fine-tuning the pre-trained CNNs to the target dataset should be considered first if the available dataset is insufficient to support training the CNN model from scratch. Castelluccio et al. (Castelluccio, Poggi, Sansone, & Verdoliva, 2015) thoroughly studied the use of CNN in remote sensing image scene classification, and reported that when the dataset is small, fine-tuning produces better results than full training. Subsequently, Cheng et al. (Cheng, Yang, Yao, Guo, & Han,

2018) proposed discriminative CNNs (D-CNNs) to simultaneously accomplish intra-class compactness and inter-class separability by imposing a metric learning-based regularisation term on the ordinary cross-entropy loss. (Y. Liu, Suen, Liu, & Ding, 2018) coupled the CNN feature with the hierarchical Wasserstein objective function (HW-CNN) to improve the discrimination ability of CNN. (Minetto, Segundo, & Sarkar, 2019) designed a novel framework for RSSC called Hydra, which employs CNN features in an ensemble way. A gated bidirectional network (GBN) was proposed by (Sun, Li, Zheng, & Lu, 2019) to aggregate the interdependent information of CNN features in different layers and further improve the classification accuracy of RS scene images.

Training CNN from scratch is also worth exploring, because most pre-training models are trained on relatively general large-scale datasets, so unknown noise is likely to be introduced due to different domains. For example, Zhang et al. (F. Zhang, Du, & Zhang, 2015b) proposed a gradient boosting random convolutional network (GBRCN) for RSSC by assembling different deep neural networks. Chen et al. (G. Chen et al., 2018) introduced the strategy of knowledge distillation into RS scene image classification to effectively improve the performance of lightweight CNNs. (B. Zhang, Zhang, & Wang, 2019) introduced dilated convolution and channel attention to extracting discriminative features, and a multi-dilation pooling module to further improve performance.

In addition to the methods described above, researchers are also committed to developing different deep learning models to improve the effectiveness of RSSC tasks, including the global and local feature fusion methods (Bian, Chen, Tian, & Du, 2017; Y. Yu & Liu, 2018; Zeng, Chen, Chen, & Li, 2018), transfer learning-based methods (Cheng, Ma, Zhou, Yao, & Han, 2016; Hu et al., 2015; Marmanis et al., 2015; Othman et al., 2017; Xie, He, Fang, & Plaza, 2019), data augmentation (X. Yu, Wu, Luo, & Ren, 2017), and generative model learning (Bashmal et

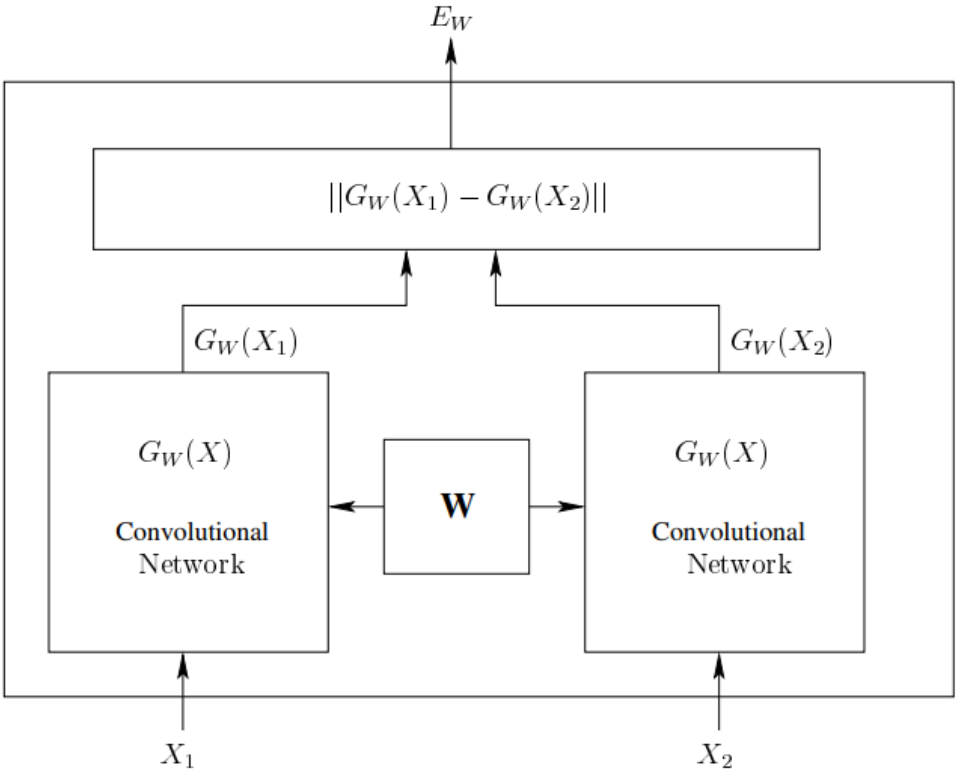


Figure 2.2: The structure of a criterion Siamese architecture, adopted from (Bromley et al., 1993). Where X_1 and X_2, $G_W(X_1)$ and $G_W(X_2)$, as well as E_W denote a pair of images, two points in the low-dimensional space that are generated by mapping of paired images, and a scalar energy function, respectively.

al., 2018).

2.5.3 Siamese Neural Network

The emergence of the Siamese neural network can be traced back to 1993 and it has been rapidly developed with the popularity of deep learning after 2005 (Bromley et al., 1993). As shown in Figure 2.2, the main idea of Siamese architecture is to seek a function that maps the input pattern to the target space so that the simple distance like Euclidean distance in the target space can approximate the "semantic" distance in the input space. Taking into account the advantages

of the Siamese network in mining data similarity metrics, this particular structure has been widely used in tasks such as image classification (Koch, Zemel, & Salakhutdinov, 2015), person re-identification (Varior, Haloi, & Wang, 2016), remote sensing image retrieval (Chaudhuri, Banerjee, & Bhattacharya, 2019) and change detection (Z. Zhang, Vosselman, Gerke, Tuia, & Yang, 2018). Furthermore, (H. He, Chen, Chen, & Li, 2018) applied the Siamese CNN for matching remote sensing images with complex background variations. In (Hughes, Schmitt, Mou, Wang, & Zhu, 2018), authors proposed a pseudo-siamese CNN architecture that allows to solve the task of identifying corresponding patches in very high-resolution optical and synthetic aperture radar (SAR) remote sensing imagery. Moreover, (Bashmal et al., 2018) and (X. Liu et al., 2019) adopted the Siamese architecture to learn invariant representations for RSSC tasks and aerial vehicle image categorisation, respectively.

2.5.4 Multi-scale & Multi-layer-based Deep Learning Methods

The intention of learning multi-scale or multi-layer features is to capture multiple different regions of a given image or multiple response features on different layers of CNN, so as to improve the generalisation ability of the model on the test data. (W. Zhao & Du, 2016) utilised CNN to extract features from multi-scale image patches, and then encoded these features with the BoVW model to form a holistic representation. (E. Li, Xia, Du, Lin, & Samat, 2017) integrated features extracted from multi-scale patches by using outputs of multiple layers of CNN structure. (Zheng, Yuan, & Lu, 2019) proposed multi-scale pooling method for the feature extraction and employed fisher vectors to generate higher-order representations. Similarly, (G. Wang, Fan, Xiang, & Pan, 2017) extracted multi-layer CNN features and encoded them through the vector of locally aggregated descriptor (VLAD). He et al. (N. He, Fang, Li, Plaza, & Plaza, 2018) proposed a

multi-layer covariance pooling framework for RSSC tasks. In addition, (L. Huang et al., 2016) and (C. Chen et al., 2016) attempted to improve the performance of RSSC by extracting multi-scale features based on LBP algorithm. (L.-J. Zhao et al., 2014) proposed a concentric circle-structured multi-scale BoVW feature model for Land-use scene classification.

2.5.5 Attention-based Deep Learning Methods

Attention mechanism is proposed to imitate the human visual system by automatically concentrating on the distinguishing parts of inputs. When only distinguishing regions are concerned, the redundant information that originally exists can be effectively eliminated, which will benefit to improve the classification results. The visual attention model based on saliency is proposed by (Itti, Koch, & Niebur, 1998), which is the predecessor of the attention map to extract salient objects in the image. For example, (F. Zhang et al., 2015a) exploited a saliency-based sampling method to extract feature from the RS images, which is effective to remove the noise information. More work on saliency-based attention methods can be found in the field of RS object detection, such as (Z. Li & Itti, 2010) and (Han, Zhou, et al., 2014). The saliency-based object detection assumes that the region of interest is salient, but neglects the fact that the saliency feature map lacks the degree of considering the importance of the salient part. This motivated researchers to seek ways to incorporate attention mechanisms into deep learning models. Xu et al. (Xu, Tao, Lu, & Zhong, 2018) proposed a novel neural network for the RSSC task by attaching two different attention mechanisms to its mask and trunk branches. Chen et al. (J. Chen et al., 2018) employed a computational visual attention model to automatically extract salient regions in unlabelled images and adopted sparse filters to learn the corresponding features. Considering the importance of features of different scales, (J. Wang, Shen, Qiao, Dai, &

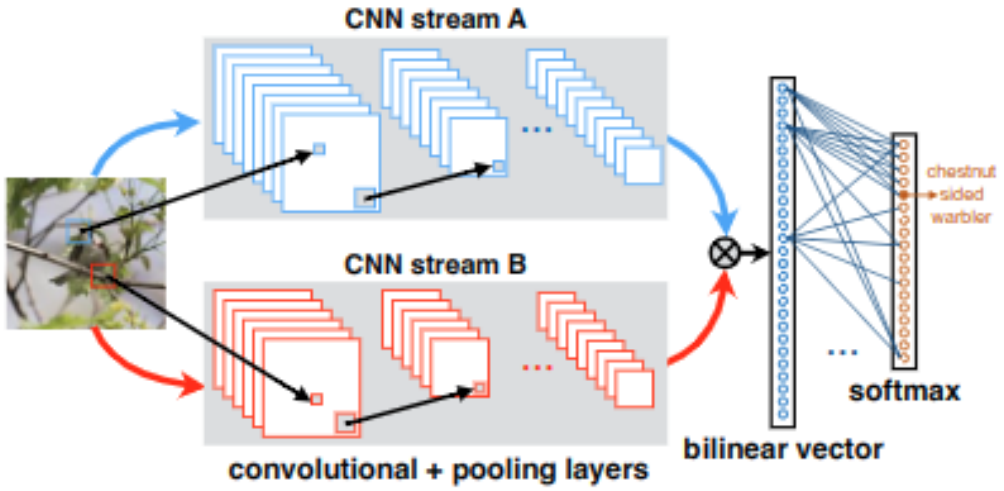


Figure 2.3: The structure of a criterion bilinear CNN for image classification, adopted from (Lin et al., 2015).

Li, 2019) proposed a class-specific attention model into a unified framework to endure these problems. (Q. Wang et al., 2018) proposed a novel end-to-end attention recurrent convolutional network (ARCNet), which uses LSTM to selectively focus on certain key areas and process them only on high-level features.

2.5.6 Second-order Statistical Feature-based Methods

Bilinear pooling (Lin et al., 2015), as one of the first end-to-end second-order pooling methods, collects the second-order statistics of local CNN features over the whole image to form a holistic representation (shown in Figure 2.3). (Ionescu, Vantzos, & Sminchisescu, 2015) presented a matrix back-propagation structure named DeepO$_2$P for Singular Value Decomposition (SVD) and Eigenvalue Decomposition (EIG) in deep learning. In 2017, Lin and Maji investigated how to improve the performance of bilinear pooling by using a combination of different normalisation methods, including matrix square root normalisation, element-wise square-root normalisation and L_2 normalisation (Lin & Maji, 2017). In (P. Li, Xie,

Wang, & Zuo, 2017), authors verified the feasibility of applying second-order features to large-scale image datasets. Acharya et al. (Acharya, Huang, Pani Paudel, & Van Gool, 2018) proposed a covariance pooling framework, which exploits the Riemannian manifold for facial expression recognition. Furthermore, various methods have been proposed to solve the high-dimensional problem of bilinear features, including Random Maclaurin method (Gao, Beijbom, Zhang, & Darrell, 2016), Tensor Sketch method (Gao et al., 2016), low-rank approximation (Kong & Fowlkes, 2017), Gaussian RBF kernel (Cui et al., 2017) and Grassmann manifold (Wei, Zhang, Gong, Zhang, & Zheng, 2018). Especially, Li et al. (P. Li, Xie, Wang, & Gao, 2018) proposed an iterative-based algorithm called iSQRT-COV, which allows the use of Newton-Schulz iterations in forward and backward propagation to speedily calculate the square root of the global covariance matrix. Due to the powerful distinguishing ability of second-order features, they have also been introduced into the field of RSSC tasks recently. (N. He et al., 2018) introduced an MSCP network that brings together multi-layer stacked covariance features used to classify RS images. Later, (N. He, Fang, Li, Plaza, & Plaza, 2019) proposed another an end-to-end learning model named skip-connected covariance (SCCov) network to further improve the performance of categorising RS scene images.

2.5.7 Research Opportunities

The aforementioned methods improve the accuracy of classification by slightly adjusting the off-the-shelf deep learning models (Cheng, Han, & Lu, 2017; Xia, Hu, et al., 2017) in the field of computer vision and applying them to the RSSC task, while ignoring the unique challenges in remote sensing images. Although the latest methods such as D-CNN (Cheng et al., 2018) and MSCP (N. He et al., 2018) have further improved the classification results by introducing regular terms as constraints or using covariance features, there is still a lack of systematic and

in-depth analysis of RS scene images. Taking into account the collection facilities (Land satellite or aerial drone) and imaging characteristics (overhead) of RS scene images, its classification will also encounter different challenges from conventional image classification tasks, including visual-semantic discrepancy, nuisance variations, clutter backgrounds and etc (More details have been summarised in the section of Introduction 1). This thesis will propose corresponding solutions to the above challenges from multiple perspectives, and finally form a model that can be regarded as a paradigm of effectively handling remote sensing classification problems. In Chapter 3, a recurrent transform network (RTN) (Z. Chen et al., 2018) will be presented to alleviate the impact of large visual-semantic problem by progressively localising multiple distinct image parts and learning corresponding bilinear feature. Then, through analysing the ontological structure of RS scene image datasets, a more effective multi-granularity canonical appearance pooling (MG-CAP) (S. Wang, Guan, & Shao, 2020) will be proposed to capture granular second-order statistical features (In Chapter 4). Vectorised second-order features will produce high-dimensional feature space, and traditional distance measurement may not adequately meet the measurement requirements. Therefore, a novel lower-norm cosine similarity loss is introduced to accurately measure the angle formed between the embedded high-dimensional features and the corresponding weights, and then further improve the discriminative ability of second-order features (will be presented in Chapter 5 as CFE model (S. Wang et al., -)). Finally, it will propose orthogonal constraints that can be used to effectively compress high-dimensional tensor features, and then analyse its feasibility from the tensor representation of group theory, and form a paradigm model that can be used to solve problems in the RSSC task (can be found in Chapter 6 named IDCCP model (S. Wang, Ren, et al., 2020)).

3 | Recurrent Transformer Network

3.1 Introduction

The remote sensing scene image exhibits the actual arrangement of an indefinite number of heterogeneous objects or regions in a specific area on the earth's surface, which is the most noticeable difference between it and the ideal scene image that usually only displays a piece of unique texture information. Because of the low resolution of early remote sensing images, low-level handcrafted features (Swain & Ballard, 1991; Oliva & Torralba, 2001; Jain et al., 1997; Lowe, 2004; Bay et al., 2006) like colour, shape, texture or their combination are capable of extracting useful information at the pixel level. With the gradual increase of image resolution, many efforts were spent on how to take advantage of unsupervised learning algorithms (e.g., K-means (MacQueen et al., 1967), PCA (Jolliffe, 2011), Autoencoder (Hinton & Salakhutdinov, 2006) and Sparse coding (Olshausen & Field, 1997)) to aggregate from low-level features to mid-level features with semantic information. Nevertheless, using unsupervised learning to aggregate mid-level descriptors from low-level features is not only difficult to completely represent the abstract information contained in image labels, but also is impractical to be applied to large-scale datasets due to the high computational cost and time-consuming of the clustering algorithm.

In recent years, with the continuous accumulation of high-resolution remote sensing data, the variations of RS scene images have become more diverse. As can be seen in Figure 3.1, remote sensing scene images not only appear magnificent diversity within the same category but also there are extremely high similarities between images of different categories due to the co-occurrence of the covered content. Furthermore, due to the dramatic changes in the scale and size of the covered

Figure 3.1: Examples from NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017). **Within-class diversity**: *Palace* (1st row), *Church* (2nd row) and *Railway station* (3rd row). **Between-class similarity**: *Railway station* versus *Stadium* versus *Church*; *Airport* versus *Railway* versus *Free way*; *Dense residential* versus *Commercial area* versus *Industrial area*; *Meadow* versus *Forest* versus *Wetland* (Please follow the order from top to bottom and left to right.)

content, it also leads to a huge semantic gap between visual features and image labels. According to taxonomy, the semantic labels of remote sensing datasets have their own taxonomic hierarchical structures, but such hierarchical information is difficult to be reflected when training the model. For example, given an image at the subordinate-level in its ontology tree, the prediction of this image may depend on the features extracted from the basic-level image, therefore it will produce a large semantic gap between the semantic label and the image content-based visual feature. Moreover, the high-risk overfitting problem of using only a small number of training samples to train neural networks cannot be underestimated.

Although advanced deep learning models have been applied to RSSC tasks and show superior performance over traditional machine learning methods (Cheng, Han, & Lu, 2017), they still have critical shortcomings. More specifically, the success of deep learning in the field of image recognition can be attributed to its unique stacked hierarchical structure, and the pooling layer plays a non-negligible

role in assuring the invariance of local translation (Goodfellow et al., 2016). However, due to the limited range of receptive field, the pooling layer cannot achieve global invariance. Data augmentation techniques (Perez & Wang, 2017; Tanner & Wong, 1987) are indeed the most straightforward way, but they rely heavily on human prior knowledge and cannot ensure that the augmented data is adequate for the test data.

In order to solve the aforementioned problems, the model should be able to learn the features of multiple discriminative regions from the input image while also discovering transformations that are beneficial for classification. Inspired by the success of the spatial transformer network (STN) (Jaderberg, Simonyan, Zisserman, et al., 2015), a recursive transformer network (RTN) is proposed to gradually find the multiple distinct regions and their canonical transformations expected by the defined objective function. Rather than using multiple independent parallel structures like the original STN (Jaderberg et al., 2015), the proposed RTN concerns more with the hidden relationships between adjacent streams. The contributions of the proposed STN can be briefly summarised as follows:

- RTN can accurately locate multiple different regions and learn robust transformation invariance features. The attention mechanism based on spatial transformation can deal with variations, while the regional features based on localising help to reduce the semantic gap between semantic labels and visual features.

- RTN guarantees to retain more discriminative information in CNN features with using bilinear pooling. Meanwhile, it can progressively discover the subtle differences presented in image regions by introducing the pairwise ranking loss function.

- Extensive experiments were conducted on three challenging RSSC datasets

and the latest accuracy was obtained. The model can be trained in an end-to-end manner using only category-level labels.

3.2 Method

In this section, it will introduce in detail how to use RTN to classify remote sensing scene images. The core of RTN is to recursively discover transformation-invariant regions and learn the implied relationships between region-based feature representations. As shown in Figure 3.2, the RTN model consists of several main elements, including the recurrent warp operation, the bilinear pooling operation, the intra-scale classification loss L_{intra} and inter-scale pairwise ranking loss L_{inter}. Especially, the proposed RTN model ensures that the multi-scale transformed areas are automatically discovered and their canonical appearances are learned. By using the classification loss function and the pairwise ranking loss function, RTN effectively can handle the variations of the input data in a mutually enhanced manner, and then obtain competitive results on the publicly available RSSC datasets.

3.2.1 Recurrent Warp Operation

The recurrent warp operation is proposed to handle the variations of the input image by gradually learning the multiple scaled discriminant parts. Inspired by the spatial transformer networks (STN) (Jaderberg et al., 2015), learning invariance of input data can improve the generalisation ability of the CNN model. In the original STN paper (Jaderberg et al., 2015), multiple CNN streams are considered independent. Namely, an individual stream only responds to learn specific features that are invariant to transformations such as cropping, translation and scaling. In this way, it not only ignores the latent relationship between different streams, but also neglects that region-based features are not sufficient to represent the infor-

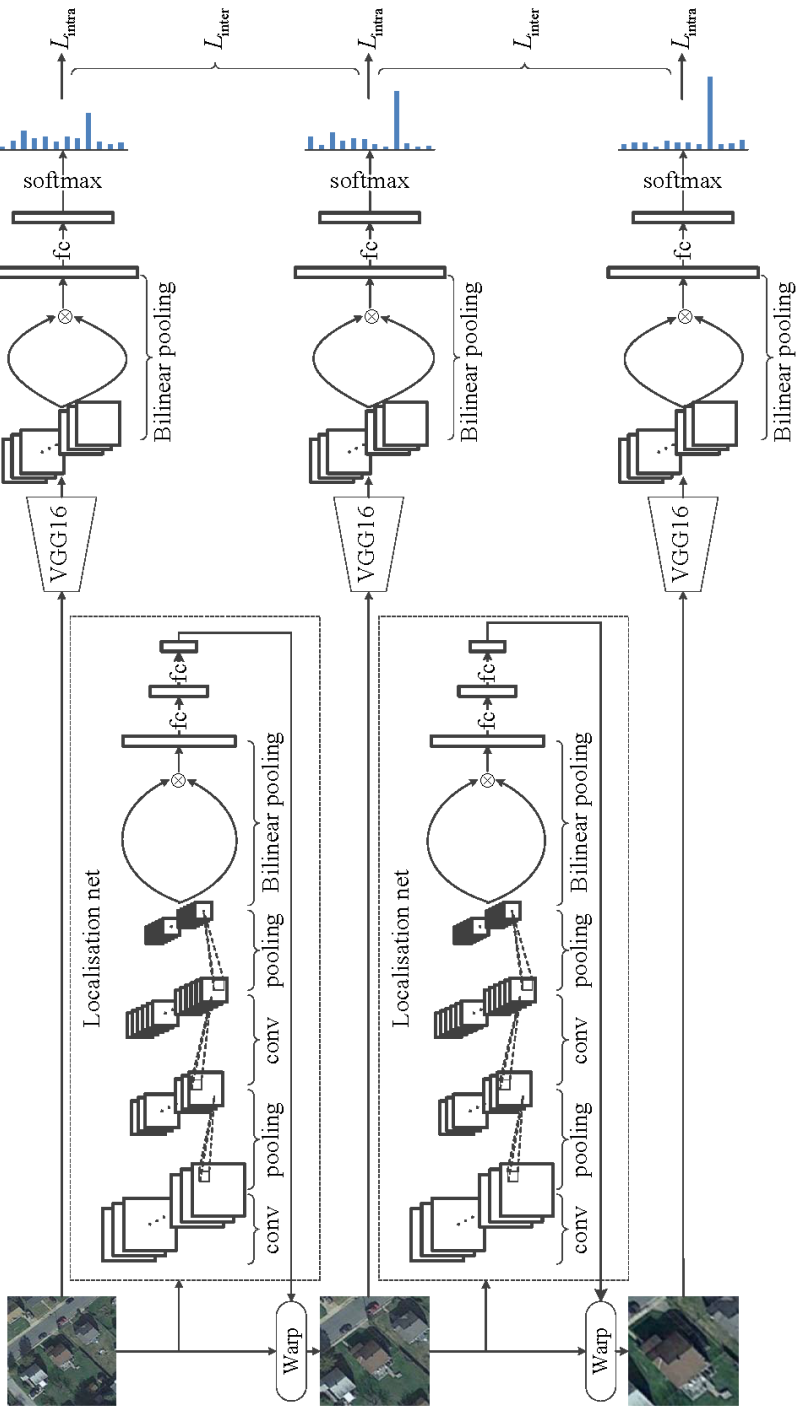


Figure 3.2: The overall structure of the recurrent transformer network (RTN). Given the input image, the localisation network will predict the transformer parameters accordingly by learning the features of the input image. By repeatedly applying warp operations, the network can gradually focus on distinguishing regions and generate multi-scale sub-images (i.e., a total of three streams in the scheme). The classification loss L_{intra} is used to evaluate the results of each stream, and the pairwise ranking loss L_{inter} is applied to discover the relationship between the current stream and nearby streams. All regression layers are based on the bilinear pooling denoted as $\otimes$. Where $conv$, $pooling$ and fc represent the convolutional layer, the max-pooling layer and the fully-connected layer, respectively.

mation contained in the whole image. In order to solve these shortcomings, it is assumed that there are implicit associations between different CNN streams, and it is necessary to incorporate these associations during training to improve the discriminative ability of CNN features.

To achieve the above goals, a recurrent warp operation is proposed, which can be used to intercept multiple discrimination regions from the original image in a recursive manner. This also makes the proposed RTN completely different from the original STN (Jaderberg et al., 2015) in terms of learning strategy. The learning process can be specifically denoted as:

$$I^{(s)} = f_{warp}(\theta^{(s)}\tau^{(s)}, I^{(s-1)}), \tag{3.1}$$

where I^s expresses the s^{th} scale image (Note: I^0 is the raw image). θ^s is the transformation parameters and can be computed by the function: $\theta^s = f_{loc}^{(s)}(I^{s-1})$. τ^s is the target coordinates of the regular grid located in the output image or feature map. Each warp operation f_{warp} will follow the similar processing method as described in the original STN (Jaderberg et al., 2015). Taking the first warp operation as an example, the raw image I^0 is fed into the localisation network $f_{loc}^{(s)}(I^{s-1})$ to generate the transformation parameter θ^1. Suppose the i^{th} target point of output image as $\tau_i^{(s)} = [x_i^{(s)}, y_i^{(s)}, 1]^T$, the corresponding source coordinates are generated in the following way:

$$\begin{bmatrix} x_i^{(s-1)} \\ y_i^{(s-1)} \\ 1 \end{bmatrix} = \begin{bmatrix} \theta_{1,1}^s & \theta_{1,2}^s & \theta_{1,3}^s \\ \theta_{2,1}^s & \theta_{2,2}^s & \theta_{2,3}^s \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i^{(s)} \\ y_i^{(s)} \\ 1 \end{bmatrix}, \tag{3.2}$$

Suggested by (Jaderberg et al., 2015), the above transformation can be regarded as a unique attention mechanism with simply forcing $\theta_{1,2}^s$ and $\theta_{1,2}^s$ to 0. In this way, it

can implement operations such as cropping and translation. The sampling kernel also needs to be incorporated in the warp operation and process the input image I^{s-1} to produce the value at a particular pixel in the next scaled image I^s. For example, a more refined amplified output can be obtained by employing standard bilinear interpolation on the previous input. The process can be written as:

$$I_i^{(s)} = \sum_{\tilde{h}=1}^{H} \sum_{\tilde{w}=1}^{W} I_{\tilde{h}\tilde{w}}^{(s-1)} \max(0, 1 - |x_i^{(s-1)} - \tilde{w}|) \max(0, 1 - |y_i^{(s-1)} - \tilde{h}|)), \quad (3.3)$$

where H and W denote the height and width of the input image $I^{(s-1)}$. The superscript of the Equation 3.3 is omitted because the channels are done identically in the warp operation. By repeating the warp operation, the network can automatically generate multiple fine regions containing discriminative information.

3.2.2 Intra-scale Loss and Inter-scale Loss

Based on the recurrent warp operation, the proposed RTN can generate the most relevant regions from coarse to fine. Next, it is necessary to extract distinguishable feature representations for each stream. However, the commonly used first-order pooling method (the global average pooling is also known as the mean vector) inserted at the end of the network explicitly ignore spatial information in the process of modelling statistical representations. To cope with this problem, (Lin et al., 2015) proposed a simple yet effective pooing method, which attempts to preserve the spatial information of CNN features by collecting their second-order statistical information. Since the second-order statistical feature does not introduce invariance into the deformation as completely as the first-order feature but maintains its selectivity, the second-order statistical feature is significantly better than the traditional first-order feature in terms of classification performance. (Kong & Fowlkes, 2017). Therefore, the second-order statistical features are adopted to

replace the first-order features to enhance the distinguishability of CNN features. The standard bilinear pooling is usually written as:

$$B(\mathcal{X}) = \sum_{i=1}^{hw} x_i x_i^T, \tag{3.4}$$

where $B(\mathcal{X}) \in \mathbb{R}^{c \times c}$ denotes the outer product of the non-activated CNN features $\mathcal{X} \in \mathbb{R}^{h \times w \times c}$ and $x_i \in \mathbb{R}^c$. For example, $\mathcal{X}$ could be the convolutional features generated from $conv5\_3 \in \mathbb{R}^{14 \times 14 \times 512}$ in VGG16 (Simonyan & Zisserman, 2014). According to the Eq.(3.4), a second-order feature based on bilinear pooling can be obtained with a dimension of $c \times c$. After the conventional vectorisation process, the fully-connected layer can map the feature representation to the feature vector matching the response category. Finally, the obtained feature vector can be input into the softmax function and then converted into a probability output. The progress can be written as:

$$\mathbf{p}(B(\mathcal{X})) = f\left(\text{vec}(W^T W) \times \text{vec}(B(\mathcal{X}))\right), \tag{3.5}$$

where $\text{vec}(W^T W) \in \mathbb{R}^{c \times c}$ represents the overall parameters of the vectorised bilinear pooling and $\mathbf{p}$ is the probability distribution of category entries. Once the probability is generated, the whole framework can be optimised by utilising the cross-entropy loss function. However, using only the classification loss will lose the correlation between different streams. To alleviate the impact of this problem, the pairwise ranking loss function is imposed on each neighbouring stream. The overall objective function of the proposed RTN model can be denoted as:

$$L = \sum_{s=1}^{S} L_{intra}^{(s)} + \alpha \sum_{s=2}^{S-1} L_{inter}^{(s)}, \tag{3.6}$$

where α is a hyperparameter worked as a constraint to learn the latent relationship between neighbouring streams and then adjust the total loss. $L_{intra}^{(s)}$ is the intra-scale loss which has shown in Figure 3.2. By using the softmax function, it can calculate the cross-entropy between the encoded labels and the estimated probability. The intra-scale loss can be written as:

$$L_{intra}^{(s)} = -\sum_{k=1}^{N} P_k^* \log P_k^{(s)}, \tag{3.7}$$

where N is the number of categories. To ensure that the streams learn a mutually reinforcing manner, an inter-scale loss is introduced in adjoining scales, which is defined as:

$$\begin{aligned} L_{inter}^{(s)} &= \max\left(0, \sum_{k=1}^{n} P_k^* \left(\log P_k^{(s+1)} - \log P_k^{(s)}\right) - margin\right), \\ &= \max\left(0, L_{intra}^{(s+1)} - L_{intra}^{(s)} - margin\right), \end{aligned} \tag{3.8}$$

in particular, it enforces $L_{intra}^{(s+1)} < L_{intra}^{(s)} + margin$ when training the network. Through this ingenious design, each finer scale is closely associated with the nearest former scale and then the classification accuracy is improved by progressively zooming in the distinguishable image regions. The final accuracy is determined by considering multiple scales, thereby reducing the impact of visual semantic gaps. Therefore, RTN can effectively improve the performance of remote sensing scene image classification.

3.2.3 Gradient Descent Analysis

In this subsection, it will demonstrate how RTN performs gradient optimisation. The gradient of the warp operation can be found in the original STN paper (Jaderberg et al., 2015). The bilinear pooling computes the outer product of matrices and it

is fully differentiable which can be optimised by the standard back-propagation method (Lin et al., 2015). Therefore, it will provide update rules for the combination of inter-scale and intra-scale loss functions. Without losing of generality, it will consider the convolution weight $\bar{w}$ of the scale s in the feature extraction based on VGGNet (Simonyan & Zisserman, 2014). Concretely, the update of the weight $\bar{w}$ can be calculated using stochastic gradient descent (SGD):

$$\begin{aligned}\bar{w} &= \bar{w} - \frac{\eta}{m}\sum_{i=1}^{m}\frac{\partial L_i}{\partial \bar{w}}, \\ &= \bar{w} - \frac{\eta}{m}\sum_{i=1}^{m}\frac{\partial\left(L_{intra,i}^{(s)} + \alpha L_{inter,i}^{(s-1)} + \alpha L_{inter,i}^{(s)}\right)}{\partial \bar{w}}, \\ &= \bar{w} - \frac{\eta}{m}\sum_{i=1}^{m}\left(\left(1 + \alpha\delta_i^{(s-1)} - \alpha\delta_i^{(s)}\right)\frac{L_{intra,i}^{(s)}}{\partial \bar{w}}\right),\end{aligned} \tag{3.9}$$

where η denotes the initial learning rate, α is a hyper-parameter that has been introduced in Eq.(3.6), L_i refers to the value of the loss function at the i-th training sample, m is the batch size, η is associated with Eq.(3.8) to decide the options of the returned value, which can be defined as:

$$\delta_i^{(s-1)} = \begin{cases} 1, & \text{if } L_{\text{intra,i}}^{(s-1)} < L_{\text{intra,i}}^{(s)} - \text{ margin }; \\ 0, & \text{otherwise }; \end{cases} \tag{3.10}$$

more specifically, the value of the δ refers to the degree of relevance of the adjacent scales. For instance, if the intra-scale loss of I^s is significantly higher than I^{s-1}, the learning rate of the weights needs to be increased by α to shorten the differences between I^s and I^{s-1}, and vice versa.

3.3 Experiments

3.3.1 Implementation Details

For a fair comparison, the proposed RTN was evaluated using the backbone of the VGGNet-16 network (Simonyan & Zisserman, 2014) that has been pre-trained on the large-scale ImageNet dataset. To avoid the *orderless* problem of features, the *conv5_3* features in the VGGNet-16 network are extracted by removing the max-pooling layer. The localisation network is composed of two convolutional layers and each convolutional layer is followed by a max-pooling layer. The second-order statistical features will be aggregated, and then two fully-connected layers that can be used to predict the transformation parameters. It is worth noticing that the proposed RTN was trained without utilising the conventional data augmentation method described in the original STN (Jaderberg et al., 2015). Besides, the raw images were resized to 224 $\times$ 224 resolutions and then feed into RTN. The initial learning rates were set to 0.0001 and 0.01 for the localisation network and classification network with a weight decay rate of 0.0005. The training batch size was 36. To ensure the model can be trained stably, α and $margin$ in Eq.(3.6) and Eq.(3.8) were set to 0.1 and 0.05 empirically. The model was trained iteratively for about 80k using standard SGD.

3.3.2 Experimental Results and Comparison

Existing methods applied to RSSC tasks can be approximately divided into two categories, namely methods based on deep learning and methods based on non-deep learning (Non-deep learning-based methods usually refer to handcrafted features or methods based on unsupervised learning). Table 3.1 summarises the overall accuracy and standard deviation of the previous methods. It can be clearly

Table 3.1: Comparison of the overall accuracy and standard deviation of the proposed RTN model with previous methods. T.R. is short for the training ratio.

		NWPU-RESISC45		AID		UC-Merced
		T.R.=10%	T.R.=20%	T.R.=20%	T.R.=50%	T.R.=50%
Non Deep Learning Methods	BoVW(SIFT) (Xia, Hu, et al., 2017)	41.72±0.21	44.97±0.28	61.40±0.41	67.65±0.49	71.90±0.79
	SPM(SIFT) (Xia, Hu, et al., 2017)	27.83±0.61	32.96±0.47	38.14±0.75	45.28±0.66	56.26±1.56
	LLC(SIFT) (Xia, Hu, et al., 2017)	38.81±0.23	40.03±0.34	56.36±0.68	59.92±0.63	69.41±1.14
Deep Learning Methods	Transferred AlexNet (Cheng, Han, & Lu, 2017)	76.69±021	79.85±0.13	86.86±0.47	89.53±0.31	95.02±0.81
	Transferred VGGNet-16 (Cheng, Han, & Lu, 2017)	76.47±0.18	79.79±0.15	86.59±0.29	89.64±0.36	94.14±0.69
	Transferred GoogLeNet (Cheng, Han, & Lu, 2017)	76.19±0.38	78.48±0.26	83.44±0.40	86.39±0.55	92.70±0.60
	D-CNN with AlexNet (Cheng et al., 2018)	85.56±0.20	87.24±0.12	85.62±0.10	94.47±0.12	96.67±0.10
	D-CNN with VGGNet-16 (Cheng et al., 2018)	89.22±0.50	91.89±0.22	90.82±0.16	**96.89**±010	**98.93**±0.10
	D-CNN with GoogLeNet (Cheng et al., 2018)	86.89±0.10	90.49±0.15	88.79±0.10	96.22±0.10	97.07±0.12
	RTN with VGGNet-16	**89.53**±0.21	**92.20**±0.34	**92.75**±0.21	95.09±0.16	98.33±0.71

seen that traditional machine learning algorithms pale in comparison with deep learning methods, and even transferred deep features can significantly improve the classification effect. Especially, on NWPU-RESISC45 datasets (Cheng, Han, & Lu, 2017), the accuracy of the proposed RTN model is approximately twice that of the typical bag-of-visual-words (BoVW) algorithm (Xia, Hu, et al., 2017). Furthermore, the proposed method obtains 89.53% on NWPU-RESISC45 with 10% training samples, which almost three times accurate than the spatial pyramid matching (SPM) based method (Xia, Hu, et al., 2017). The locality-constrained linear coding method (Xia, Hu, et al., 2017) achieves 69.41% accuracy on the small UC-Merced dataset (Y. Yang & Newsam, 2010), which is higher than the SPM-based method, but is far from the RTN method.

All the listed deep learning methods can achieve acceptable results, especially compared with traditional methods, the methods based on simple transferring deep learning feature can significantly improve the classification results. Surprisingly, the overall result obtained by GoogLeNet (Szegedy et al., 2015) with a relatively deep structure is not as good as that obtained by VGGNet-16 (Simonyan & Zisserman, 2014), and occasionally even worse than the shallowest AlexNet (Krizhevsky et al., 2012). This may be because the deeper the network, the learned high-level features for natural image processing tasks are not proper for RS images. On UC-Merced dataset (Y. Yang & Newsam, 2010), the transferred AlexNet reaches 95.02% accuracy, which is very close to the recent proposed D-CNN (Cheng et al., 2018) method. However, on the large-scale dataset like NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017), there exists visible distances between the transferred deep learning methods and the proposed RTN method (e.g, the relative gain is about 13% when using 10% of the training samples).

Before the proposed RTN model, the best accuracy on experimental RSSC datasets is made by the recently proposed D-CNN method (Cheng et al., 2018). The D-

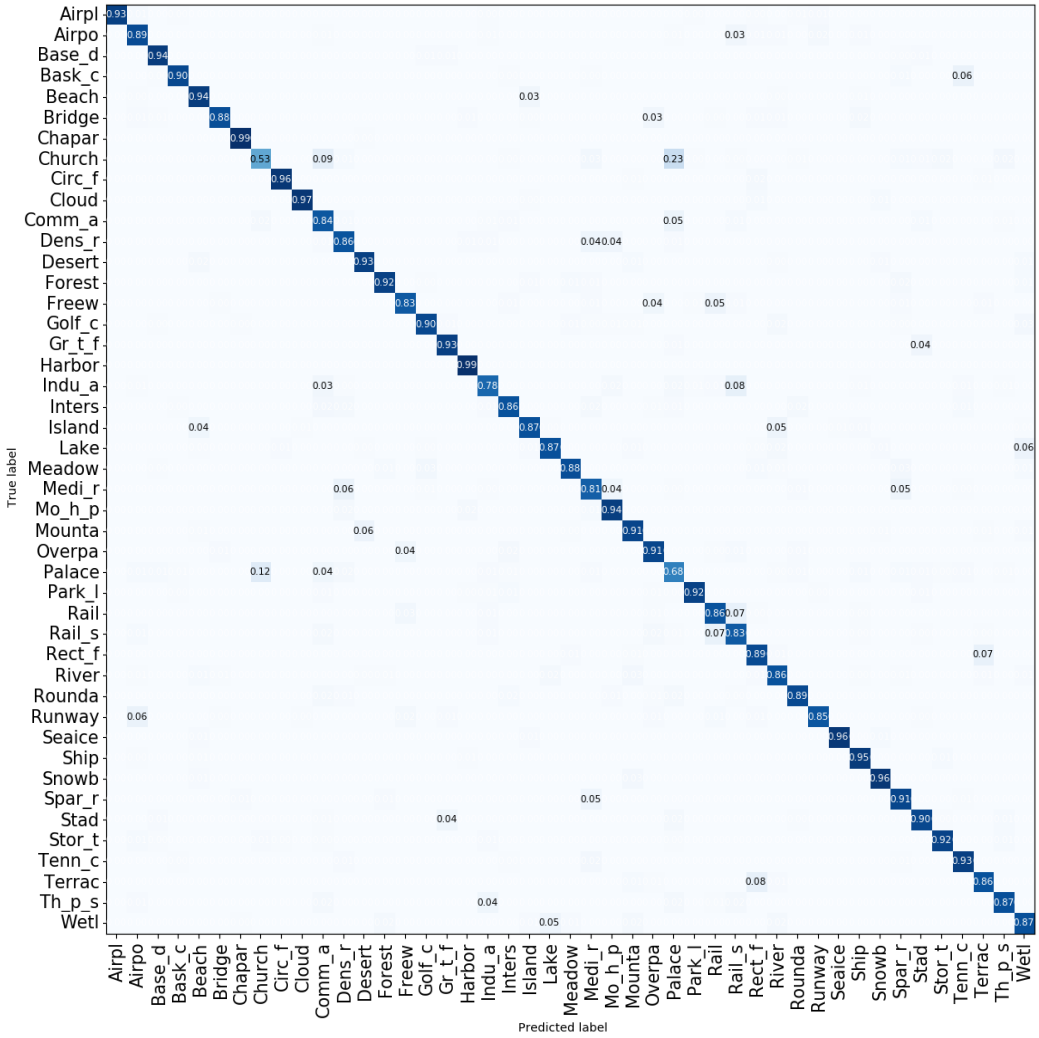


Figure 3.3: The confusion matrix on NWPU-RESISC45 dataset under the training ratio of 10%. For the sake of clarity, values less than 0.03% are omitted.

CNN model takes advantage of using metric learning as the regularisation term, which encourages the model to learn more discriminative features. As reported in (Cheng et al., 2018), D-CNN using VGGNet-16 generally performs better than using AlexNet and GoogLeNet. On UC-Merced dataset, the accuracy of D-CNN with VGGNet-16 (Cheng et al., 2018) presents the best among all the listed methods. By using 50% of the training samples, D-CNN with VGGNet-16 achieves 96.89% on AID dataset (Xia, Hu, et al., 2017), which is 1.8% higher than the

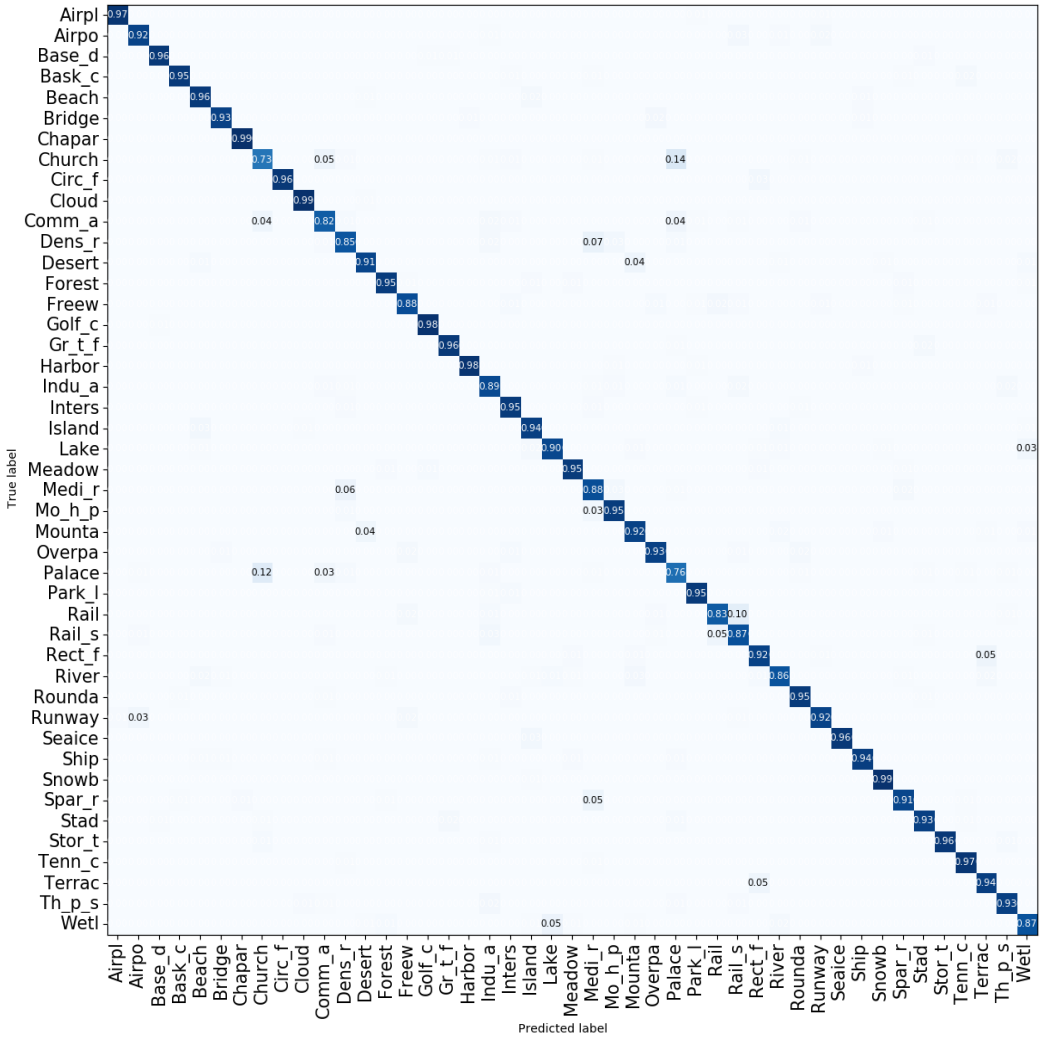


Figure 3.4: The confusion matrix on NWPU-RESISC45 dataset under the training ratio of 20%. For the sake of clarity, values less than 0.03% are omitted.

RTN model. However, the proposed RTN model obtained 92.75% accuracy on AID dataset under the 20% training ratio, with 1.95% gain compared with D-CNN with VGGNet-16. In addition, the RTN model reports the highest accuracy under two different partition ratios on the most challenging NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017).

The confusion matrix is an effective way to illustrate classification details at the

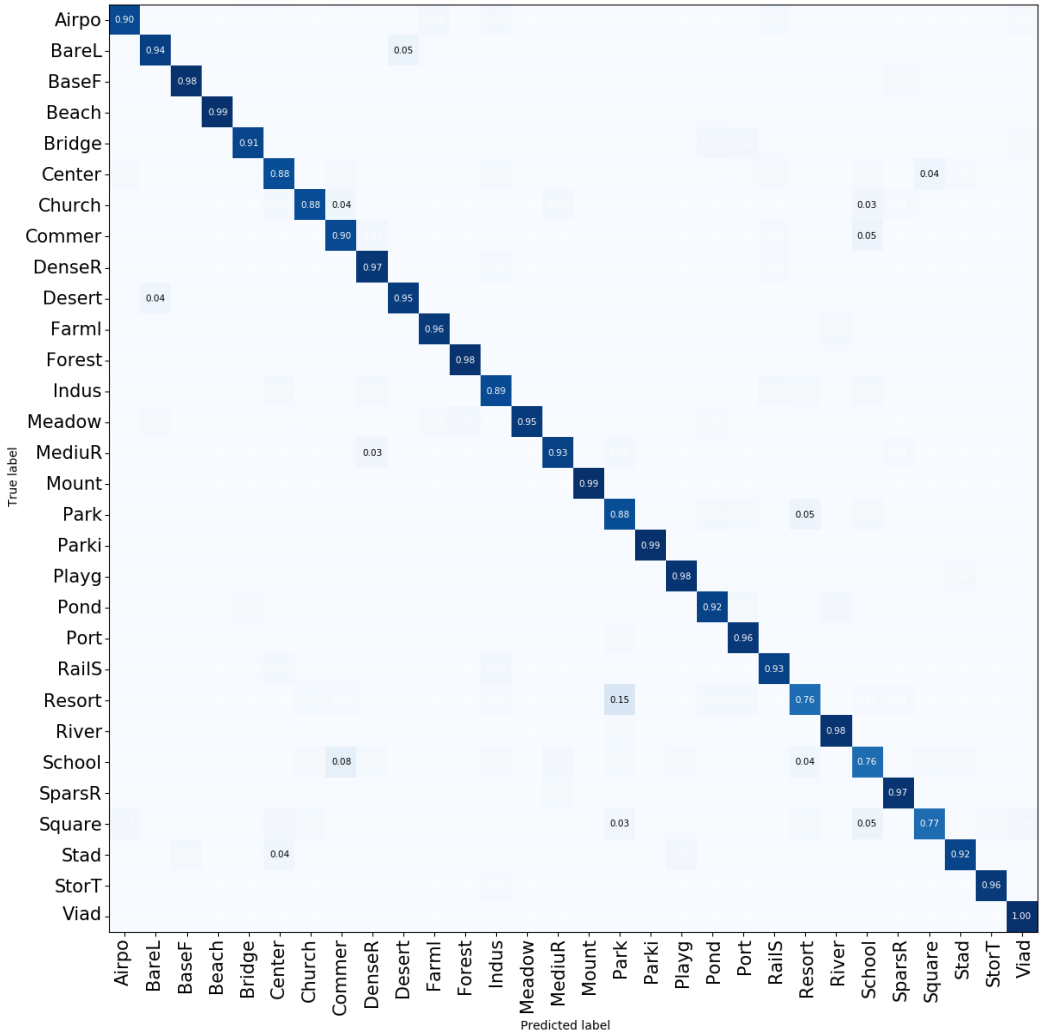


Figure 3.5: The confusion matrix on AID dataset under the training ratio of 20%. For the sake of clarity, values less than 0.03% are omitted.

class level and is used to evaluate the proposed method. To save space, it will randomly select a test result in different experimental settings for displaying. When using the RTN model on NWPU-RESISC45 dataset (10% training ratio), the number of categories with an accuracy of more than 80% is 42, while the number of categories has become 23 by using the Transferred VGGNet-16 (Cheng, Han, & Lu, 2017). For the **Palace** category, the accuracy of the proposed algorithm reaches 68%, which surpasses the Transferred VGGNet-16 (Cheng, Han, & Lu,

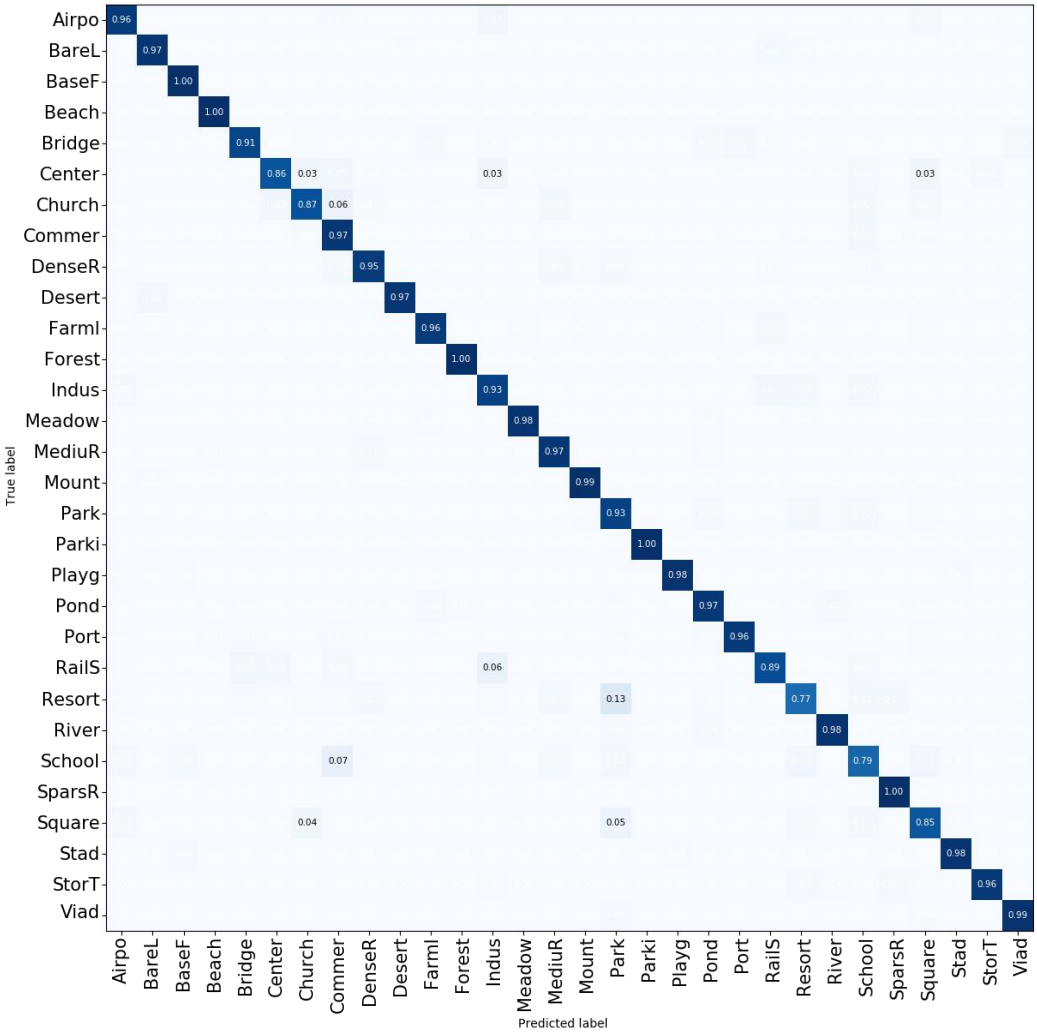


Figure 3.6: The confusion matrix on AID dataset under the training ratio of 50%. For the sake of clarity, values less than 0.03% are omitted.

2017) by a large margin of 21% (see Figure 3.3). When 20% of the samples of NWPU-RESISC45 dataset are available for training the model, the results obtained will also be significantly improved (see Figure 3.4). For example, the classification accuracy of **Palace** category obtained is 76%, which is 24% and 3% higher than the Transferred VGGNet-16 (Cheng, Han, & Lu, 2017) and D-CNN with VGGNet-16 (Cheng et al., 2018), respectively. As it can be seen in Figure 3.5, the most difficult to distinguish categories in AID dataset (with 20% training

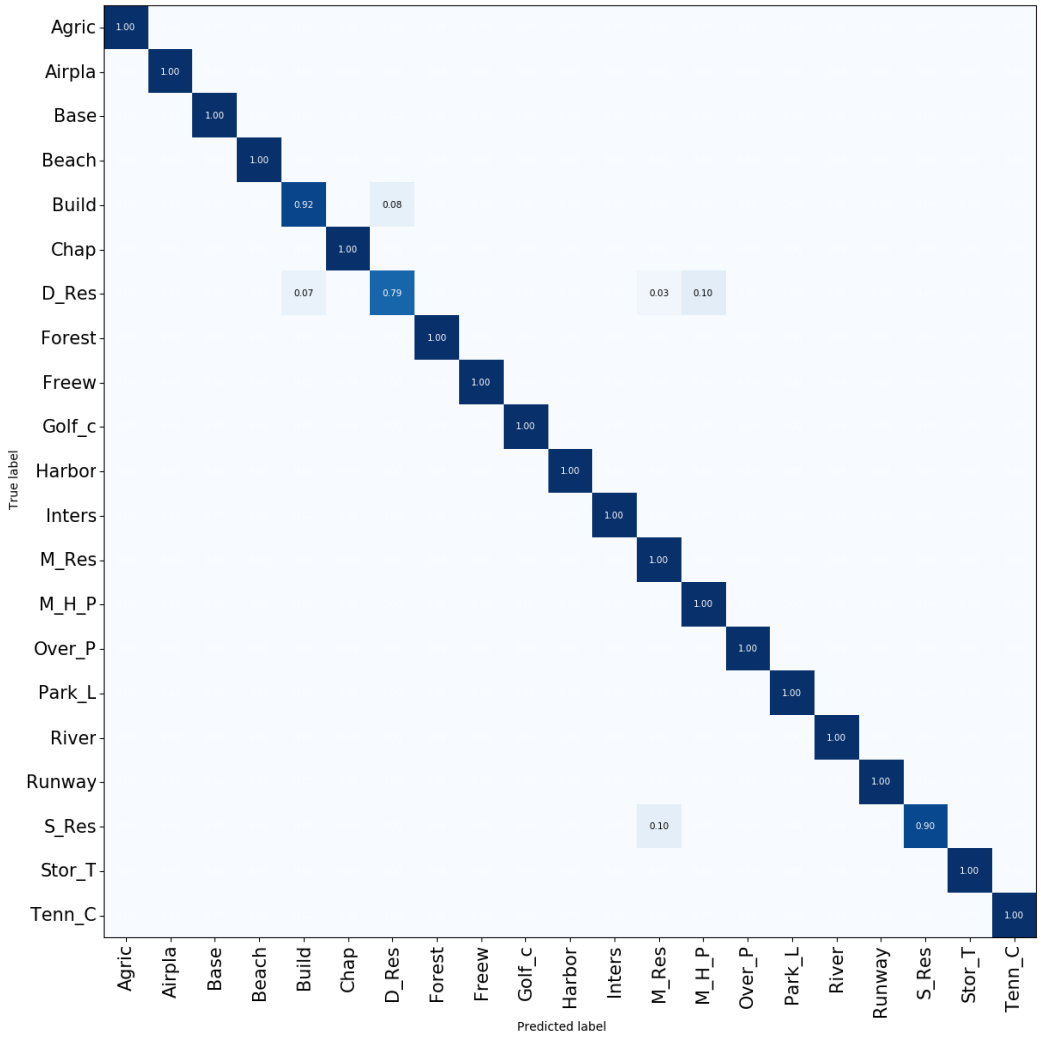


Figure 3.7: The confusion matrix on UC-Merced dataset under the training ratio of 80%. For the sake of clarity, values less than 0.03% are omitted.

ratio) are **Resort** and **Park**. The reason for this problem is that the frequency of co-occurring areas in the images of these two categories is relatively high. In particular, 15% of **Resort** images are misclassified into **Park** category. However, most categories of images can be classified correctly (i.e., more than 90%), and there are categories that can be completely predicted correctly such as **Viaduct** category. Figure 3.6 shows the confusion matrix obtained by the RTN model using 50% of the data on the AID dataset. With the increase of training data, the overall

Table 3.2: Comparison of the classification accuracy of RTN models with different number of scales and whether there is inter-scale loss used. Experiments are conducted on NWPU-RESISC45 dataset under the training ratio of 20% (Cheng, Han, & Lu, 2017).

Scales.	scale 1	scale 2	scale 3	scale (1+2)	scale (1+2+3) w/o L_{inter}	scale (1+2+3) w/ L_{inter}
Acc.	91.20%	91.84%	90.20%	92.35%	92.49%	**92.71%**

classification results have also improved, especially the classification accuracy of 21 categories has reached more than 95%. The correct classification of **Resort** images accounted for 77%, which is slightly higher than the result obtained using 20% of the training data. Since the number of categories in UC-Merced dataset is relatively small, it is comparatively easy to use 80% of the data to train the model and then predict the labels of the remaining data. From the diagonal colour of the confusion matrix in Figure 3.7 and the overall sparseness, it can be seen that almost all categories can be accurately classified. The two categories with larger errors are **Dense Residential** and **Sparse Residential**, in which 10% of the test images of these two categories are incorrectly classified as **Mobile Home park** and **Medium Residential**.

3.3.3 Qualitative Analysis and Visualisation

Table 3.2 first shows that classification accuracy varies with different scales. As it can be found that the second scale exhibits a higher accuracy than both the third scale and the first scale (the raw image). This phenomenon reflects that a finer scale can be used to improve the classification accuracy, but excessive amplification will damage the classification accuracy. Furthermore, the result of combining multiple different scales is higher than the result of any individual scale. Especially, the average of the three proposed scales is better than the average of the

Figure 3.8: Visualisation of test images selected from NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017). The first row represents the raw image, and the second and the third rows are two finer scales.

first and second scales (92.49% vs 92.35%). In addition, it also evaluates the effectiveness of using L_{inter} between adjacent scales. More specifically, by imposing a complete pairwise ranking loss on three different scales, it can obtain a classification accuracy of 92.71%, which is the best result in all cases.

As mentioned in the method section, it can be regarded as an attention mechanism by specifically setting the value of the affine transformation matrix. Figure 3.8 exhibits how the randomly selected test images (not cherry-picked) gradually discover more detailed discriminatory areas in the image through this attention mechanism. Without manual intervention, the proposed RTN model can automatically learn multiple regions that are gradually amplified, and can also adjust the input image and the localised finer regions to the most canonical appearance, thereby significantly improving the interpretability of the model.

3.4 Conclusion

In this chapter, a novel method called Recurrent Transformer Network (RTN) is introduced that can be used to improve the performance of remote sensing scene classification. The model benefits from a multi-stream transformation mechanism under the constraint of the pair-wise ranking loss to find multiple discrimination regions that are conducive to classification and powerful bilinear feature learning. Nevertheless, the RTN model has two critical issues need to be discussed. First, why does the multi-stream feature based on the gradually enlarged image area help improve the classification performance? Second, can the slow-converging STN (Jaderberg et al., 2015) be replaced by a more effective method? In the next chapter, it will explore these problems in depth from the dataset structure and design a new and more efficient network.

4 | Multi-Granularity Canonical Appearance Pooling

4.1 Introduction

In the previous chapter (Chapter 3), the recurrent transformer network (RTN) (Z. Chen et al., 2018) learned different levels of features with multiple interrelated stream models and introduced elaborate affine variations for the input images through the ingenious use of spatial transformations. The parameterised transformation method can broadly introduce various transformations, but it also means that it takes longer to find the most appropriate transformation (Jaderberg et al., 2015) and is extremely sensitive to initial values of the transformation matrix and the learning rate. This motivates me to explore a new structured transformation method to more efficiently reduce the impact of the above problems.

Before proposing new solutions, it is necessary to briefly revisit the challenging problems in RSSC tasks to be solved and the reasons for these problems. The first thing to bear is the huge visual-semantic discrepancy caused by the lack of precise alignment between visual features and semantic labels. Especially for remote sensing images full of heterogeneous content, the datasets they belong to lack well-constructed ontology structure, which apparently causes the high-level semantics in the category labels cannot be included in the learned features. Another challenging problem to be solved is the naturally presenting variations in the remote sensing scene datasets. Specifically, the existing variations can be summarised as intra-class diversity and inter-class similarity. As shown in Figure 4.1, the left *railway* image is visually similar to *freeway* images but is different from the right *railway* image belonging to the same category.

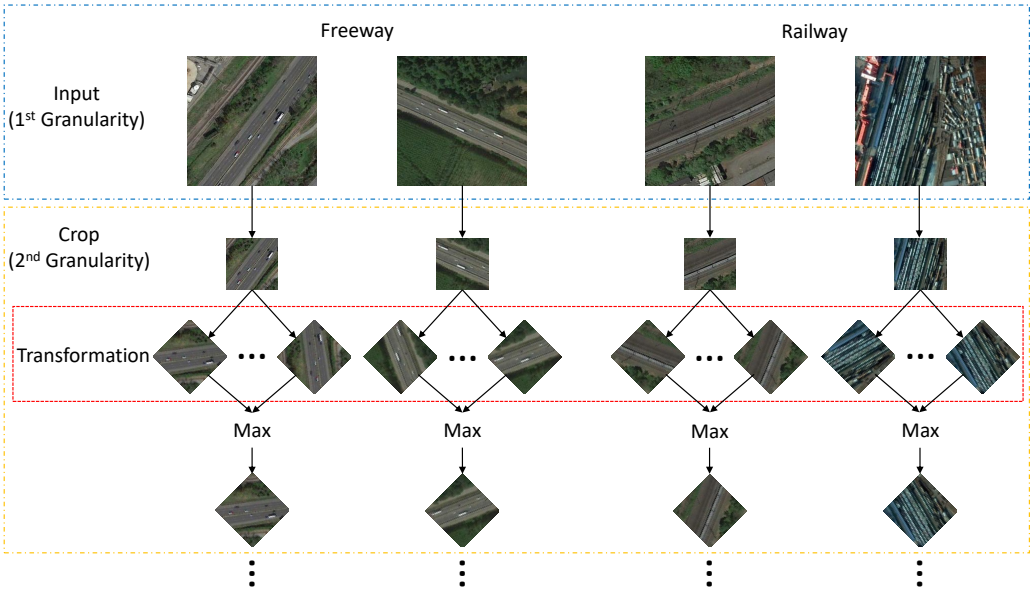


Figure 4.1: Example images selected from two different categories in NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017). In order to distinguish visually similar images, it is necessary to zoom in to observe the subtle differences. However, the differences will be more significant and vivid if the zoomed regions can be transformed into their canonical appearances.

Providing detailed annotations for all heterogeneous regions in each image may be the most direct and effective way to solve the above problems. However, collecting well-annotated data is impractical because it requires massive amounts of manpower and is time-consuming and subjective. These problems are even more critical in remote sensing datasets since many categories have hierarchical ontologies. For example, in NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017), *airplane* and *runway* may belong to the same parent category *airport*, similarly, *railway* and *railway_station* may come from the category of *railway* while*bridge* pertains to *freeway*. Moreover, *airport*, *railway* and *freeway* are the branches of *transportation*. In the taxonomy, these relationships can be categorised into three levels according to the class inclusion and degree of specificity, including the superordinate-level categories, the basic-level categories and the subordinate-

level categories (Ungerer & Schmid, 2013; Croft & Cruse, 2004). Namely, the further up in the taxonomy a category is located, the more general it is, and vice versa. It is relatively easy to classify the superordinate-level categories *transportation* or the basic-level categories like *railway* and *airport*, but identifying subordinate-level classes requires more discriminative features, such as *airplane* and *runway*, *railway* and *railway_station*, as well as *bridge*. Similar hierarchical relationships can also be found among categories in AID dataset (Xia, Hu, et al., 2017), such as *dense residential*, *medium residential* and *sparse residential*.

Based on the above findings, an assumption can be established that there is a latent ontology between the basic-level and the subordinate-level category labels in remote sensing scene datasets. As discussed earlier, incorporating the latent hierarchical structures is a feasible solution for decreasing the large visual-semantic discrepancy. However, manually designed ontologies are expensive to acquire and often suffer from subjective problems. Therefore, another strategy worth choosing is to incorporate hierarchical information by learning granular feature representation. Notably, the desired learning features should not only contain distinctive information from different granularities, but also be consistent with the underlying ontological structures of the datasets.

To achieve the above goals, a novel multi-granularity canonical appearance pooling model (MG-CAP for short) is proposed to learn the granular feature representation for classifying RS scene images (seen in the Figure 4.2). In this model, sub-images symbolising different granularities are generated by gradually cropping the raw image multiple times. For each specific level of granularity, a certain number of instances can be generated based on a set of predefined transformations with the same number. Inspired by (Bromley et al., 1993), the Siamese-style CNN architecture is applied to extract features and learn the dependencies between different instances. Then, the second-order statistics of the standard CNN

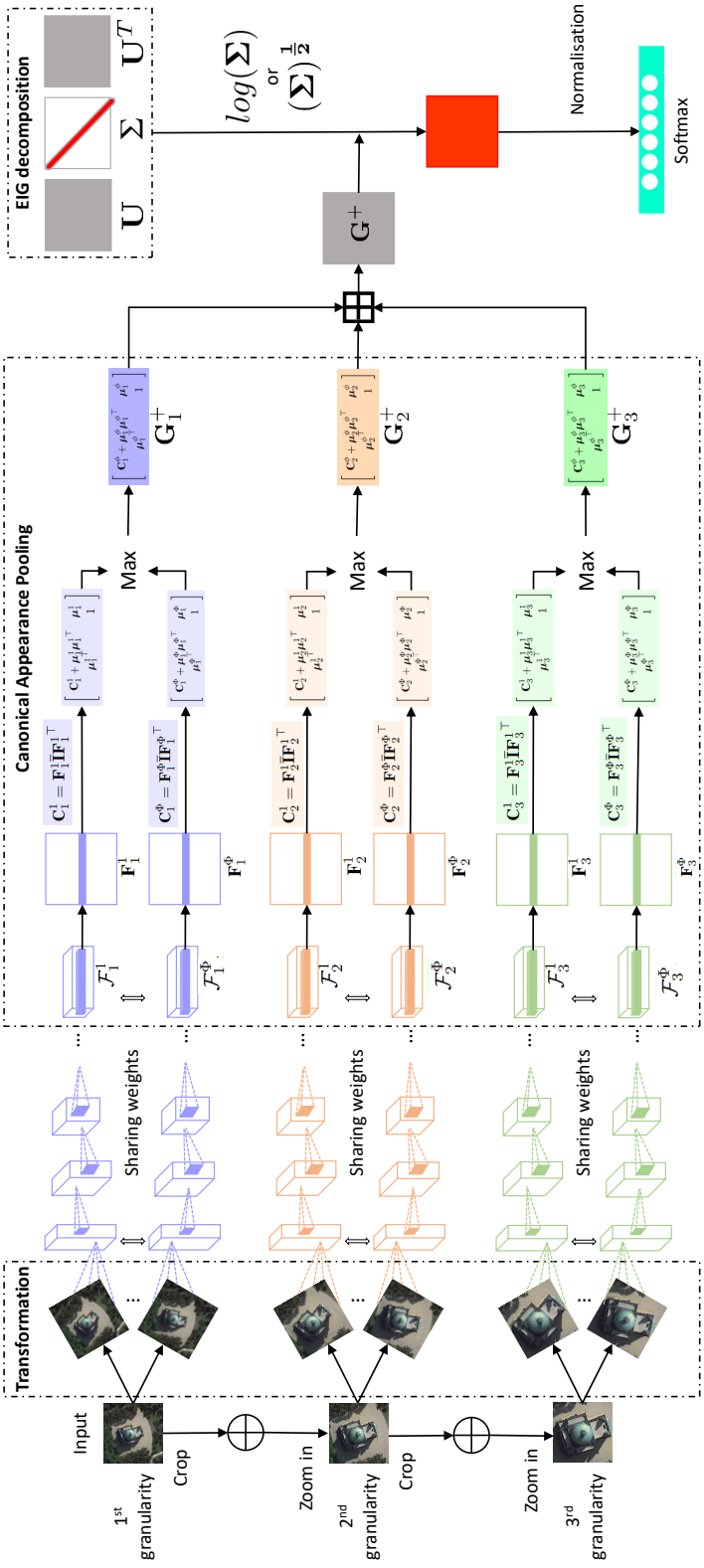


Figure 4.2: An overview of the proposed MG-CAP framework. It investigates three different granularities from coarse to fine. At one specific grain level, the image is transformed according to a set of pre-defined transformation Φ. Then, the transformed image as instances will be fed into the Siamese networks for feature extraction $\mathcal{F}_s^{\phi}$. $\mathcal{F}_s^{\phi}$ will be transformed into a Gaussian covariance feature $(\mathbf{G}_s^{\phi})^+$. Subsequently, an element-wise max operation is used to learn the optimal covariance feature $\mathbf{G}_s^+$. To capture multi-grained information, it applies an element-wise stacking operation $\boxplus$ and averages to obtain $\mathbf{G}^+$. The obtained $\mathbf{G}^+$ is an SPD matrix, which can be factorised by EIG decomposition through powerful matrix normalisation methods.

features are summarised and transformed into Gaussian covariance matrices as the global representation. At the end of the Siamese architecture, the maximum operation is selected to produce a unique Gaussian covariance matrix as a feature corresponding to the canonical appearance of the generated image instance. The function learned in this way is guaranteed to be invariant to the predefined global transformations, which can mitigate the impact of large intra-class variations. The obtained multiple Gaussian covariance matrices are Symmetric Positive Semi-definite (SPD) matrices, which have been endowed with a special geometric structure (i.e., pseudo-Riemannian manifold). In addition, it also implements the non-linear EIG decomposition function supported by the GPU, and allows the use of appropriate matrix normalisations to learn the geometric structure of pseudo-Riemannian manifolds. Finally, it combines different granular features and feeds the results into the classifier. The contributions of MG-CAP model can be summarised as follows:

- It derives a novel Multi-Granularity Canonical Appearance Pooling, which incorporates the latent ontological structure of remote sensing scene datasets, thereby alleviating the visual-semantic discrepancy.
- It progressively leverage the Siamese-style CNN architecture to learn transformation invariant features to solve the large intra-class variation problem.
- It offers a stable EIG-decomposition function supported by the GPU, which makes the exploitation of Gaussian covariance geometry more efficient by using different matrix normalisations.

4.2 Method

4.2.1 Overview

The core idea of the proposed MG-CAP model is to seek a feasible solution to learn multiple transform-invariant features in a fine-grained manner, so as to reduce the large visual-semantic disparity and nuisance variations in the RSSC task without the need of detailed part annotations. The flow chart of the MG-CAP model has been presented in Figure 4.2. Throughout the chapter, it will employ boldface lowercase letters (e.g., $\mathbf{v} \in \mathbb{R}^{I_1}$), boldface uppercase letters (e.g., $\mathbf{M} \in \mathbb{R}^{I_1 \times I_2}$) and calligraphic letters (e.g., $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$) to denote vectors, matrices and higher-order tensors, respectively. Given an input image $\mathcal{X} \in \mathbb{R}^{H \times W \times C}$, where H, W and C are the image height, width and channels.

The input image will be cropped multiple times to generate sub-images of different granularities, and all sub-images will be adjusted to a uniform scale through bilinear interpolation algorithms to facilitate subsequent processing. Each granular image is transformed according to a set of predefined transformations, and then the Siamese-style CNN network is used to extract deep learning features for all transformed instances: $\mathcal{F}_s \in \mathbb{R}^{H^{'} \times W^{'} \times C^{'}}$, where s is the index of granularity. Inspired by the recent success of second-order statistical research (Acharya et al., 2018; Ionescu et al., 2015; Lin et al., 2015; Lin & Maji, 2017), it flattens ordinary CNN $\mathcal{F}_s$ to generate features in matrix form $\mathbf{F}_s \in \mathbb{R}^{H^{'} W^{'} \times C^{'}}$. The obtained matrix features will be converted into a covariance matrix and summarised as a Gaussian SPD matrix represented by $\mathbf{G}_s^{+} \in \mathbb{R}^{(C^{'}+1) \times (C^{'}+1)}$. Then it will use element-wise maximum and mean operations in turn to obtain the feature with the largest response in each granularity and the average of all granularities. The formed SPD matrix will go through the EIG function with appropriate matrix normalisations

to further improve its discriminative power and regard it as the final representation. The goal of the task is essentially to learn discriminative features containing multi-granularity information, and then use it to generate a probability distribution **p** over all categories. This process can be written as:

$$\mathcal{X} \mapsto \mathbf{G}^{+} \in Sym^{+}, \qquad \mathbf{p}(\mathbf{G}^{+}) = f_r(\mathbf{W} \circ \mathbf{G}^{+}) \tag{4.1}$$

where $\mathcal{X} \mapsto \mathbf{G}^{+}$ denotes the procedure of achieving Gaussian covariance features $\mathbf{G}^{+}$ from an input image $\mathcal{X}$. The Sym^{+} is used to represent the property of the positive semi-definiteness of the SPD matrix. It is worth mentioning that $\mathbf{G}^{+}$ is an SPD matrix, because it is the average product of the multiple SPD matrices at different granularities. $f_r(\cdot)$ represents the softmax layer, which maps the weighted SPD matrix $\mathbf{W} \circ \mathbf{G}^{+}$ to the feature vector and then converts the results to probabilities. $\mathbf{W}$ indicates that the overall model parameters of the feature representation $\mathbf{G}^{+}$ which can be achieved by averaging the multiple granular SPD matrices in an element-wise manner. It can be represented as:

$$\mathbf{G}^{+} = \frac{1}{S} \sum_{s=1}^{S} \mathbf{G}_s^{+} \tag{4.2}$$

$\mathbf{G}_s^{+}$ represents the SPD matrix corresponding to a specific granularity, which can be derived from the canonical appearance pooling layers (see next subsection). S is the total number of granularities. Once all the canonical SPD matrices have been obtained, the channel-wise averaging operator can be applied to yield the unique SPD matrix $\mathbf{G}^{+}$ to incorporate information from different granularities.

4.2.2 Canonical Appearance Pooling Layers

Canonical appearance pooling layers are proposed to learn transformation-invariant features. For multiple instances generated, a multi-column network can be considered to process each instance and average the results of all individual networks to obtain the final prediction. This process is known as multiple instance learning (MIL) (Dietterich, Lathrop, & Lozano-Pérez, 1997) and can be simply expressed as: $\mathcal{B}(\mathcal{X}) = \underset{\phi \in \Phi}{mean}\, \mathcal{A}(\phi(\mathcal{X}))$, where Φ, $\mathcal{B}$ and $\mathcal{A}$ denote the set of transformations, the algorithm output and input, respectively. The algorithm $\mathcal{B}(\mathcal{X})$ in this way is given transformation-invariant property (J. Wu, Yu, Huang, & Yu, 2015), but this invariance is for the model as a whole rather than for individual features (Laptev, Savinov, Buhmann, & Pollefeys, 2016). In order to take full advantage of the interdependence between the individual features, a proposal is presented to learn the features of the transformed instance that has the highest response to classification. In particular, the Siamese-style CNN architecture (Bromley et al., 1993) is introduced in this scenario to comprehend the inherent relationship between individual features and avoid the explosive growth of the number of model parameters. The process is written as:

$$\mathcal{F}_s^{\phi} = f_e\left(\phi\left(\mathcal{X}_s\right)\right), \tag{4.3}$$

where $\phi \in \Phi$ is the set of pre-defined transformations. In this chapter, it only consider rotation transformations that can be derived from: $\phi_r = \frac{360^\circ}{\dim(\Phi)}$, with $\dim(\cdot)$ denotes the length of the transformation set, $\phi(\mathcal{X}_s)$ represents the transformed images, and $f_e(\cdot)$ indicates the feature extraction process using the standard deep learning architecture.

The above process is simple to implement yet remains invariant under certain transformations. To learn the optimal feature representation from the transformed instances, a simple maximum operator is adopted to produce a unique feature in

an element-wise manner. Formally,

$$\mathcal{F}_s^{\phi} \mapsto (\mathbf{G}_s^{\phi})^{+}, \qquad \mathbf{G}_s^{+} = \max_{\phi \in \Phi} f_c\left((\mathbf{G}_s^{\phi})^{+}\right), \tag{4.4}$$

where $\mathcal{F}_s^{\phi} \mapsto (\mathbf{G}_s^{\phi})^{+}$ is a procedure that transforms CNN features into Gaussian covariance matrices. $f_c(\cdot)$ is adopted to learn the optimal second-order feature from the accumulated covariance matrices $(\mathbf{G}_s^{\phi})^{+}$. Hence, the generated feature $\mathbf{G}_s^{+}$ can be regarded as a new feature with transformation-invariant properties. Since the texture of the input image varies with the granularity, the weight sharing in the Siamese architecture is only allowed to be used when extracting features for instances at specific granularity, so that the independence of an individual granularity can be guaranteed to the greatest extent.

The global average pooling layer that is often attached to the end of the deep learning architecture only captures the first-order statistical information of CNN features while neglecting the correlations between the spatial positions and channels. This first-order pooling method retains the invariance of CNN features, but it is often more reasonable to maintain the selectivity of spatial information for image classification tasks (Kong & Fowlkes, 2017). In order to maximise the preservation of the spatial information in the discriminated area of the image, the traditional CNN features are written in the form of a matrix and then their covariance is calculated.

At a specific granularity, the covariance feature $\mathbf{G}_s^{+}$ with transformation invariance can be obtained through the Eq.(4.4). Specifically, ordinary CNN features $\mathcal{F}_s^{\phi}$ can be expressed in a matrix form $\mathbf{F}_s^{\phi} = [\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_N]$ by flattening the spatial structure of $\mathcal{F}_s^{\phi}$, where $\mathbf{f}_i \in \mathbb{R}^{C'}$ and $N = H' \times W'$. In this way, the following computation of the covariance matrix $\mathbf{C}_s^{\phi}$ can be seen as the compact summarisation of the

second-order information of $\mathcal{F}_s^{\phi}$, which is given by:

$$\mathbf{C}_s^{\phi} = \mathbf{F}_s^{\phi}\bar{\mathbf{I}}\mathbf{F}_s^{\phi\top}, \tag{4.5}$$

where $\mathbf{C}_s^{\phi} \in \mathbb{R}^{C^{'}\times C^{'}}$. $\bar{\mathbf{I}}$ can be calculated as: $\bar{\mathbf{I}} = \frac{1}{N}(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^{\top})$ with $\mathbf{I} \in \mathbb{R}^{N\times N}$ and $\mathbf{1} = [1, \ldots, 1]^{\top}$, where N represents the vector size.

The obtained $\mathbf{C}_s$ encodes the second-order statistics of local CNN features. In particular, the covariance matrix $\mathbf{C}_s$ is a SPD matrix when its components are linearly independent in the corresponding vector feature space $\mathbf{F}_s^{\phi}$ and the spatial number N is greater than C$^{'}$. As suggested by (Acharya et al., 2018; Z. Huang, Wang, Shan, Li, & Chen, 2015), the Gaussian SPD matrix is usually superior to the standard covariance matrix in classification tasks because it simultaneously incorporates the first-order and second-order information of CNN features. The Gaussian covariance matrix can be obtained by transforming $\mathbf{C}_s^{\phi}$ into a single Gaussian model $\mathcal{N}(\boldsymbol{\mu}, \mathbf{C}_s^{\phi})$ represented as:

$$\mathbf{G}_s^{\phi} = \begin{bmatrix} \mathbf{C}_s^{\phi} + \boldsymbol{\mu}\boldsymbol{\mu}^{\top} & \boldsymbol{\mu} \\ \boldsymbol{\mu}^{\top} & 1 \end{bmatrix}, \tag{4.6}$$

where $\boldsymbol{\mu} = \sum_{n=1}^{N} \mathbf{f}_n$. The dimension of the Gaussian covariance matrix $\mathbf{G}_s$ in this case becomes $(C^{'}+1)\times(C^{'}+1)$. The elements of the obtained covariance matrix naturally reside on the Riemannian manifold of the SPD matrix. Since the direct flattening operation will destroy the geometry structure of the formed Riemannian manifold $\mathbf{G}_s^{\phi}$, the logarithmic operation is used to flatten the spatial structure of the Riemannian manifold so that all distance measurements in Euclidean space can be adopted. In addition, to maintain the singularity of $\mathbf{G}_s^{\phi}$, a small ridge is

introduced and added to the Gaussian covariance matrix $\mathbf{G}_s$:

$$(\mathbf{G}_s^{\phi})^{+} = \mathbf{G}_s^{\phi} + \lambda \text{trace}(\mathbf{G}_s^{\phi})\mathbf{I}_g, \tag{4.7}$$

where λ is a hyper-parameter and $\mathbf{I}_g \in \mathbb{R}^{(C^{'}+1)\times(C^{'}+1)}$ denotes the identity matrix. This indeed can be seen as a regularisation operation to transform the symmetric positive semi-definite matrix into a symmetric positive definite matrix.

By reflecting the subtle differences of similar images, the features learned in this way are usually more discriminative than ordinary CNN features, which helps to deal with the nuisance variations of input images, especially the large intra-class variations. As illustrated in Eq.(4.5)-Eq.(4.7), it has shown the flexibility of learning second-order features through the standard covariance matrix or Gaussian covariance matrix. The Gaussian covariance matrices representing different granularity features obtained can be fused by the general concatenation method. However, cascading the vectorised Gaussian matrices will yield a very high-dimensional feature vector, which can result in an exponential increase in computation time and it is usually cannot be implemented in practice. It is recommended to fuse features by averaging the SPD matrix instead of concatenating and this has been expressed as the latter part of the Eq.(4.2). It is worth noting that this not only allows us to capture multi-granularity information, but also preserves the excessive expansion of the feature dimension of $\mathbf{G}^{+}$.

4.2.3 EIG-decomposition Layers

The obtained SPD matrix $\mathbf{G}^{+}$ can be regarded as the feature and directly used to train the classifier. However, if this feature makes proper use of the geometry of the SPD manifold, it can be more distinguishable. To this end, the EIG-decomposition function is considered for decomposing the SPD matrix, especially

in the case of non-linearity. Specifically, the EIG-decomposition function offers an effective way to scale the spectrum of SPD matrix by using appropriate normalisation methods. Since the power of the matrix can be represented by the power of the eigenvalue, the EIG decomposition is expressed as:

$$(\mathbf{G}^{+})_k = f_d^{(k)}\left((\mathbf{G}^{+})_{k-1}\right) = \mathbf{U}_{k-1}\mathbf{F}(\mathbf{\Sigma}_{k-1})\mathbf{U}_{k-1}^{\top}, \tag{4.8}$$

where $\mathbf{F}(\mathbf{\Sigma}_{k-1})$ is the normalised diagonal matrix of eigenvalues and it can be denoted as:

$$\mathbf{F}(\mathbf{\Sigma}_{k-1}) = \begin{cases} \text{diag}\left(\log(\nu_1), \cdots, \log(\nu_c)\right); \\ \text{diag}\left((\nu_1)^{\frac{1}{2}}, \cdots, (\nu_c)^{\frac{1}{2}}\right). \end{cases} \tag{4.9}$$

Here $\text{diag}(\cdot)$ denotes the diagonal operation of the matrix, and $\log(\nu_i)$ is the logarithm of eigenvalues ν_i, where $i = 1, ..., c$ and $c = C^{'} + 1$, arranged in non-increasing order.

As shown in Eq.(4.9), two methods have been introduced to normalise eigenvalues. For SPD matrix, the natural choice is to compute the logarithm of eigenvalues because it succeeds in endowing the Riemannian manifold of SPD matrix with a Lie group structure (Acharya et al., 2018). Accordingly, the flattened Riemannian space allows the computation operations of the Euclidean to be applied in the Log-Euclidean space. Although points in the tangent space can be locally approximated to a flattened SPD manifold, the logarithm of the eigenvalue matrix (Log-E) is usually numerically unstable in the case of non-linearity. The square root of the eigenvalue matrix (Sqrt-E), as a stable alternative solution, has attracted increasing attention.

Furthermore, the Log-E metric requires that eigenvalues to be strictly positive and it will considerably change the magnitudes of eigenvalues, especially it will overstretch the smaller eigenvalues and even reverse the order of the eigenvalues,

which affects the importance of the inherent linear relationship. To avoid these problems, the rectification function that has been introduced in (Acharya et al., 2018) will be employed and written as:

$$\mathbf{R} = \max(\varepsilon \mathbf{I}, \mathbf{\Sigma}_{k-1}), \tag{4.10}$$

where ε is a threshold and $\mathbf{I}$ is an identity matrix. To prevent elements of eigenvalues from being close to non-positive ones, ν_i is replaced by $\mathbf{R}(i,i)$ in the sequel. The functionality of the Eq.(4.10) is similar to the ReLU activation function in the standard neural networks (Glorot et al., 2011), which can be viewed as a non-linear rectification function. However, it is more powerful and ideal in our scheme because it does not produce sparsity like ReLU (Glorot et al., 2011). Then, the diagonal elements can be defined as:

$$\mathbf{R}(i,i) = \begin{cases} \mathbf{\Sigma}_{k-1}(i,i), & \mathbf{\Sigma}_{k-1}(i,i) > \varepsilon; \\ \varepsilon, & \mathbf{\Sigma}_{k-1}(i,i) \leq \varepsilon, \end{cases} \tag{4.11}$$

where $\mathbf{\Sigma}_{k-1} = \text{diag}(\nu_1, \cdots, \nu_c)$ and it can be obtained by the standard EIG function as follows:

$$(\mathbf{G}^{+})_{k-1} = \mathbf{U}_{k-1} \mathbf{\Sigma}_{k-1} \mathbf{U}_{k-1}^{\top}. \tag{4.12}$$

The above rectification layer is specially designed for the Log-E metric to ensure that the normalised eigenvalues to be positive real numbers and then to improve the numerical stability. The Sqrt-E metric has a slight advantage in comparison because square root normalisation allows non-negative eigenvalues. A similar rectification function was introduced in (Acharya et al., 2018). However, it needs to be incorporated into the additional EIG decomposition function in advance, which leads to a large demand for computation costs and makes it time-consuming. Instead, the proxy parameter $\mathbf{R}(i,i)$ is introduced in the proposed method to allow

the EIG decomposition function to run only once.

4.2.4 Back-propagation

Deep learning heavily relies on efficient gradient computation algorithms for back-propagation. However, existing methods (Lin & Maji, 2017; P. Li et al., 2017) usually compute the gradient of the EIG-decomposition function on CPUs because the CUDA platform does not yet support it well. Specifically, the gradient of the EIG-decomposition function approaches infinity when the given matrix is a degenerate one. This implies that one or more of its normalised eigenvalues may be identical. The corresponding eigenvectors can be arbitrary in this situation. To circumvent this problem, it replaces the infinite gradient values with 0, which effectively prevents the gradient computation from being interrupted during back-propagation.

To demonstrate the back-propagation of the introduced algorithm, the method described in (Ionescu et al., 2015) is adopted, which computes the gradient of the general matrix by establishing the corresponding chain rule with first-order Taylor expansion and approximation. Compared with the back-propagation for the standard EIG-decomposition function, it will also provide the gradient calculations for the normalisation function and rectification function. Given the final loss function l, the gradients for the classification layer in Eq.(4.1) can be calculated by $L_c = l \circ f_r$. Let $L^k = L_c \circ f_d^{(l)} \circ f_d^{(l-1)} \circ, ..., \circ f_d^{(1)}$ denote the corresponding gradient in the k-th layer of the EIG-decomposition function. The chain rule can be expressed as:

$$\frac{\partial L^{(k)}\left(\left(\mathbf{G}^{+}\right)_{k-1}, y\right)}{\partial\left(\mathbf{G}^{+}\right)_{k-1}}=\frac{\partial L^{(k+1)}\left(\left(\mathbf{G}^{+}\right)_{k}, y\right)}{\partial\left(\mathbf{G}^{+}\right)_{k}} \frac{\partial f_d^{(k)}\left(\left(\mathbf{G}^{+}\right)_{k-1}\right)}{\partial\left(\mathbf{G}^{+}\right)_{k-1}}, \tag{4.13}$$

where y is the output of the classification layer. $\left(\mathbf{G}^{+}\right)_{k} = f_{d}^{(k)}\left(\left(\mathbf{G}^{+}\right)_{k-1}\right)$ has been previously introduced in Eq.(4.8). Suppose that $\mathscr{F}$ is a function that describes the variations of the present layer to the previous layer in the EIG-decomposition function, which can be written as: $d\left(\mathbf{G}^{+}\right)_{k} = \mathscr{F}\left(d\left(\mathbf{G}^{+}\right)_{k-1}\right)$. The chain rule will become as:

$$\frac{\partial L^{(k)}\left(\left(\mathbf{G}^{+}\right)_{k-1}, y\right)}{\partial\left(\mathbf{G}^{+}\right)_{k-1}} = \mathscr{F}^{*}\left(\frac{\partial L^{(k+1)}\left(\left(\mathbf{G}^{+}\right)_{k}, y\right)}{\partial\left(\mathbf{G}^{+}\right)_{k}}\right), \tag{4.14}$$

where $\mathscr{F}^{*}$ is a non-linear adjoint operator of $\mathscr{F}$ ($i.e.$, $\mathbf{A} : \mathscr{F}(\mathbf{B}) = \mathscr{F}^{*}(\mathbf{A}) : \mathbf{B}$). Specifically, : denotes the matrix inner product in the Euclidean $vec^{'}$d matrix space, which has the property of colon-product and can be written as $\mathbf{A} : \mathbf{B} = \sum_{i,j} \mathbf{A}_{ij}\mathbf{B}_{ij} = \text{trace}(\mathbf{A}^{\top}\mathbf{B})$. As all operations rely on EIG-decomposition functions ($i.e.$, Eq. 4.12), a virtual operation ($i.e.$, $k^{'}$ layer) is introduced as an example. Then, the updated chain rule based on Eq.(4.14) can be written as:

$$\begin{aligned}
&\frac{\partial L^{(k)}((\mathbf{G}^{+})_{k-1}, y)}{\partial(\mathbf{G}^{+})_{k-1}} : d(\mathbf{G}^{+})_{k-1} \\
=&\mathscr{F}^{*}\left(\left(\frac{\partial L^{(k^{'})}}{\partial \mathbf{U}_{k-1}}\right) + \left(\frac{\partial L^{(k^{'})}}{\partial \mathbf{\Sigma}_{k-1}}\right)\right) : d(\mathbf{G}^{+})_{k-1} \\
=&\frac{\partial L^{(k^{'})}}{\partial \mathbf{U}_{k-1}} : \mathscr{F}(d(\mathbf{G}^{+})_{k-1}) + \frac{\partial L^{(k^{'})}}{\partial \mathbf{\Sigma}_{k-1}} : \mathscr{F}(d(\mathbf{G}^{+})_{k-1}) \\
=&\frac{\partial L^{(k^{'})}}{\partial \mathbf{U}} : d\mathbf{U} + \frac{\partial L^{(k^{'})}}{\partial \mathbf{\Sigma}} : d\mathbf{\Sigma}.
\end{aligned} \tag{4.15}$$

It is worth noting that the subscripts of $d\mathbf{U}_{k-1}$ and $d\mathbf{\Sigma}_{k-1}$ in the last line have been removed to improve the readability. Both $d\mathbf{U}$ and $d\mathbf{\Sigma}$ are derived from the variation of the standard EIG function:

$$d(\mathbf{G}^{+})_{k-1} = d\mathbf{U}\mathbf{\Sigma}\mathbf{U}^{\top} + \mathbf{U}d\mathbf{\Sigma}\mathbf{U}^{\top} + \mathbf{U}\mathbf{\Sigma}d\mathbf{U}^{\top}, \tag{4.16}$$

After some rearrangements, $d\mathbf{U}$ and $d\mathbf{\Sigma}$ can be denoted as:

$$\begin{aligned} d\mathbf{U} &= 2\mathbf{U}(\mathbf{Q}^\top \odot (\mathbf{U}^\top d(\mathbf{G}^+)_{k-1}\mathbf{U})_{sym}) \\ d\mathbf{\Sigma} &= (\mathbf{U}^\top d(\mathbf{G}^+)_{k-1}\mathbf{U})_{diag} \end{aligned} \tag{4.17}$$

where $\odot$ denotes the Hadamard product of the matrix. Besides, $\mathbf{M}_{sym} = \frac{1}{2}(\mathbf{M} + \mathbf{M}^\top)$ and $\mathbf{M}_{diag}$ is $\mathbf{M}$ with all off-diagonal elements that are set to 0. Then, $\mathbf{Q}$ can be achieved by:

$$\mathbf{Q}(i,j) = \begin{cases} \frac{1}{\nu_i - \nu_j}, & i \neq j; \\ 0, & i = j. \end{cases} \tag{4.18}$$

Readers are referred to (Ionescu et al., 2015) for more details in terms of deriving Eq.(4.17). The specific partial derivatives of the loss function can be derived by plugging Eq.(4.17) into Eq.(4.15). In addition, the property of the inner product of the matrix introduced previously can naturally be expressed as:

$$\begin{aligned} &\frac{\partial L^{(k)}}{\partial (\mathbf{G}^+)_{k-1}} = \frac{\partial L^{(k)}\left((\mathbf{G}^+)_{k-1}, y\right)}{\partial (\mathbf{G}^+)_{k-1}} \\ =&\mathbf{U}\left(\left(\mathbf{Q}^\top \odot \left(\mathbf{U}^\top \frac{\partial L^{(k')}}{\partial \mathbf{U}}\right)\right) + \left(\frac{\partial L^{(k')}}{\partial \mathbf{\Sigma}}\right)_{diag}\right)\mathbf{U}^\top, \end{aligned} \tag{4.19}$$

where $\frac{\partial L^{(k')}}{\partial \mathbf{U}}$ and $\frac{\partial L^{(k')}}{\partial \mathbf{\Sigma}}$ can be calculated by employing a strategy similar to $\frac{\partial L^{(k)}}{\partial (\mathbf{G}^+)_{k-1}}$ described in Eq.(4.15). Then, it can derive the partial derivatives of $\frac{\partial L^{(k')}}{\partial \mathbf{U}}$ and $\frac{\partial L^{(k')}}{\partial \mathbf{\Sigma}}$ as:

$$d(\mathbf{G}^+)_k = 2(d\mathbf{U}g(\mathbf{\Sigma})\mathbf{U}^\top)_{sym} + \mathbf{U}g'(\mathbf{\Sigma})d\mathbf{\Sigma}\mathbf{U}^\top. \tag{4.20}$$

Following the chain rule introduced in Eq.(4.15), the derivatives of $\frac{\partial L^{(k')}}{\partial \mathbf{U}}$ and

$\frac{\partial L^{(k^{'})}}{\partial \mathbf{\Sigma}}$ can be obtained by:

$$\begin{aligned}\frac{\partial L^{(k^{'})}}{\partial \mathbf{U}} &= 2\left(\frac{\partial L^{(k+1)}}{\partial (\mathbf{G}^{+})_k}\right)_{sym}\mathbf{U}g(\mathbf{\Sigma}) \\ \frac{\partial L^{(k^{'})}}{\partial \mathbf{\Sigma}} &= g^{'}(\mathbf{\Sigma})\mathbf{U}^{\top}\left(\frac{\partial L^{(k+1)}}{\partial (\mathbf{G}^{+})_k}\right)\mathbf{U},\end{aligned} \tag{4.21}$$

The partial derivatives of $\frac{\partial L^{(k^{'})}}{\partial \mathbf{U}}$ and $\frac{\partial L^{(k^{'})}}{\partial \mathbf{\Sigma}}$ in Log-E form can be written as:

$$\begin{aligned}\frac{\partial L^{(k^{'})}}{\partial \mathbf{U}} &= 2\left(\frac{\partial L^{(k+1)}}{\partial (\mathbf{G}^{+})_k}\right)_{sym}\mathbf{U}\log(\mathbf{R}) \\ \frac{\partial L^{(k^{'})}}{\partial \mathbf{\Sigma}} &= \mathrm{diag}(\nu_1^{-1},...,\nu_c^{-1})\mathbf{U}^{\top}\left(\frac{\partial L^{(k+1)}}{\partial (\mathbf{G}^{+})_k}\right)\mathbf{U},\end{aligned} \tag{4.22}$$

similarly, the partial derivatives in Sqrt-E form can be obtained by:

$$\begin{aligned}\frac{\partial L^{(k^{'})}}{\partial \mathbf{U}} &= 2\left(\frac{\partial L^{(k+1)}}{\partial (\mathbf{G}^{+})_k}\right)_{sym}\mathbf{U}(\mathbf{R})^{\frac{1}{2}} \\ \frac{\partial L^{(k^{'})}}{\partial \mathbf{\Sigma}} &= \frac{1}{2\sqrt{\nu_1}}\left(\mathrm{diag}(\nu_1^{-\frac{1}{2}},...,\nu_c^{-\frac{1}{2}})\mathbf{U}^{\top}\left(\frac{\partial L^{(k+1)}}{\partial (\mathbf{G}^{+})_k}\right)\mathbf{U}\right) \\ &\quad - \mathrm{diag}\left(\frac{1}{2\sqrt{\nu_1}}\mathrm{trace}\left((\mathbf{G}^{+})_k\frac{\partial L^{(k+1)}}{\partial (\mathbf{G}^{+})_k}\right),0,\ldots,0\right),\end{aligned} \tag{4.23}$$

where $\mathbf{R}$ is the resulting matrix introduced in Eq.(4.10). Specifically, $g^{'}(\mathbf{\Sigma})$ in $g^{'}(\mathbf{\Sigma})\mathbf{U}^{\top}\left(\frac{\partial L^{(k+1)}}{\partial (\mathbf{G}^{+})_k}\right)\mathbf{U}$ will be replaced by $g^{'}(\mathbf{R})\mathbf{U}^{\top}\left(\frac{\partial L^{(k+1)}}{\partial (\mathbf{G}^{+})_k}\right)\mathbf{U}$ in the sequel. The corresponding gradient of $\mathbf{R}$ can be formed by:

$$\mathbf{R}(i,i) = \begin{cases} 1, & \mathbf{\Sigma}_{k-1}(i,i) > \varepsilon; \\ 0, & \mathbf{\Sigma}_{k-1}(i,i) \leq \varepsilon. \end{cases} \tag{4.24}$$

Once the partial derivatives of $\frac{\partial L^{(k^{'})}}{\partial \mathbf{U}}$ and $\frac{\partial L^{(k^{'})}}{\partial \mathbf{\Sigma}}$ have been obtained, they can be

plugged into Eq.(4.19), resulting in the back-propagation of the Riemannian SPD matrices under logarithm normalisation and more robust square-root normalisation.

When the partial derivative of $\frac{\partial L^{(k)}}{\partial (\mathbf{G}^{+})_{k-1}}$ has been obtained, it can be used to compute the gradient of learning canonical transformation. $\nabla f_c\left((\mathbf{G}_s^{\phi})^{+}\right)$ is introduced to denote the gradient of the feature $f_c\left((\mathbf{G}_s^{\phi})^{+}\right)$. Then, it can derive the gradient of $\frac{d\mathbf{G}_s^{+}}{df_c\left((\mathbf{G}_s^{\phi})^{+}\right)}$ with:

$$\frac{d\mathbf{G}_s^{+}}{df_c\left((\mathbf{G}_s^{\phi})^{+}\right)} = \nabla f_c\left((\mathbf{G}_s^{\phi})^{+}\right), \tag{4.25}$$

where $\phi = \underset{\phi \in \Phi}{\operatorname{argmax}}\ f_c\left((\mathbf{G}_s^{\phi})^{+}\right)$ is the optimal appearance ϕ with respect to input $\mathcal{X}_s$ at a specific granularity.

Finally, it can derive the gradient of the loss function for matrix $\mathbf{F}_s^{\Phi}$ that has been reshaped by CNN features in Eq.(4.5). Specifically, it can be expressed as:

$$\frac{\partial L^{(k)}}{\partial \mathbf{F}_s^{\Phi}} = \left(\frac{\partial L^{(k)}}{\partial \mathbf{F}_s^{\Phi}} + \left(\frac{\partial L^{(k)}}{\partial \mathbf{F}_s^{\Phi}}\right)^{\top}\right)\bar{\mathbf{I}}\mathbf{F}_s^{\Phi}. \tag{4.26}$$

The above formulations have shown the back-propagation of the proposed method in detail. For clarity, it can derive the gradients of EIG-decomposition layers, SPD matrices layers and canonical appearance pooling layers in sequence. The gradient of the entire framework can be calculated by cascading these layers together because the rest of the MG-CAP model (such as the extraction of CNN features) is fully differentiable.

4.3 Experiments

4.3.1 Implementation Details

The proposed framework was implemented using the VGGNet-16 (Simonyan & Zisserman, 2014) architecture, which has been pre-trained on the large-scale ImageNet dataset. During training, data augmentation techniques were used to avoid overfitting. These include randomly cropping 224 $\times$ 224 patches from 256 $\times$ 256 images, followed by horizontal flipping. The generated patch image will be transformed according to the predefined transformation rules. Then, the zero-padding method will be used to fill the transformed image to 317 $\times$ 317 pixels to avoid that the transformed image will not exceed the boundary of the original image when the rotation angle is not a multiple of 90 degrees. After the transformation is completed, all transformed images will be adjusted to a size of 224$\times$224 using the bilinear interpolation, so that the transformed images can be fed into the Siamese architecture for feature extraction. It retains the parameters that appeared before the last non-activated convolutional features of VGGNet-16 (Simonyan & Zisserman, 2014) (i.e., conv5_3).

It initially trains the classification layer with a learning rate of 0.1 and then fine-tunes the entire network with a small learning rate of 10^{-3}. The learning rate is annealed by 0.15 in every 30 epochs during the warm-up stage and then decayed after every 3 epochs during the fine-tuning stage. The batch size is 12 for the experiments of 3 granularities with 12 different transformations. The weight decay rate is 5×10^{-4}. The framework is optimised using the momentum optimiser with a constant momentum factor of 0.9. It is worth noting that the datasets has been randomly split ten times for training and test. Meanwhile, it reports the corresponding mean and the standard deviation of the overall accuracy. The value of λ

Table 4.1: Comparison of the overall accuracy and standard deviation obtained by the MG-CAP model and previous work on NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017). H.F., U.L.F., D.L.F. and T.R. are abbreviations for handcrafted feature, unsupervised learning feature, deep learning feature and training ratio, respectively.

	Method	NWPU-RESISC45	
		T.R.=10%	T.R.=20%
H.F.	GIST (Cheng, Han, & Lu, 2017)	15.90±0.23	17.88±0.22
	LBP (Cheng, Han, & Lu, 2017)	19.20±0.41	21.74±0.18
	Colour Histogram (Cheng, Han, & Lu, 2017)	24.84±0.22	27.52±0.14
U.L.F.	BoVW+SPM (Cheng, Han, & Lu, 2017)	27.83±0.61	32.96±0.47
	LLC (Cheng, Han, & Lu, 2017)	38.81±0.23	40.03±0.34
	BoVW (Cheng, Han, & Lu, 2017)	41.72±0.21	44.97±0.28
D.L.F.	AlexNet (Cheng et al., 2018)	76.69±0.21	79.85±0.13
	GoogLeNet (Cheng et al., 2018)	76.47±0.18	79.79±0.15
	VGGNet-16 (Cheng et al., 2018)	76.19±0.38	78.48±0.26
	AlexNet+BoVW (Cheng, Li, et al., 2017)	55.22±0.39	59.22±0.18
	GoogLeNet+BoVW (Cheng, Li, et al., 2017)	78.92±0.17	80.97±0.17
	VGGNet-16+BoVW (Cheng, Li, et al., 2017)	82.65±0.31	84.32±0.17
	MG-CAP (Bilinear)	89.42±0.19	91.72±0.16
	MG-CAP (Log-E)	88.35±0.23	90.94±0.20
	MG-CAP (Sqrt-E)	**90.83±0.12**	**92.95±0.13**

in Eq.(4.7) is empirically set to 1×10^{-4}. The threshold parameter ε in Eq.(4.10) is used to rectify the value of eigenvalues and was originally introduced in (Acharya et al., 2018) for Log-E normalisation. It has been applied to both Sqrt-E and Log-E in our experiments to clip the eigenvalue into $[1 \times 10^{-5}, 1 \times 10^{5}]$. Besides, the framework is implemented using the GPU version of TensorFlow 1.0.

4.3.2 Experimental Results and Comparison

The results of the proposed MG-CAP network on the most challenging dataset (Cheng, Han, & Lu, 2017) are compared with several benchmark methods. As shown in TABLE 4.1, the Colour histograms method achieves the best classification results among the listed handcrafted feature-based methods. Specifically,

the Colour histogram feature performs better than the LBP feature and the global GIST feature under the two different datasets partitions. Furthermore, unsupervised feature learning-based methods achieve higher accuracy than all of the listed handcrafted feature-based methods. An algorithm combining BoVW and SPM was proposed to incorporate more spatial information from images, but it only achieves accuracies of27.83% and 32.96% (Cheng, Han, & Lu, 2017). The LLC is slightly better than BoVW+SPM, but still falls short when compared with the BoVW algorithm (i.e., about 3% and 5% differences). Deep learning-based methods demonstrate their superior performances and overshadow both handcrafted and unsupervised based feature learning methods. To be precise, with a linear SVM classifier, transferred neural networks (Cheng et al., 2018) achieves an accuracy of about 76% for the 10% training split, while the accuracy is further increased by about 3% for the 20% training ratio. Furthermore, the combination of VGGNet-16 and BoVW achieves the best performance among all registered methods. However, the combination of AlexNet and BoVW only produces accuracies of 55.22%$\pm$0.39 and 59.22%$\pm$0.18 (Cheng, Li, et al., 2017), which is surprisingly lower than other deep learning-based algorithms and even lower than the transferred AlexNet (Cheng et al., 2018).

According to the main architecture of the MG-CAP network, three variants are proposed, including the original bilinear pooling, the logarithm of the eigenvalue (Log-E) and the square root of the eigenvalue (Sqrt-E). It can be seen from TABLE 4.1 that all the variants of the MG-CAP network are much more accurate than the previous benchmark method. The Log-E based MG-CAP model achieves more than double the accuracy of the BoVW method (i.e., 88.35% versus 41.72% under the training ratio of 10%, and 90.94% versus 44.97% under the training ratio of 20%). Interestingly, the bilinear pooling performs better than the Log-E based method. This is because the logarithm of eigenvalue has the potential to consider-

Table 4.2: Comparison of the overall accuracy and standard deviation of the proposed MG-CAP model with baseline methods and state-of-the-art methods, where T.R. is the abbreviation of Training Ratio.

Deep Learning based Methods	NWPU-RESISC45		AID		UC-Merced
	T.R.=10%	T.R.=20%	T.R.=20%	T.R.=50%	T.R.=80%
AlexNet+SVM (Cheng et al., 2018)	81.22±0.19	85.16±0.18	84.23±0.10	93.51±0.10	94.42±0.10
GoogLeNet+SVM (Cheng et al., 2018)	82.57±0.12	86.02±0.18	87.51±0.11	95.27±0.10	96.82±0.20
VGGNet-16+SVM (Cheng et al., 2018)	87.15±0.45	90.36±0.18	89.33±0.23	96.04±0.13	97.14±0.10
MSCP with AlexNet (N. He et al., 2018)	81.70±0.23	85.58±0.16	88.99±0.38	92.36±0.21	97.29±0.63
MSCP+MRA with AlexNet (N. He et al., 2018)	88.31±0.23	87.05±0.23	90.65±0.19	94.11±0.15	97.32±0.52
MSCP with VGGNet-16 (N. He et al., 2018)	85.33±0.17	88.93±0.14	91.52±0.21	94.42±0.17	98.36±0.58
MSCP+MRA with VGGNet-16 (N. He et al., 2018)	88.07±0.18	90.81±0.13	92.21±0.17	96.56±0.18	98.40±0.34
DCNN with AlexNet (Cheng et al., 2018)	85.56±0.20	87.24±0.12	85.62±0.10	94.47±0.10	96.67+0.10
DCNN with GoogLeNet (Cheng et al., 2018)	86.89±0.10	90.49±0.15	88.79±0.10	96.22±0.10	97.07+0.12
DCNN with VGGNet-16 (Cheng et al., 2018)	89.22±0.50	91.89±0.22	90.82±0.16	**96.89**±0.10	98.93±0.10
RTN in Chapter 3 (Z. Chen et al., 2018)	89.53±0.21	92.20±0.34	92.75±0.21	95.09±0.16	98.33±0.71
the proposed MG-CAP with Bilinear	89.42±0.19	91.72±0.16	92.11±0.15	95.14±0.12	98.60±0.26
the proposed MG-CAP with Log-E	88.35±0.23	90.94±0.20	90.17±0.19	94.85±0.16	98.45±0.12
the proposed MG-CAP with Sqrt-E	**90.83**±0.12	**92.95**±0.13	**93.34**±0.18	96.12±0.12	**99.0**±0.10

ably change its magnitude, especially for smaller eigenvalues (P. Li et al., 2017). This change will reverse the significances of eigenvalues, which is detrimental to performance. The Sqrt-E based MG-CAP can avoid this problem reasonably and obtain the best classification results.

To adequately evaluate the effectivenesses of the MG-CAP network, it has also been compared with state-of-the-art approaches. The results reported in TABLE 4.1 and TABLE 4.2 are obtained by initialising three granularities and 12 rotations per granularity. From TABLE 4.2, the VGGNet-16 (Simonyan & Zisserman, 2014) based architecture usually achieves a more desirable classification accuracy than GoogLeNet (Szegedy et al., 2015) and AlexNet (Krizhevsky et al., 2012). The Log-E based MG-CAP method obtains 88.35% accuracy when using 10% of the training samples, surpassing all of the MSCP based methods (N. He et al., 2018). When using Bilinear pooling, the MG-CAP model can achieve an higher accuracy than the metric learning-based Discriminative CNNs (DCNN) (Cheng et al., 2018) and is very close to RTN (Z. Chen et al., 2018) in Chapter 3. The Sqrt-E based MG-CAP algorithm obtains 90.64% and 92.75% classification accuracy under the different cases, which is the best results so far on NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017). On AID dataset (Xia, Hu, et al., 2017), the Log-E based MG-CAP model also performs very competitively. For example, the classification accuracy exceeds all of the listed architectures with the linear SVM methods (Cheng et al., 2018). When using the VGGNet-16 architecture (Simonyan & Zisserman, 2014), the results obtained are slightly lower than DCNN (Cheng et al., 2018) and significantly lower than MSCP (N. He et al., 2018) and RTN (Z. Chen et al., 2018) in Chapter 3. Surprisingly, MSCP with MRA (N. He et al., 2018) achieved more reliable results than DCNN (Cheng et al., 2018) on AID dataset under the training ratio of 20%. However, MSCP (N. He et al., 2018) cannot be trained in an end-to-end manner. Again, the MG-CAP

model with Sqrt-E achieves the best classification accuracy on AID dataset under the training ratio of 20%. Specifically, it obtains the accuracy of 93.34%, with relative gains of 2.52% and 1.1% compared with DCNN (Cheng et al., 2018) and RTN (Z. Chen et al., 2018) in Chapter 3. Although DCNN (Cheng et al., 2018) performs slightly better than the proposed algorithm under the training ratio of 50%, the Sqrt-E based MG-CAP model exceeds all linear SVM based methods and obtains competitive performance to MSCP (N. He et al., 2018). The UC-Merced dataset (Y. Yang & Newsam, 2010) contains 21 categories, which is comparatively less than other datasets. Using a large number of training samples, all of the listed deep learning approaches can achieve very similar results. For example, the reported accuracy of the fine-tuned VGGNet-16 (Simonyan & Zisserman, 2014) model is 97.14%, which is on a par with the GoogLeNet based DCNN architecture (Cheng et al., 2018). In addition, the Sqrt-E based MG-CAP model can achieve 99.0% classification accuracy, which is the best result among all the compared methods.

The confusion matrix is a powerful evaluation method that can show the category-level performance of the algorithm. A confusion matrix (obtained by using the Sqrt-E based MG-CAP model) is randomly selected from the experiments conducted in five different scenarios as the display. As shown in Figure 4.3-Figure 4.7, it is not difficult to see that the darkest colour blocks are displayed on the diagonal of all confusion matrices. The appearance of this phenomenon means that most images can be accurately classified into their respective categories by the proposed algorithm. Among all confusion matrices, the sparsest one is the confusion matrix displayed on UC-Merced dataset (Y. Yang & Newsam, 2010).

From Figure 4.7, the proposed Sqrt-E based MG-CAP can correctly classify most test images. Specifically, 9% and 5% of images in the **Dense residential** category were misidentified as the **Medium residential** and **Mobile home park**. On AID

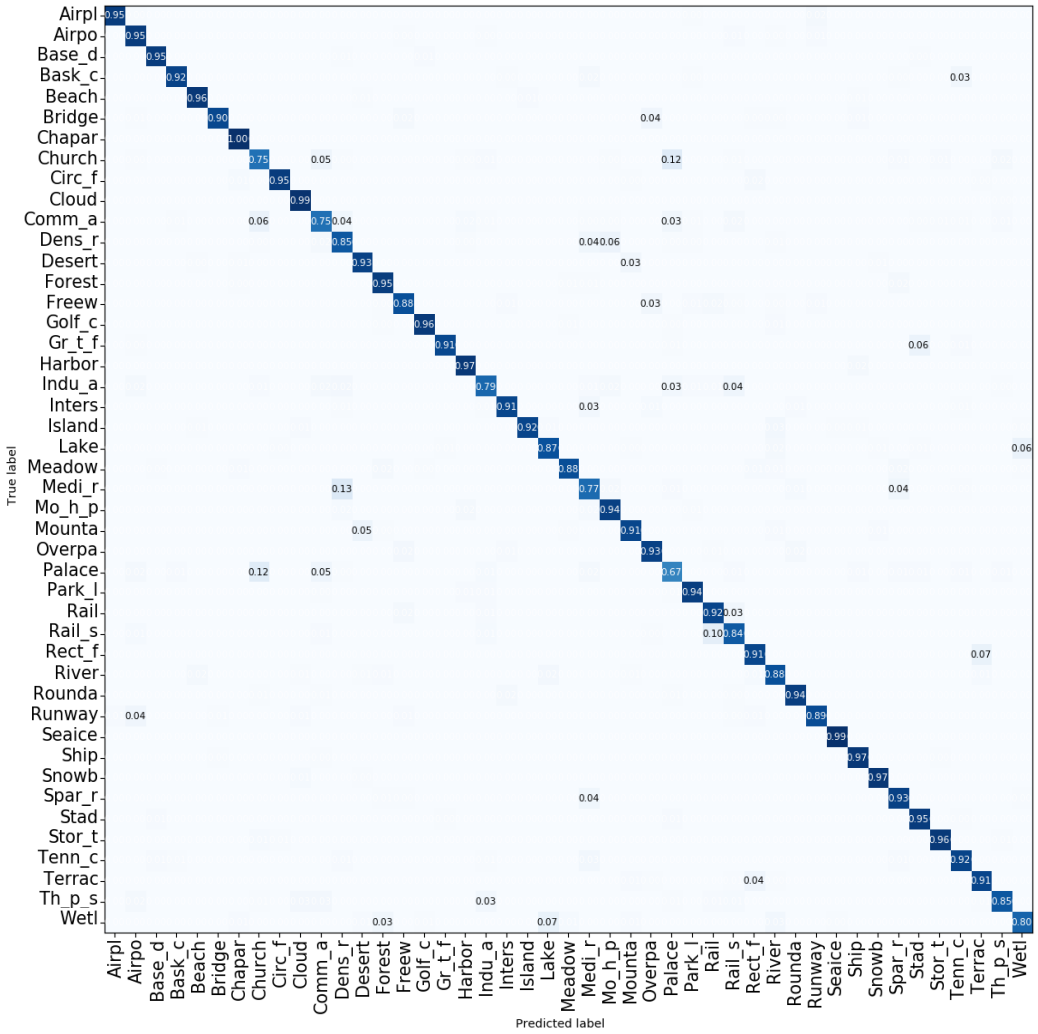


Figure 4.3: The confusion matrix on NWPU-RESISC45 dataset under the training ratio of 10%. For the sake of clarity, values less than 0.03% are omitted.

dataset (Xia, Hu, et al., 2017), when only 20% of the training samples are available (shown in Figure 4.5), the classification accuracy of 27 categories (totally 30 categories) can reach more than 80%. When using half the amount of data to train the model (in Figure 4.6), the accuracy of all categories exceeds 80%, and the lowest classification accuracy rate is 83%. For example, 3% of the images in the **Square** category were misidentified as **Resort** and **School**. This is because there are many areas in **Resort** and **School** images that appear to be square. Furthermore, 9% of

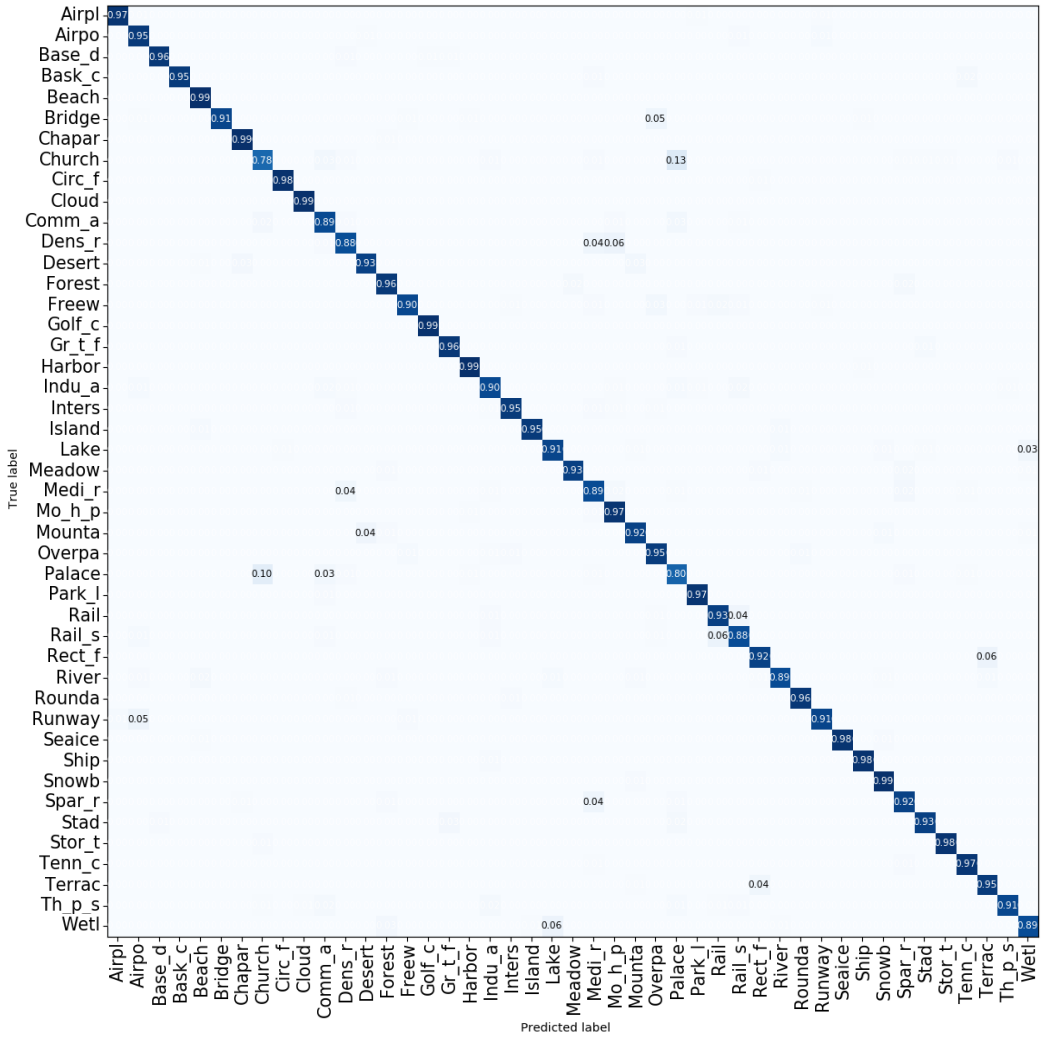


Figure 4.4: The confusion matrix on NWPU-RESISC45 dataset under the training ratio of 20%. For the sake of clarity, values less than 0.03% are omitted.

Resort images have been misclassified into **Park** category due to visually similar areas. On the most challenging NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017), using 10% of the training samples (shown in Figure 4.3) , 40 out of all 45 categories have an accuracy higher than 80%. Figure 4.4 shows the confusion matrix obtained by using 20% of the training samples of NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017). As can be found that 36 out of all 45 categories have an accuracy higher than 90%. For confusing categories like **Palace**, the accu-

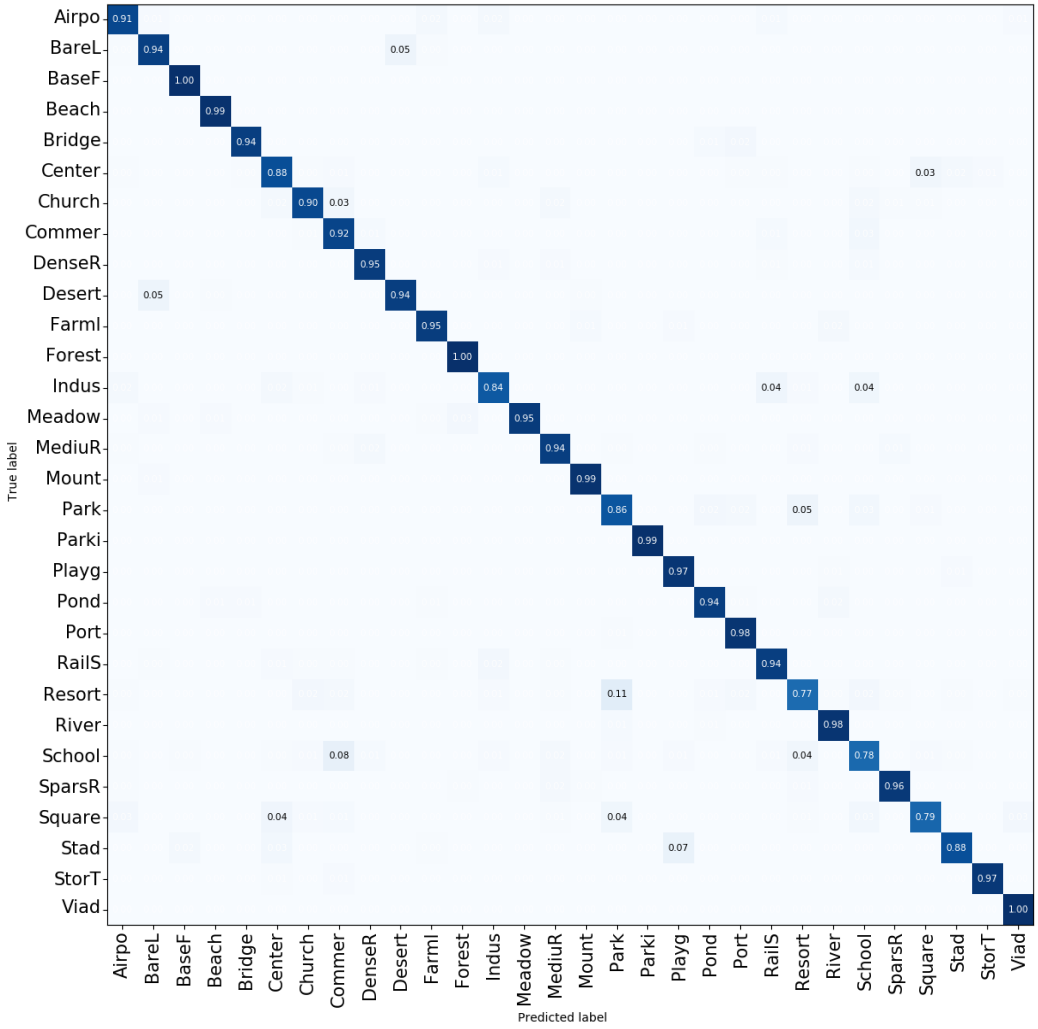


Figure 4.5: The confusion matrix on AID dataset under the training ratio of 20%. For the sake of clarity, values less than 0.03% are omitted.

racy rate of the proposed MG-CAP model reached 73%, which is 21% and 9% higher than the transferred VGGNet-D and the fine-tuned VGGNet-D (Cheng, Li, et al., 2017), respectively. Furthermore, the transferred VGGNet-D (Cheng, Li, et al., 2017) can only achieve the accuracy of 57% on the **Tennis court** category while the proposed MG-CAP method obtains the accuracy of 96%. The class of **Railway station** is easily confused by **Rail** because they may contain one or more similar objects or texture information. The Sqrt-E based MG-CAP model

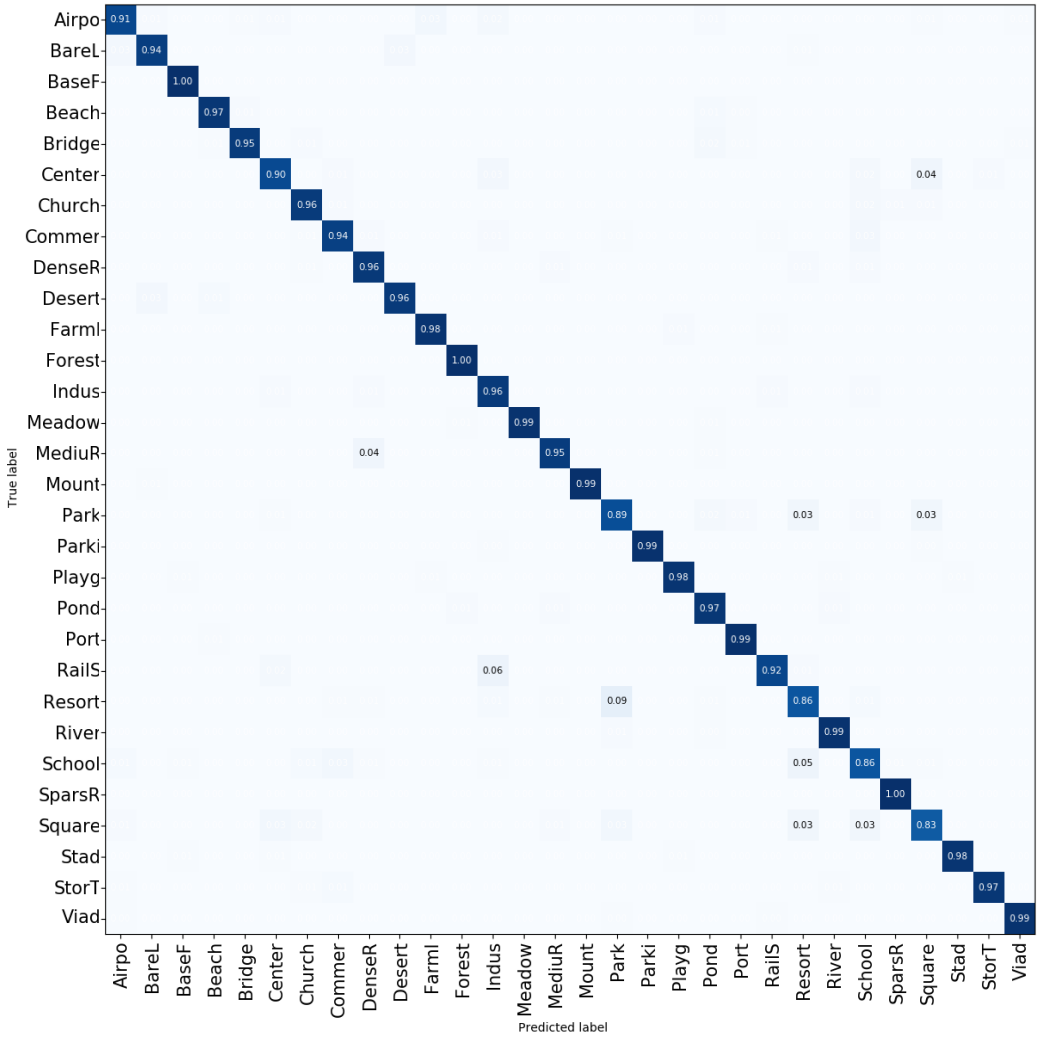


Figure 4.6: The confusion matrix on AID dataset under the training ratio of 50%. For the sake of clarity, values less than 0.03% are omitted.

can achieve 88% accuracy on railway station class while the fine-tuned VGGNet-D (Cheng, Li, et al., 2017) can only obtain 75%, which is a significant increase of 13%. Through the above comparison, it can be confirmed that the proposed MG-CAP model is very capable of distinguishing categories that are easy to be confused in the classification of remote sensing scene images.

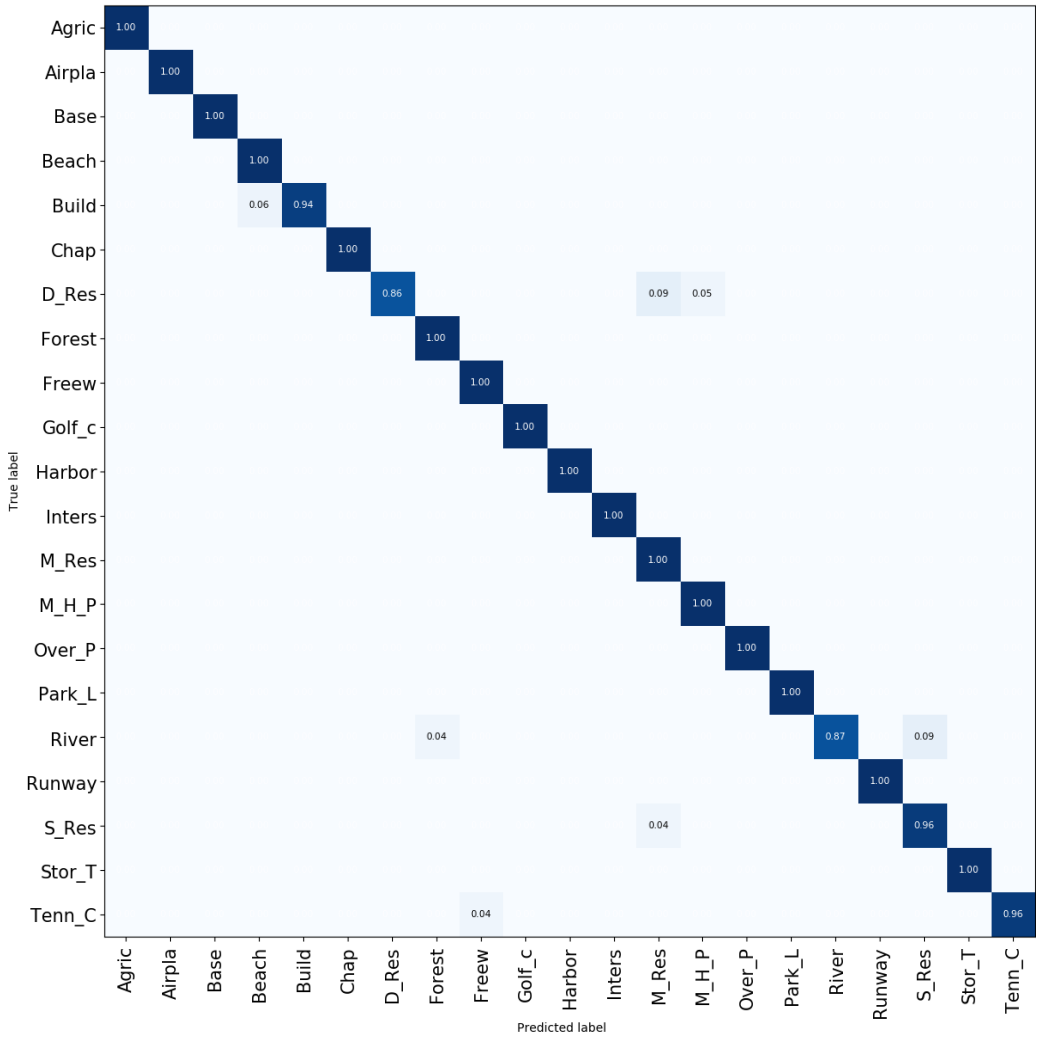


Figure 4.7: The confusion matrix on UC-Merced dataset under the training ratio of 80%. For the sake of clarity, values less than 0.03% are omitted.

4.3.3 Ablation Studies

4.3.3.1 Effect of Granularity

It is not difficult to find that different granularities are dedicated to discovering different response areas in a given image, which will have different degrees of impact on the final result. Increasing the number of granularities may help improve

Table 4.3: Comparison of accuracy obtained under different granularities when using a training ratio of 10% on NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017).

Granularities	G.1	G.2	G.3	G.1+G.2	G.2+G.3	G.1+G.2+G.3
MG-CAP (Log-E)	85.50	86.22	85.79	87.13	86.57	88.45
MG-CAP (Sqrt-E)	88.63	89.05	87.84	90.17	89.82	90.95

accuracy, but it will also take up more memory on the PC, so it is very necessary to weigh the gains and losses between the two. Therefore, it designed an ablation study for the number and combination of granularities and presented the results in Table 4.3. It can be seen that the best results are obtained by combining three different granularities. For an individual granularity, the second granularity can achieve the best classification accuracy, while the third granularity performs the worst. The result of combining the first and second granularities is higher than the result of combining the second and third granularities (i.e., 87.13% versus 86.57% with Log-E, 90.17% versus 89.82% with Sqrt-E). These results indicate that incorporating finer granularity can improve classification accuracy, but excessively fine granularity may harm performance.

4.3.3.2 Impact of Transformations

Figure 4.8 reflects how the number of transformations affects the final classification accuracy of the MG-CAP model based on Log-E and Sqrt-E. It can be seen that the classification result improvements as the number of transformations increases. It is worth noting that the accuracy is improved by about 3% by only rotating the patch image of each granularity three times. However, as the number of transformations continues to increase, the growth rate of classification accuracy has become relatively small. Since the rise in the number of transformations will greatly increase the memory burden, in view of the results shown in Figure 4.8,

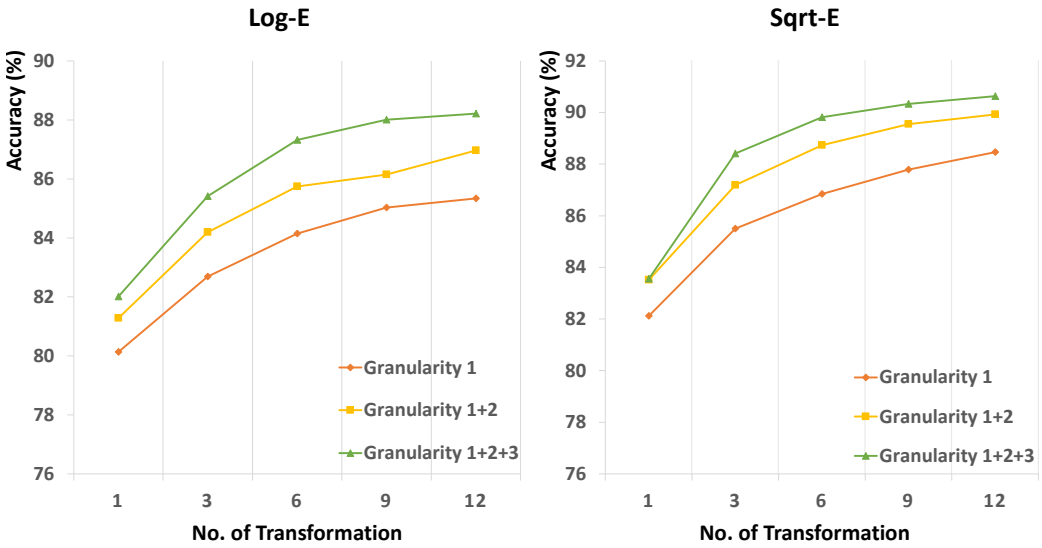


Figure 4.8: Classification accuracy using different numbers of transformations.

the number of transformation of all experiments is finally set to 12.

4.3.4 Qualitative Visualisation & Analysis

In addition to improving accuracy, consideration is also paid to the interpretability of the proposed model. There exist two ways to solve this problem. On the one hand, the canonical appearance can be naturally derived from Eq.(4.4) and the corresponding derivative of Eq.(4.25). Concretely, the optimal transformation for any granularity can be obtained by: $\phi = \underset{\phi \in \Phi}{\operatorname{argmax}} f_c((\mathbf{G}_s^{\phi})^+)$. On the other hand, Grad-Cam (Selvaraju et al., 2017) is an off-the-shelf algorithm for displaying the attention heatmap of an image, which can be used to visualise the most discriminative parts of test images.

It randomly choose several test images from NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017) to show the effectiveness of the MG-CAP model in learning to the canonical appearances and the discriminative features. In Figure 4.9, it shows that the MG-CAP model tends to orient visually similar images or image

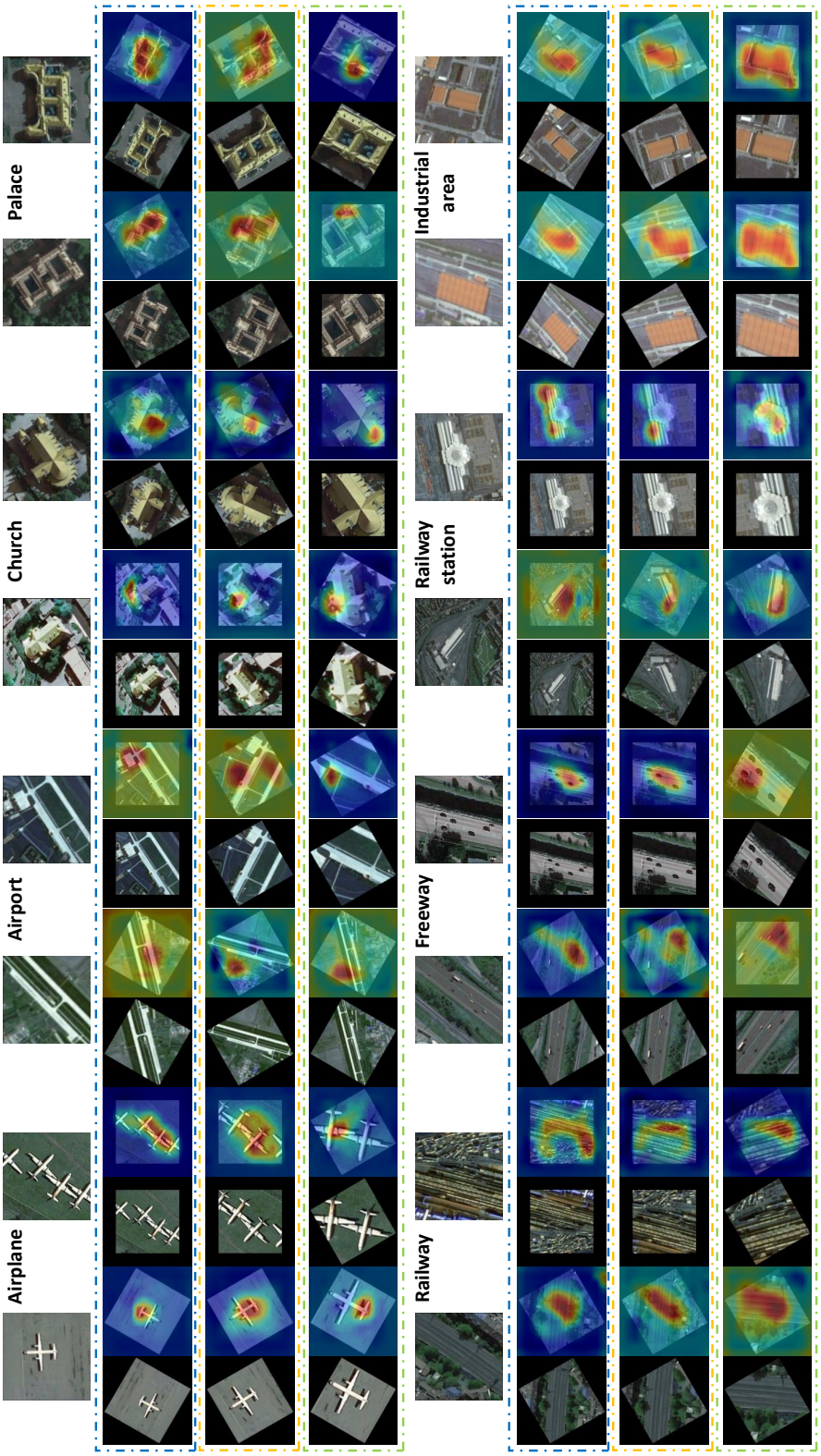


Figure 4.9: Visualisation example of the MG-CAP model on NWPU-RESISC45 (Cheng, Han, & Lu, 2017), where the blue, yellow and green dashed lines denote the 1^{st}, 2^{nd} and 3^{rd} granularity, respectively. (Best seen in colour)

Table 4.4: Comparison of complexity and inference time of the models, where n denotes the number of streams. The two sides of the backslash indicate the inference time of the model on the CPU and GPU respectively.

	Model Complexity	#Params (MB)	Inference Time (sec/img)
RTN (Z. Chen et al., 2018) (in Chapter 3)	*O(n(LocNet+BilinearVGG))*	68.69	0.434/0.073
MG-CAP (Norm-E)	*O(n(CovarianceVGG))*	55.99	1.837/0.216

regions in approximately the same direction, and vice versa. For example, canonical appearances learned at different levels of granularity have almost the same directions for *Church* and *Palace* images. Furthermore, the canonical appearance has changed at the third granularity of the *Palace* images. The reason for this problem is that the main object only partially presents in this granularity. Besides, it is worth noting that the canonical transformation has almost no change in some images, such as *Railway*, *Freeway* and *Railway station*. This is because the texture information is visually similar for different granularities. However, it is worth noting that due to the change of the patch image content, the corresponding attention heatmap can show the difference between different granularities.

The time complexity of the algorithm is another aspect that needs to be evaluated and compared. In particular, it reproduced the recently proposed multi-stream-based RTN model (Z. Chen et al., 2018) (in Chapter 3) and appended it to the comparison. For a fair comparison, experiments are conducted using a PC with a 6-core Intel® Core™ i7-9800X@3.80 GHz CPU and a GeForce RTX 2080Ti GPU. From TABLE 4.4, it can be seen that the model complexity of RTN (Z. Chen et al., 2018) (in Chapter 3) is *O(n(LocNet+BilinearVGG))*, which is more complicated than the proposed MG-CAP model. Specifically, RTN (Z. Chen et al., 2018) (in Chapter 3) needs localisation networks to be recursively applied in order to predict the transformation parameters. In terms of model parameters, the

MG-CAP based method requires 55.99 MB of memory, while RTN (Z. Chen et al., 2018) (in Chapter 3) takes up an extra 12.7 MB. Although RTN (Z. Chen et al., 2018) (in Chapter 3) presents a shorter inference time on the CPU, the GPU-based MG-CAP can make predictions in only 0.216 seconds, which is very close to the RTN model (Z. Chen et al., 2018) (in Chapter 3). Since the MG-CAP model implements a stable GPU-supported version of the EIG decomposition function, its inference time has been greatly reduced. In addition, by learning the compact representation of the Gaussian covariance matrix or using a more powerful GPU, the cost of the matrix decomposition function can be further reduced.

4.4 Conclusion

In this chapter, a novel MG-CAP method has been introduced to solve the large visual-semantic discrepancy and variation problems in RSSC tasks. The learning model is devised in a multi-granularity manner to mine the latent ontological structures of datasets. For each specific granularity, it will find distinguishing features corresponding to the canonical appearance of the cropped image. Common CNN features are successively summarised into a matrix and a covariance matrix and finally converted into a Gaussian matrix. Through flexible log-E and sqrt-E normalised EIG decomposition function, the discriminative ability of the Gaussian matrix can be further improved. More importantly, it presents solutions that enable the EIG-decomposition function to be well supported by GPU acceleration and can train the entire framework in an end-to-end manner. Although MG-CAP meets the expectations of using GPU to accelerate training and reduces the amount of parameters by adopting Siamese architecture, the high-dimensional space generated by multi-granular vectorised second-order features still makes the model have a certain computational burden. If the Siamese-style CNN architec-

ture of learning multiple instances is effective, is it worth attempting to reduce the number of granularities and instead seek a more accurate method to measure the distance of samples in the high-dimensional feature space to improve the classification accuracy? In the next chapter, it will introduce another algorithm to examine this hypothesis.

5 | Covariance Feature Embedding

5.1 Introduction

The previous two chapters (Chapter 3 and Chapter 4) explored methods to expand the model's ability to incorporate prior knowledge by introducing a plethora of transformations at the input of deep convolutional neural networks. The essence of these two models is to expand the range of the input data distribution by introducing unknown transformations, and then converge the models to the most desired transformation represented by the minimum loss through the constraints of the objective function. The discriminative power of the extracted features is also enhanced in this case, especially the second-order statistical features are introduced to replace the commonly used CNN features. If the discrimination of features is closely associated to the classification performance of the model, how to improve the discriminative power of features to a greater extent is worthy of in-depth study.

The latest success of discriminatory metric learning confirms that in addition to incorporating the prior knowledge of the model and designing feature extraction schemes, appropriate metric methods are also crucial to the classification results. The purpose of metric learning is to learn a similarity function (a.k.a. distance function). Prevailing deep metric learning usually uses neural networks to automatically extract distinguishing features $\mathbf{x}_1$, $\mathbf{x}_2$, and then simple distance metrics, such as Euclidean distance $\|\mathbf{x}_1 - \mathbf{x}_2\|_2$. For example, the ordinarily used softmax loss towards encouraging well-separated features to have bigger magnitudes, which limits its discrimination ability (F. Wang, Xiang, Cheng, & Yuille, 2017). To cope with this problem, Wen et al. pioneered the centre loss which penalises the distance between deep learning features and their corresponding cluster cen-

tres in Euclidean space to realise intra-class compactness (Wen, Zhang, Li, & Qiao, 2016). Liu et al. (W. Liu et al., 2017) introduced the Sphereface loss which can be used together with the standard softmax loss to train the model, but the precipitous change in the target logit hinders convergence. To relieve the need for joint supervision from the softmax loss, (H. Wang et al., 2018) and (F. Wang, Cheng, Liu, & Liu, 2018) directly added cosine margin penalty to the target logit. Recently, an additive angular margin loss called ArcFace was proposed by (Deng, Guo, Xue, & Zafeiriou, 2019), which measures the geodesic distance on the normalised hypersphere to simultaneously enhance the intra-class compactness and the inter-class discrepancy.

The above-mentioned metric learning methods impose appropriate constraints to increase the inter-class distance while tightening the intra-class distance, which can be used to alleviate the impact of disturbing variations in RSSC tasks (The high intra-class diversity and inter-class similarity can be seen from the example images in Figure 5.1). This motivates me to explore the method that can enhance the discriminative power of second-order statistics of CNN features to a greater extent. In this chapter, a covariance feature embedding model shorted in CFE is proposed, which contains the following components. First, it tries to use the expanded Siamese-style CNN architecture (Bromley et al., 1993) to learn rotation-invariant CNN features for the input image. This idea is inspired by the fact that most buildings, trees and other contextual objects have no absolute orientation in the remote sensing image, and rotating the input image can affirm the feature corresponding to the optimal appearance without changing the image content. Second, the non-linear eigenvalue (EIG) decomposition function is used to exploit the geometric structure of the covariance matrix generated by the second-order statistics of local CNN features. Two complementary matrix Frobenius norms are appended before and after the EIG decomposition function to capture use-

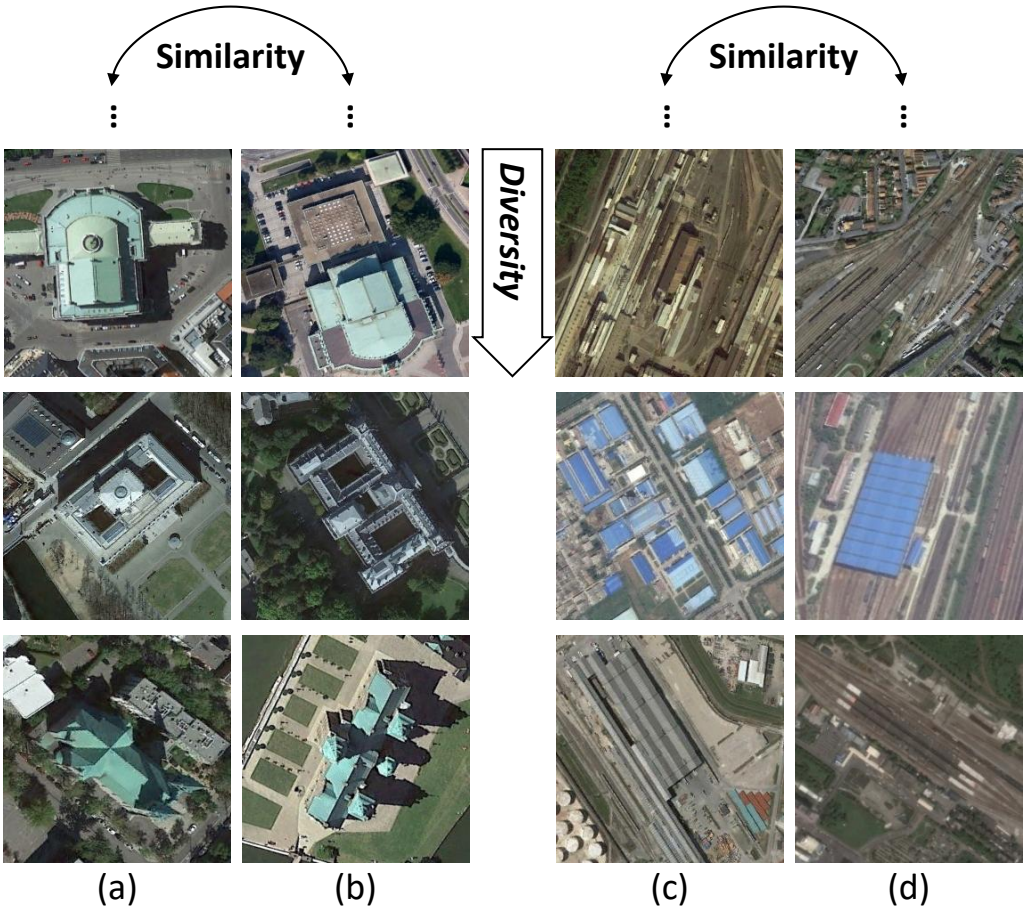


Figure 5.1: Example images to show intra-class diversity and inter-class similarity. Images are selected from NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017). From (a) to (d), the category names are *Church*, *Palace*, *Industrial area* and *Railway station*, respectively.

ful properties that are invariant under matrix rotation. Third, a novel low-norm cosine similarity (LnCS) loss is proposed, which vectorises the extracted second-order features into an angle vector space, so that the high intra-class diversity and inter-class similarity can be alleviated through punishing the angles between the feature and the corresponding weights. The contribution of the CFE model can be briefly summarised from the following aspects:

- **Elegant:** a LnCS loss is proposed to simultaneously encourage the intra-

class compactness and inter-class separability of second-order features in the embedded space.

- **Enhanced:** the discriminative power of second-order features can be enhanced by the eigenvalue (EIG) decomposition function and two complementary Frobenius norms.

- **Easy:** the CFE model is easy to implement and can be optimised in an end-to-end manner through GPU acceleration.

5.2 Method

The goal of the CFE model is to solve the ubiquitous variations that naturally exist in the remote sensing scene image datasets by maximising the feature discriminative power. Figure 5.2 shows an overview of the proposed CFE model, which is specifically designed for three main parts including image input, feature extraction and high-dimensional space measurement. To extend the model's perception of prior knowledge, the input image will be transformed through random cropping and manually defined rotation rules. The image instances generated according to different rotation angles will extract features through a Siamese-style CNN architecture with shareable parameters, and finally output features with rotation invariance in a max-out manner. The resulted CNN features are then converted into a covariance matrix, followed with multiple matrix norms to further improve its representativeness. The output second-order features will be vectorised and mapped into a hypersphere (the dimension of the flattened feature is close to 262,k, far exceeding the dimension of the general vectorised CNN feature of 4,096), in which the low-norm metric is found to be more effective than the conventional L2 distance metric. Therefore, It induces a low-norm cosine similarity loss, which is an additional margin loss that can optimise the CFE model by penalising the angles

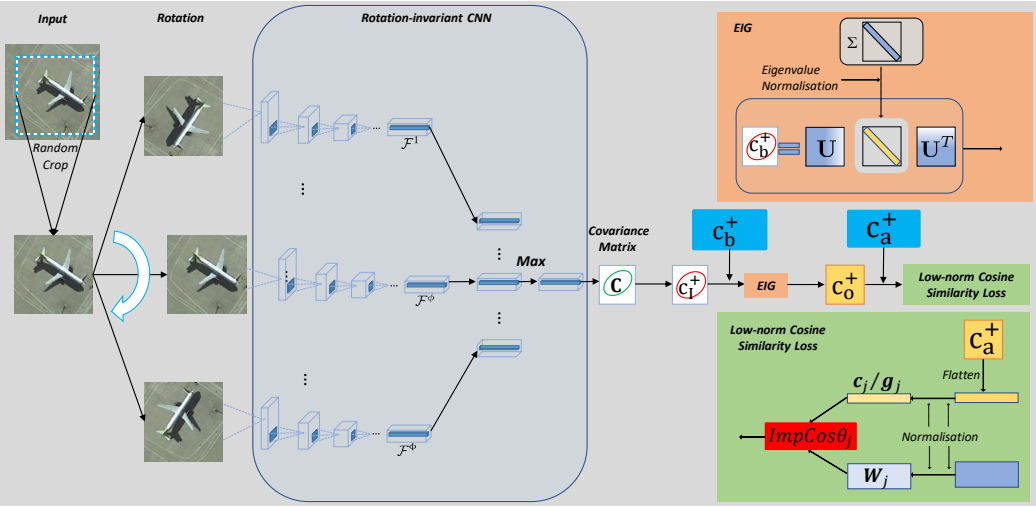


Figure 5.2: An overview of the proposed Covariance Feature Embedding model, where $\mathcal{F}$, $\mathbf{C}$ and $\mathbf{W}$ denote CNN features, covariance matrix and initialised regression weights, respectively.

between the vectorised features and the corresponding weights.

5.2.1 Rotation-invariant CNN Features

For image classification tasks, incorporating adequate prior knowledge for the input data can improve the generalisation ability of the CNN model, and vice versa. Since the remote sensing image is displayed as an overhead view, it means that the model should predict an identical result no matter how the input image is rotated. Without increasing the volume of the datasets, the proposed model tends to dynamically expand the distribution of input data during model training. Specifically, random cropping will be used to obtain a certain range of patches from the input image as sub-images. Each patch then can be rotated according to the predefined rotation rule to form multiple augmentations at different angles. The generated augmentations can be regarded as the inputs to a multi-column CNN network to extract feature representations. However, this will cause the network parameters to grow exponentially and ignore the dependencies between

different augmentations.

To compare and capture the similarities between different entities, a parallel, weight-sharing Siamese-style CNN structure will be applied (Bromley et al., 1993). Unlike the original Siamese structure (Bromley et al., 1993), which only contains two identical network components, an extended Siameses-style CNN network equivalent to the number of augmentations is used to extract multiple CNN features at once. These subnetworks with the identical configuration mean that the updated parameters of the model can be reflected in all subnets while maintaining the overall network weights consistent with an individual subnetwork. Because it expects the model to be invariant to rotation variations, the maximum operator is applied at the end of the Siamese-style CNN architecture. Through this design, it can not only obtain a certain number of feature maps corresponding to the most important classification response but also ensure that the resulting features are invariant to the predefined rotation transformations.

Given an input image denoted as $\mathcal{X} \in \mathbb{R}^{H \times W \times C}$, where H, W and C denote image height, width and channels, respectively. The CFE model first randomly crops the patch from the input image, and then rotates the patch according to a predefined set of transformations Φ. Considering that CNN itself has a certain degree of translation invariance, only random cropping and rotation are investigated in the CFE model, which can approximate the variations brought by image affine transformation in almost the simplest way. The augmentations of the patch image will be sent to the Siamese-style network based on the VGG backbone for feature extraction. This procedure can be represented as:

$$\mathcal{F}^{\phi} = f_e(\phi(\mathcal{X})), \tag{5.1}$$

where f_e denotes the function of feature extraction, $\phi(\cdot)$ is the set of rotations and

can be derived from: $\phi_r = \frac{360^\circ}{\dim(\Phi)}$ with $\dim(\cdot)$ denotes the length of rotation set. $\mathcal{F}^\phi \in \mathbb{R}^{H' \times W' \times C'}$ is the CNN feature for a rotated regional image augmentation, where H',W' and C' denote the feature height, width and channels, respectively.

The obtained CNN features of different augmentations will be stacked along the new axis and decompressed on the same axis using the element-wise maximum operation. This ensures that the output feature maps always return the value that has the largest response to the classification function at the same position. Formally, this process can be denoted as:

$$\mathcal{F} = \max_{\phi \in \Phi} f_t(\mathcal{F}^\phi), \tag{5.2}$$

where f_t is the function used to learn the CNN feature corresponding to the highest response of the classification function. In this way, the dimension of the generated CNN features $\mathcal{F}$ is the same as the output of an individual subnetwork.

5.2.2 Forward Propagation of Covariance Matrix

Recently, it turns out that the second-order statistical feature is more powerful than the first-order counterparts (P. Li et al., 2017; Lin et al., 2015; Gao et al., 2016; Acharya et al., 2018). Especially, in the context of image classification, general spatial pooling introduces the invariance to transformations while second-order statistics maintain selectivity (Kong & Fowlkes, 2017). This also indicates that it is necessary to convert CNN features into a covariance matrix to form a holistic representation. As described in the Eq.(5.2), the generated rotation-invariant CNN features $\mathcal{F}$ can be flattened and expressed in matrix form $\mathbf{F}$. Thus, the matrix $\mathbf{F}$ is composed of a series of vectors: $[\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_i, \cdots \mathbf{f}_N]$, where $\mathbf{f_n} \in \mathbb{R}^{C'}$ with

$N = H' \times W'$. Then, the covariance matrix can be achieved by following:

$$\mathbf{C} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{f}_i - \boldsymbol{\mu})(\mathbf{f}_i - \boldsymbol{\mu})^\top, \tag{5.3}$$

where $\boldsymbol{\mu}$ denotes the mean of feature vectors and can be computed by: $\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{f}_i$. Considering that the first-order statistics of deep CNN features conform to the assumption of Gaussian distribution, it is worth inferring that the statistical distribution of the intermediate representation should also have Gaussian properties. Then, the resulted covariance matrix in Eq.(5.3) can be modelled in a single Gaussian distribution as illustrated in Chapter 4. However, the conversion of Gaussian covariance will involve more steps and is not the centre of this chapter, so only an ordinary covariance matrix is adopted as the final form of the second-order feature.

Covariance matrices $\mathbf{C}^{\phi}$ are symmetric and positive definite only when they satisfy the properties of linearly independent components in the space of feature vectors $[\mathbf{f_1}, \mathbf{f_2}, \cdots, \mathbf{f_N}]$. However, it is actually difficult to guarantee that the resulting matrix is a strict SPD matrix. To tackle this problem, a regularisation operation is introduced:

$$\mathbf{C}_I^+ = \mathbf{C} + \lambda \mathrm{tr}(\mathbf{C})\mathbf{I}, \tag{5.4}$$

where λ, tr and $\mathbf{I}$ denote a small ridge parameter, the matrix trace operation and the identity matrix, respectively.

It is impractical to directly measure the distance of elements on the SPD matrix due to the particular structure of the SPD (i.e., Riemannian) manifold. In this case, suitable metric functions are highly needed to estimate the true distance of the elements on the SPD manifold. As mentioned in (P. Li et al., 2017), the measurement of Riemannian manifold usually involves two measurement func-

tions: Affine Invariant Riemannian Metric (AIRM) and Log-Euclidean Metric. The former metric is affine-invariant by computing the Frobenius norm of the logarithm of matrices. Specifically, given two SPD matrices $\mathbf{C}_1$ and $\mathbf{C}_2$, the AIRM is defined as: $\mathrm{d}(\mathbf{C}_1, \mathbf{C}_2) = \|\log(\mathbf{C}_1^{-1/2}\mathbf{C}_2\mathbf{C}_1^{1/2})\|_F$, where $\|\cdot\|_F$ denotes the matrix Frobenius norm. Because it needs to calculate the inverse of the matrix, AIRM is computationally expensive and coupled, which implies it is difficult for AIRM to challenge large-scale datasets. For large-scale datasets, the latter Log-Euclidean metric is more preferable because it is a decoupled metric and is invariant to the similarity transformations under the orthogonal transformation and scaling (i.e., the computation of using Log-Euclidean metric is invariant concerning a change of ordinates by the similarity).

The logarithm of Riemannian SPD matrices succeeds in endowing the Riemannian manifold of SPD matrices with a Lie group structure (Acharya et al., 2018). Therefore, the Euclidean metric can be used to measure the Riemannian manifold in the logarithmic Euclidean space obtained by flattening operations. However, the logarithm of the SPD matrix suffers from the problem of numerical instability. The reason for the numerical instability is that the logarithmic function greatly changes the magnitude of the eigenvalues, and even reverses the significance of the eigenvalues, especially small eigenvalues are overstretched. It will use a more robust matrix square root normalisation as an approximation of the logarithm of the covariance matrix (i.e., the logarithm requires the eigenvalue to be strictly positive while the square root normalisation is not).

Proper matrix normalisation methods typically bring about an unexpected effect in accelerating the convergence speed and improving the generalisation ability of the model, which can be found in (Lin & Maji, 2017; Wei et al., 2018; Cui et al., 2017; P. Li et al., 2017, 2018) and also been investigated in MG-CAP model (S. Wang, Guan, & Shao, 2020) (Chapter 4). Taking into account the simple

and practical properties of the Frobenius norm that is invariant under rotations (namely, given a input matrix $\mathbf{M}$, it has $\|\mathbf{M}\|_F = \|\mathbf{UM}\|_F = \|\mathbf{MU}\|_F$ for any unitary matrix $\mathbf{U}$) (Watkins, 2004), it will further explore the method of normalising the second-order statistics through the matrix Frobenius norm. Directly appending the Frobenius norm of the matrix after the EIG decomposition function may degrade performance (P. Li et al., 2017). This is because the Frobenius norm has non-trivially changed the magnitude of the input matrix. As a more feasible solution, a pair of complementary matrix Frobenius norms (i.e., $\mathbf{C}_b^+$ and $\mathbf{C}_a^+$ that will be introduced in Eq.(5.10)) are presented before and after the EIG decomposition function. As shown in Figure 5.2, the before-norm $\mathbf{C}_b^+$ can be represented as:

$$\mathbf{C}_b^+ = \frac{1}{\|\mathbf{C}_I^+\|_F}\mathbf{C}_I^+, \tag{5.5}$$

where $\mathbf{C}_b^+$ is a matrix normalised using Frobenius norm and is ready to be used in the following EIG decomposition function. If the n-th eigenvalues of $\mathbf{C}_I^+$ are denoted as ν_n, then it has $\|\mathbf{C}_I^+\|_F = \sqrt{\sum_n^c \nu_n^2}$ and $\frac{\nu_n}{\sqrt{\sum_n \nu_n^2}} > 0$ which satisfies the property of SPD matrix (i.e., strictly positive eigenvalues).

In the Eq.(5.2), it shows how to obtain a certain number of CNN features that are invariant to rotation. After some manipulations, the obtained CNN features can be converted into SPD matrix, and the matrix Frobenius norm can be performed. With the normalised matrix $\mathbf{C}_b^+$, the EIG decomposition function can be represented as:

$$\mathbf{C}_O^+ = f_d(\mathbf{C}_b^+) = \mathbf{U}_b\mathbf{F}(\mathbf{\Sigma}_b)\mathbf{U}_b^\top, \tag{5.6}$$

where $\mathbf{C}_O^+$ denotes the output of the EIG decomposition function, $\mathbf{F}(\mathbf{\Sigma}_I)$ is the normalised matrix to scale the spectrum of decomposed eigenvalues. Then, the operation on the matrix power of the SPD matrix is equivalently transformed into the operation on its eigenvalues. Suggested by (P. Li et al., 2017; S. Wang, Guan,

& Shao, 2020), the square-root normalisation can be used to approximate the logarithm of the eigenvalue, but the performance is more robust because it allows non-negative eigenvalues. Formally,

$$\mathbf{F}(\mathbf{\Sigma}_b) = \text{diag}((\nu_1)^{\frac{1}{2}}, \ldots, (\nu_n)^{\frac{1}{2}}, \ldots, (\nu_c)^{\frac{1}{2}}), \tag{5.7}$$

where $\text{diag}(\cdot)$ is the matrix diagonal operation, $(\nu_n)^{\frac{1}{2}}$ is the square-root of eigenvalues ν_i with $i = 1, ..., c$ arranged in non-increasing order. Although the square-root normalisation allows non-negative eigenvalues, correcting eigenvalues to be positive can make it more robust. In order to achieve this goal, the following rectification function is introduced:

$$\mathbf{R} = \max(\varepsilon\mathbf{I}, \mathbf{\Sigma}_I), \tag{5.8}$$

where ε and $\mathbf{I}$ denote a threshold and an identity matrix, respectively. ν_n will be replaced with $\mathbf{R}(n, n)$ to ensure that all eigenvalues are positive. This function is similar to the activation function ReLU (Glorot et al., 2011) but does not cause sparseness (Z. Huang & Van Gool, 2017; S. Wang, Guan, & Shao, 2020). Especially, the diagonal elements can be defined as:

$$\mathbf{R}(n, n) = \begin{cases} \mathbf{\Sigma}_b(n, n), & \mathbf{\Sigma}_b(n, n) > \varepsilon; \\ \varepsilon, & \mathbf{\Sigma}_b(n, n) \leq \varepsilon. \end{cases} \tag{5.9}$$

where $\mathbf{\Sigma}_b = \text{diag}(\nu_1, \ldots, \nu_n, \ldots, \nu_c)$ and can be obtained using the standard EIG-decomposition function.

Since the pre-normalisation function in the Eq.(5.5) has non-trivially changed the magnitude of the input SPD matrix, a supplementary normalisation is needed to counteract the impact of this change. The after-norm $\mathbf{C}_a^+$ is given to solve this

problem and write it as:

$$\mathbf{C}_a^+ = \sqrt{\|\mathbf{C}_I^+\|_F}\mathbf{C}_O^+. \tag{5.10}$$

The result matrix $\mathbf{C}_a^+$ will be perceived as the final feature of training the classifier under the given objective function. In addition, due to the use of the relatively more stable square-root norm of the matrix as the approxiamation of the logarithmic norm, and the integration of the rectification function into the EIG-decomposition function, the calculation of the entire covariance section is greatly reduced compared with the methods presented in (Acharya et al., 2018; Z. Huang & Van Gool, 2017).

5.2.3 Low-norm Cosine Similarity Loss

According to the above process, it can obtain the normalised covariance matrix $\mathbf{C}_a^+$, which has the useful characteristic of being invariant under predefined transformations. The resulting covariance matrix needs to be flattened in order to fit the classifier. However, this will generate vectors in a high-dimensional space where typical L_2 norm-based measurements are most likely to degrade performance (Aggarwal, Hinneburg, & Keim, 2001). To tackle this problem, it will seek a measurement that can positively affect the enhancement of feature discrimination in high-dimensional space. Inspired by the recent success of marginal-based metric learning (W. Liu et al., 2017; F. Wang et al., 2017, 2018; H. Wang et al., 2018; Deng et al., 2019), it will consider introducing proper measurements in the vectorised high-dimensional space to improve the discriminative ability of second-order features. Concretely, the learned feature is mapped to an angular space and the angles produced by the multiplication operation between the flattened features and the corresponding weights are penalised. Before presenting the proposed loss function, it is necessary to introduce the widely used cross-entropy

loss function, which can be written as:

$$L_{Softmax} = \frac{1}{N}\sum_{i=1}^{N} -\log\frac{e^{\mathbf{W}_{y_i}^T\mathbf{x}_i+b_{y_i}}}{\sum_{j=1}^{n} e^{\mathbf{W}_j^T\mathbf{x}_i+b_j}}, \tag{5.11}$$

where N and n denote the batch size and the number of classes. $\mathbf{W}_j \in \mathbb{R}^d$ represents the j-th column of weight $\mathbf{W} \in \mathbb{R}^{d\times n}$. $\mathbf{x}_i \in \mathbb{R}^d$ denotes the vectorised covariance feature of the i-th sample, belonging to the y_i-th category. $b_j \in \mathbb{R}^n$ denotes the bias term and is fixed to 0 for simplicity (W. Liu et al., 2017). Following (F. Wang et al., 2018, 2017; H. Wang et al., 2018; Deng et al., 2019), the target logit can be transformed to the following form:

$$\mathbf{W}_j^T\mathbf{x}_i = \|\mathbf{W}_j\|\|\mathbf{x}_i\|\cos\theta_j, \tag{5.12}$$

where θ_j is the angle between the weight $\mathbf{W}_j$ and the feature $\mathbf{x}_i$. To ensure effective learning of features, it is recommended that the weight $\mathbf{W}_j$ remain invariable, which can be achieved by fixing it to 1 through L_2 normalisation, namely, $\|\mathbf{W}_j\| = 1$ (H. Wang et al., 2018; F. Wang et al., 2018). Since the classification score is obtained by calculating the cosine similarity between the feature vectors, the feature norm needs to be fixed and re-scaled to s by L_2 normalisation (i.e., $\|\mathbf{x}_i\| = s$) (H. Wang et al., 2018; Deng et al., 2019).

Through careful inspection of Eq.(5.12), it is actually derived from the Minkowski distance, usually written as:

$$d_{Euclid}(\mathbf{W}_j, \mathbf{x}_i) = \left[\sum_{i=0}^{n}|(\mathbf{W}_j - \mathbf{x}_i)|^p\right]^{\frac{1}{p}}, \tag{5.13}$$

where $p = 2$, $p = 1$ and $p < 1$ correspond to the Euclidean norm, Manhattan norm and fractional norm (note: it needs to remove the exponent of $\frac{1}{p}$ for the

case of $p < 1$), respectively. Euclidean distance is not an ideal distance measurement in high-dimensional space and it has been proven by (Aggarwal et al., 2001) from both theoretical and empirical perspectives. More specifically, in high-dimensional space, the distances of the nearest and the farthest neighbours to a given observation of interest approach the same (i.e., the ratios of measured distances close to 1 for a wide variety of data distributions). Furthermore, this means that for different data points, the distance becomes evenly far away and makes it difficult to distinguish. By comparing the behaviours of different L_k norms, (Aggarwal et al., 2001) reported that the lower value of k in L_k norm is consistently more preferable in high-dimensional space. Inspired by these observations, it presents a novel distance measurement method based on the signed square root of the L_1 norm, which is used to derive the angle between weights and features. Formally, it can be written as:

$$\text{Imp\_}\cos\theta_j = \frac{\text{sign}(\mathbf{W}_j)\sqrt{|\mathbf{W}_j|}\text{sign}(\mathbf{x}_i)\sqrt{|\mathbf{x}_i|}}{\|\mathbf{W}_j\|_2\|\mathbf{x}_i\|_2}, \tag{5.14}$$

The improved $\text{Imp\_}\cos\theta_j$ can be plugged into most additive marginal loss functions, including (F. Wang et al., 2018, 2017; W. Liu et al., 2017; H. Wang et al., 2018). To demonstrate the superiority of $\text{Imp\_}\cos\theta_j$, it used to replace the method to calculate the angle in (Deng et al., 2019) and produces the following Low-norm Cosine Similarity (LnCS) loss:

$$L = \frac{1}{N}\sum_{i=1}^{N} -log\frac{e^{s(\text{Imp\_}\cos(\theta_{y_i}+m))}}{e^{s(\text{Imp\_}\cos(\theta_{y_i}+m))} + \sum_{j=1,j\neq y_i}^{n} e^{s\ \text{Imp\_}\cos\theta_j}}. \tag{5.15}$$

where s and m denote the re-scale parameter and angular margin penalty, θ_j is the angle between the weight $\mathbf{W}_j$ and the feature $\mathbf{x}_j$.

show the improvement brought by the loss of LnCS, it carried out a toy game on

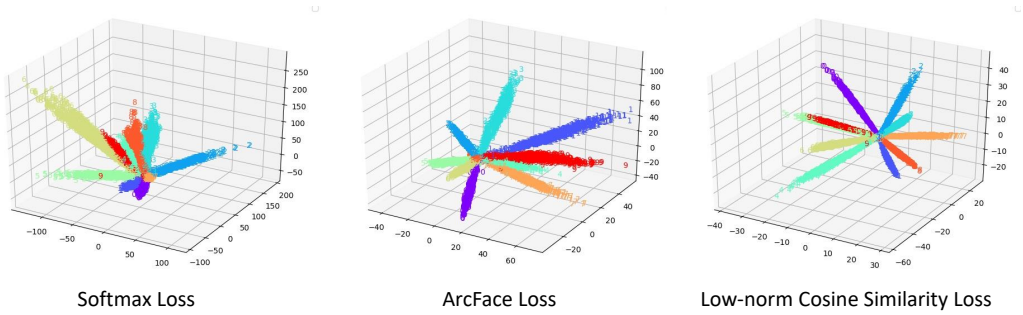


Figure 5.3: Visualisation of embedding features by using Softmax loss, ArcFace loss and the proposed loss function on MNIST dataset (LeCun & Cortes, 2010).

MNIST dataset (LeCun & Cortes, 2010). For a fair comparison, only allowed to change the loss function while maintaining the rest of architecture the architecture being consistent. Three different losses were compared, including the original Softmax, the ArcFace loss and the suggested LnCS loss. As shown in Figure 5.3, the range of embedded spaces decreases from (a)-(c). To give a concrete example, the range of the embedding space of the original Softmax loss is about 200-300. For the ArcFace loss, the range of the three coordinates of the embedding space is approximately 60-140. The proposed LnCS loss can map input features into three vector spaces with dimensions not exceeding 80. Furthermore, compared with the other two losses, the proposed LnCS loss exhibits a significant improvement in terms of the compactness of each category. Through measuring the true geodesic distance on the superellipsoid rather than the hypersphere as described in ArcFace (Deng et al., 2019), the LnCS loss function can simultaneously enhance the intra-class compactness and inter-class discrepancy of vectorised covariance feature.

5.2.4 Backward Propagation of Covariance Matrix

Efficient back-propagation algorithm is an indispensable factor of the deep learning model. The efficiency of the gradient is usually related to two aspects, includ-

ing whether it is completely differentiable and whether it can be accelerated by the GPU. On the one hand, it considers whether the proposed model can be supported by GPU acceleration so that it is possible to train the model efficiently on large-scale datasets. However, the eigenvalue and singular value decomposition (SVD) function have not been effectively supported on the NVIDIA CUDA platform. Some existing methods are forced to use the CPU to train deep learning models related to EIG and SVD, which greatly reduces training efficiency. For example, the improved bilinear pooling (Lin & Maji, 2017) employed Newton iterations as an approximation of the square-root of the matrix, while (P. Li et al., 2017) computed the EIG-decomposition algorithm in single-precision floating-point format on CPUs. On the CUDA platform, the gradient of the calculated eigenvalue is usually close to infinity, because the frequent occurrence of the identical eigenvalue will cause the corresponding eigenvector to be arbitrary in the decomposition process. To avoid this problem, it will set the infinite gradient value to 0 as introduced in Section 4.2.4, so that the gradient calculation is not interrupted during back-propagation. On the other hand, it also concerns whether the proposed method can be trained in an end-to-end manner. Especially, it takes advantage of the non-linear matrix back-propagation method introduced in (Ionescu et al., 2015; P. Li et al., 2017) to calculate the gradient of the proposed method. Considering that the final LnCS loss is completely differentiable, the derivative of L with respect to a specific layer can be expressed in a similar form as: $\frac{\partial L}{\partial C_a^+}$. For simplicity, the colon-product of two matrices is written in the matrix trace manner. After some arrangements, the chain rule can be written as follows:

$$\mathrm{tr}\left(\left(\frac{\partial L}{\partial \mathbf{C}_a^+}\right)^\top \mathrm{d}\mathbf{C}_a^+\right) = \mathrm{tr}\left(\left(\frac{\partial L}{\partial \mathbf{C}_I^+}\right)^\top \mathrm{d}\mathbf{C}_I^+ + \left(\frac{\partial L}{\partial \mathbf{C}_O^+}\right)^\top \mathrm{d}\mathbf{C}_O^+\right), \quad (5.16)$$

where $\mathrm{d}\mathbf{C}_a^+$ is the variation of $\mathbf{C}_a^+$. With some manipulations, it will produce:

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{C}_O^+} &= \sqrt{\|\mathbf{C}_I^+\|_F}\frac{\partial L}{\partial \mathbf{C}_a^+} \\ \frac{\partial L}{\partial \mathbf{C}_I^+ \mid_{\text{after}}} &= \frac{1}{2\|\mathbf{C}_I^+\|_F^{\frac{3}{2}}}\,\mathrm{tr}\left(\left(\frac{\partial L}{\partial \mathbf{C}_a^+}\right)^\top \mathbf{C}_O^+\right)\mathbf{C}_I^+,\end{aligned} \tag{5.17}$$

Once the derivative of $\frac{\partial L}{\partial \mathbf{C}_I^+ \mid_{\text{after}}}$ has been obtained, it will calculate the derivatives of $\mathrm{d}\mathbf{U}_b$ and $\mathrm{d}\mathbf{\Sigma}_b$. Analogical to the Eq.(5.16), it will generate a new chain rule for $\mathbf{C}_O^+$ ($\mathbf{C}_O^+$ and $\mathbf{C}_b^+$ are the same) which can be written as:

$$\mathrm{tr}\left(\left(\frac{\partial L}{\partial \mathbf{C}_O^+}\right)^\top : \mathrm{d}\mathbf{C}_O^+\right) = \mathrm{tr}\left(\left(\frac{\partial L}{\partial \mathbf{U}_b^+}\right)^\top : \mathrm{d}\mathbf{U}_b^+ + \left(\frac{\partial L}{\partial \mathbf{\Sigma}_b^+}\right)^\top : \mathrm{d}\mathbf{\Sigma}_b^+\right), \tag{5.18}$$

Besides, the variation of $\mathrm{d}\mathbf{C}_O^+$ can be derived from the standard variation of the EIG-decomposition function:

$$\mathrm{d}(\mathbf{C}_O^+) = \mathrm{d}\mathbf{U}_b\mathbf{F}(\mathbf{\Sigma}_b)\mathbf{U}_b^\top + \mathbf{U}_b\mathrm{d}\mathbf{F}(\mathbf{\Sigma}_b)\mathbf{U}_b^\top + \mathbf{U}_b\mathbf{F}(\mathbf{\Sigma}_b)\mathrm{d}\mathbf{U}_b^\top, \tag{5.19}$$

where $\mathrm{d}\mathbf{F}(\mathbf{\Sigma}_b) = \mathrm{diag}\left(\frac{1}{2}(\nu_1)^{-\frac{1}{2}}, \ldots, \frac{1}{2}(\nu_c)^{-\frac{1}{2}}\right)$. After some rearrangements, $\mathrm{d}\mathbf{U}_b$ and $\mathrm{d}\mathbf{\Sigma}_b$ can be denoted as the following form:

$$\begin{aligned}\mathrm{d}\mathbf{U}_b &= 2\mathbf{U}_b\left(\mathbf{Q}^T \odot \left(\mathbf{U}_b^\top \mathrm{d}\mathbf{C}_b^+ \mathbf{U}_b\right)_{sym}\right), \\ \mathrm{d}\mathbf{\Sigma}_b &= \frac{1}{2}\left(\mathrm{diag}(\nu_1)^{-\frac{1}{2}}, \ldots, (\nu_c)^{-\frac{1}{2}}\mathbf{U}_b^\top \mathrm{d}\mathbf{C}_b^+ \mathbf{U}_b\right)_{diag},\end{aligned} \tag{5.20}$$

where $\odot$ denotes the Hadamard product of matrix. Besides, $\mathbf{M}_{sym} = \frac{1}{2}(\mathbf{M} + \mathbf{M}^\top)$ and $\mathbf{M}_{diag}$ is a matrix with all off-diagonal elements set to 0. In particular, $\mathbf{Q}$ can be achieved by:

$$\mathbf{Q}(i,j) = \begin{cases} \frac{1}{\nu_i - \nu_j}, & i \neq j; \\ 0, & i = j. \end{cases} \tag{5.21}$$

More detailed information about the derivative shown in Eq.(5.19), please see (Ionescu et al., 2015). The specific partial derivatives of the loss function can be obtained by plugging Eq.(5.20) into Eq.(5.18) and it will yield:

$$\frac{\partial L}{\partial \mathbf{C}_b^+} = \mathbf{U}_O \left\{ \left(\mathbf{Q}^\top \odot \left({\mathbf{U}_b}^\top \frac{\partial L}{\partial \mathbf{U}_b} \right) \right) + \left(\frac{\partial L}{\partial \mathbf{\Sigma}_b} \right)_{diag} \right\} {\mathbf{U}_b}^\top. \tag{5.22}$$

Once the derivative of $\mathbf{C}_b^+$ has been achieved, it can obtain the partial derivatives of $\frac{\partial l}{\partial \mathbf{U}_b}$ and $\frac{\partial l}{\partial \mathbf{\Sigma}_b}$ based on the variation of $\mathrm{d}\mathbf{C}_b^+$ as:

$$\mathrm{d}\mathbf{C}_b^+ = 2(\mathrm{d}\mathbf{U}_b g(\mathbf{\Sigma}_b)\mathbf{U}_b^\top)_{sym} + \mathbf{U}_b g^{'}(\mathbf{\Sigma}_b)\mathrm{d}\mathbf{\Sigma}_b\mathbf{U}_b^\top, \tag{5.23}$$

Then, the derivatives for $\mathbf{U}_b$ and $\mathbf{\Sigma}_b$ can be achieved by:

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{U}_b} &= 2\left(\frac{\partial L}{\partial \mathbf{C}_O^+} \right)_{sym} \mathbf{U}_b(\mathbf{R})^{\frac{1}{2}} \\ \frac{\partial L}{\partial \mathbf{\Sigma}_b} &= \frac{1}{2\sqrt{\sum_i \nu_i}} \left(\mathrm{diag}(\nu_1^{-\frac{1}{2}}, ..., \nu_c^{-\frac{1}{2}}) \mathbf{U}_b^\top \left(\frac{\partial l}{\partial \mathbf{C}_O^+} \right) \mathbf{U}_b \right) - \frac{1}{2\sum_i \nu_i} \mathrm{tr}\left(\mathbf{C}_O^+ \frac{\partial L}{\partial \mathbf{C}_O^+} \right), \end{aligned} \tag{5.24}$$

where $\mathbf{R}$ is the matrix introduced in Eq.(5.9) and will be used to replace $\mathbf{F}(\mathbf{\Sigma}_I)$ in the sequel. The gradient of $\mathbf{R}$ can be obtained by calculating:

$$\mathbf{R}(n,n) = \begin{cases} 1, & \mathbf{\Sigma}_I(n,n) > \varepsilon; \\ 0, & \mathbf{\Sigma}_I(n,n) \leq \varepsilon. \end{cases} \tag{5.25}$$

When the partial derivatives of $\frac{\partial L}{\partial \mathbf{U}_b}$ and $\frac{\partial L}{\partial \mathbf{\Sigma}_b}$ are obtained, they can be plugged into Eq.(5.22) and will produce the gradient of $\frac{\partial L}{\partial \mathbf{C}_b^+}$. After that, the derivative of

before-norm can be calculating by:

$$\frac{\partial L}{\partial \mathbf{C}_I^+\big|_{\text{before}}} = -\frac{1}{\|\mathbf{C}_I^+\|_F^3}\,\text{tr}\left(\left(\frac{\partial L}{\partial \mathbf{C}_\mathbf{O}^+}\right)^\top \mathbf{C}_I^+\right)\mathbf{C}_I^+ + \frac{1}{\|\mathbf{C}_I^+\|_F}\frac{\partial L}{\partial \mathbf{C}_\mathbf{O}^+} + \frac{\partial L}{\partial \mathbf{C}_I^+\big|_{\text{after}}}, \tag{5.26}$$

Once the derivation of $\frac{\partial L}{\partial \mathbf{C}_I^+\big|_{\text{before}}}$ has been obtained, the gradient of the loss function L with respect to the input covariance matrix $\mathbf{C}$ can be obtained by:

$$\frac{\partial L}{\partial \mathbf{C}} = \left(\frac{\partial L}{\partial \mathbf{C}_I^+\big|_{\text{before}}} + \left(\frac{\partial L}{\partial \mathbf{C}_I^+\big|_{\text{before}}}\right)^\top\right)\bar{\mathbf{I}}\mathbf{C}, \tag{5.27}$$

Finally, it will introduce $\nabla f_t(\mathcal{F}^\phi)$ as the gradient of the rotation-invariant CNN feature $f_t(\mathcal{F}^\phi)$. Then, the gradient of $f_t(\mathcal{F}^\phi)$ can be obtained as:

$$\frac{d\mathbf{C}}{df_t(\mathcal{F}^\phi)} = \nabla f_t(\mathcal{F}^\phi). \tag{5.28}$$

where $\phi = \underset{\phi\in\Phi}{\text{argmax}}\ f_t(\mathcal{F}^\phi)$ is the optimal rotation ϕ with respect to input image $\mathcal{X}$. Since the CNN feature $\mathcal{F}^\phi$ is generated from the Siamese-style CNN architecture based on VGGNet-16 (Simonyan & Zisserman, 2014), the loss can naturally propagate throughout the entire network. As a result, the proposed CFE model can be trained and optimised in an end-to-end manner with GPU acceleration.

5.3 Experiments

5.3.1 Implementation Details

The model is implemented using a GPU version of Tensorflow (Abadi et al., 2016). The Siamese architecture consists of multiple VGGNet-16 (Simonyan & Zisserman, 2014) networks excluding the fully-connected layers. When training

the model, in addition to traditional data augmentation techniques like random cropping, the cropped results are also randomly flipped in the vertical and horizontal orientations. The cropped patch size is set to 224 $\times$ 224 pixels on NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017) and UC Merced Land-Use dataset (Y. Yang & Newsam, 2010), and 448 $\times$ 448 pixels on AID (Xia, Hu, et al., 2017). The patch image is rotated according to the predefined parameter Φ, and the result is filled with 0 to $\sqrt{2}$ times the size of the cropped image. Then, all rotated images are resized to a uniform scale, namely, 224 $\times$ 224 pixels to facilitate feature extraction. In addition, in order to avoid the orderless problem, it is required that all the last layers of the Siamese architecture will not be pooled.

Model training starts with training the classification layer with a learning rate of 10^{-1}, and then fine-tunes the entire network with a smaller learning rate of 10^{-3}. In the warm-up phase (i.e., the first 30 epochs), the learning rate remains the same, while in the fine-tuning phase, the learning rate is periodically annealed to 0.15 every three epochs. The maximum number of rotations is 18, and the training batch size is 12. The configuration of the PC used in the experiment includes a 6-core Intel® Core™ i7-9800X@3.80 GHz CPU and a single GeForce RTX 2080Ti GPU. The whole network is optimised by the momentum optimiser with a momentum of 0.9. The value of λ in Eq.(5.4) is empirically set to $\lambda = 1 \times 10^{-4}$. The ε used to rectify the eigenvalues in Eq.(5.9) and Eq.(5.25) is set to 1×10^{-5}. Suggested by (Deng et al., 2019), the re-scale parameter s and the marginal parameter m are set to 32.0 and 0.5 for all experiments.

Table 5.1: Comparison of overall accuracy and standard deviation obtained by the proposed CFE model and previous deep learning-based methods on NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017), where T.R. is short for the training ratio.

Methods	NWPU-RESISC45	
	T.R.=10%	T.R.=20%
Transferred AlexNet (Cheng, Han, & Lu, 2017)	76.69±0.21	79.85±0.13
Fine-tuned AlexNet (Cheng, Han, & Lu, 2017)	81.22±0.19	85.16±0.18
Transferred GoogLeNet (Cheng, Han, & Lu, 2017)	76.19±0.38	78.48±0.26
Fine-tuned GoogLeNet (Cheng, Han, & Lu, 2017)	82.57±0.12	86.02±0.18
Transferred VGGNet-16 (Cheng, Han, & Lu, 2017)	76.47±0.18	79.79±0.15
Fine-tuned VGGNet-16 (Cheng, Han, & Lu, 2017)	87.15±0.45	90.36±0.18
BoCF (Cheng, Li, et al., 2017)	82.65±0.31	84.32±0.17
Triple Networks (Y. Liu & Huang, 2017)	-	92.33±0.20
Two-Stream Fusion (Y. Yu & Liu, 2018)	80.22±0.22	83.16±0.18
MSCP with AlexNet (N. He et al., 2018)	81.70±0.23	85.58±0.16
MSCP with VGGNet-16 (N. He et al., 2018)	85.33±0.17	88.93±0.14
D-CNN with AlexNet (Cheng et al., 2018)	85.56±0.20	87.24±0.12
D-CNN with GoogLeNet (Cheng et al., 2018)	86.89±0.10	90.49±0.15
D-CNN with VGGNet-16 (Cheng et al., 2018)	89.22±0.50	91.89±0.22
RTN in Chapter 3 (Z. Chen et al., 2018)	89.53±0.21	92.20±0.34
CapsNet with VGGNet-16 (W. Zhang, Tang, & Zhao, 2019)	85.08±0.13	89.18±0.14
MG-CAP (Bilinear) in Chapter 4 (S. Wang, Guan, & Shao, 2020)	89.42±0.19	91.72±0.16
MG-CAP (Log-E) in Chapter 4 (S. Wang, Guan, & Shao, 2020)	88.35±0.23	90.94±0.20
MG-CAP (Sqrt-E) in Chapter 4 (S. Wang, Guan, & Shao, 2020)	**90.83**±0.12	**92.95**±0.13
the proposed CFE with VGGNet-16	90.64±0.16	92.77±0.13

5.3.2 Experimental Results

5.3.2.1 Comparison on NWPU-RESISC45 dataset

The NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017) is one of the most challenging datasets for remote sensing scene classification. As can be seen from Table 5.1, in addition to the recently released MG-CAP with (Sqrt-E) in Chapter 4 (S. Wang, Guan, & Shao, 2020), the proposed CFE model is superior to the other deep learning methods listed. More specifically, the results of the proposed model greatly exceed the results obtained through transferring and fine-tuning of the three commonly used networks, including AlexNet (Krizhevsky et al., 2012), GoogLeNet (Szegedy et al., 2015) and VGGNet-16 (Simonyan & Zisserman,

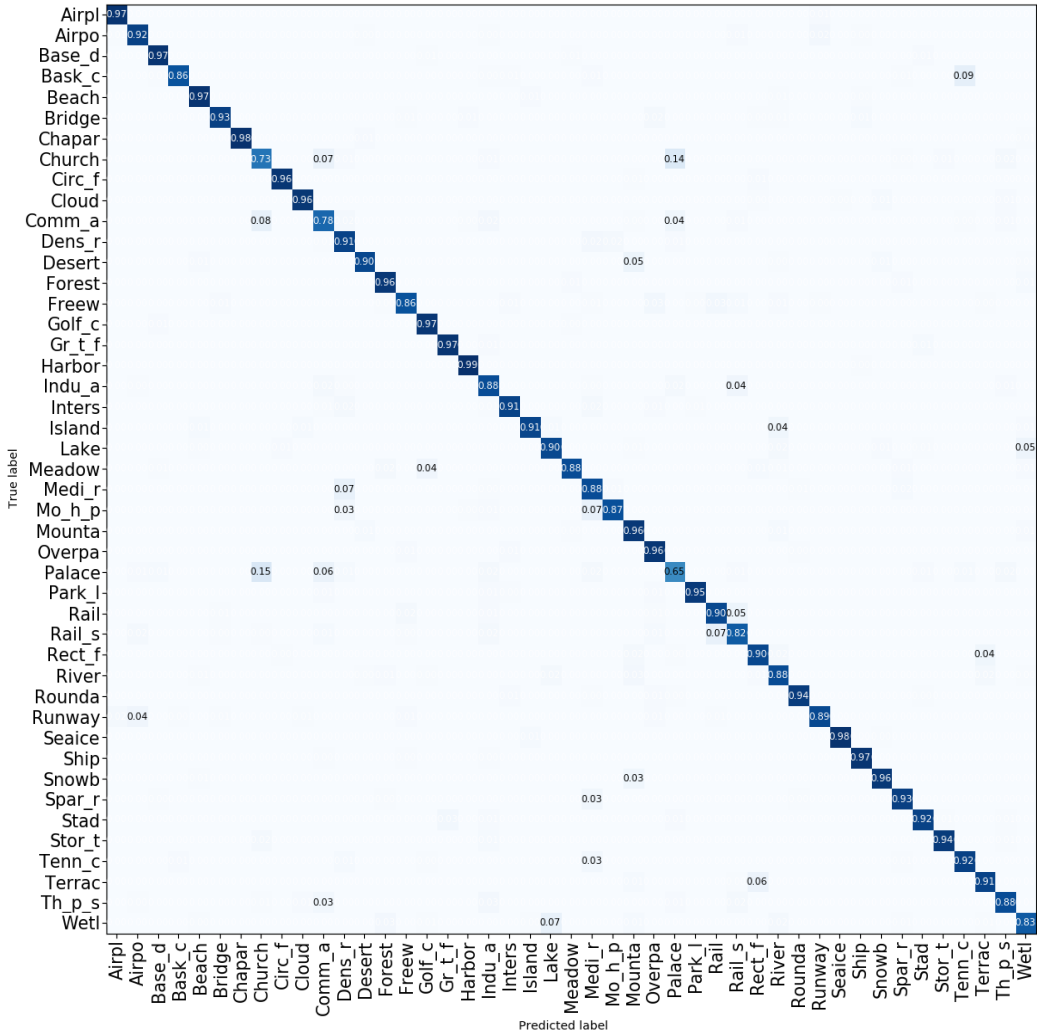


Figure 5.4: The confusion matrix on NWPU-RESISC45 dataset under the training ratio of 10%. For the sake of clarity, values less than 0.03% are omitted.

2014). Especially, compared with the transferred VGGNet-16, the results of the proposed model using 10% and 20% training samples are improved by about 4% and 13%, respectively. The similar improvement is also reflected between the proposed model and the BoCF model (Cheng, Li, et al., 2017), and the Two-Stream Fusion model (Y. Yu & Liu, 2018), as well as the CapsNet with VGGNet-16 (W. Zhang et al., 2019). When the same 10% amount of training data is used, the proposed model also improves by about 1% compared with another competitive

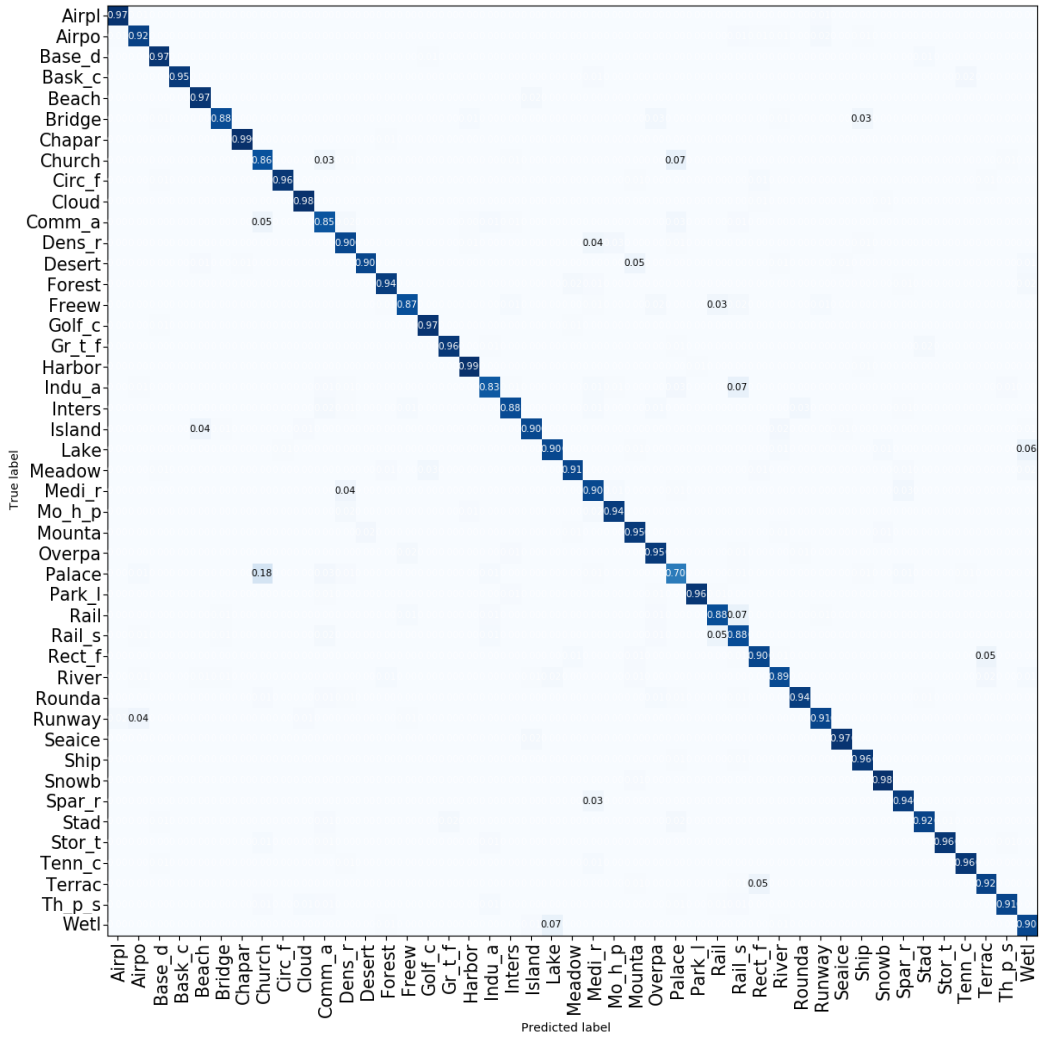


Figure 5.5: The confusion matrix on NWPU-RESISC45 dataset under the training ratio of 20%. For the sake of clarity, values less than 0.03% are omitted.

D-CNN (Cheng et al., 2018). In addition, the proposed model also has different degrees of improvement compared with similar models based on second-order statistical features, including MSCP model with AlexNet and VGGNet16 (N. He et al., 2018), RTN model with VGGNet-16 (in Chapter 3) (Z. Chen et al., 2018) and MG-CAP with Bilinear and Log-E in Chapter 4 (S. Wang, Guan, & Shao, 2020). Although the displayed accuracy of MG-CAP with Sqrt-E in Chapter 4 (S. Wang, Guan, & Shao, 2020) is slightly higher than the proposed model (namely, about

0.2% higher), it needs to crop the input image twice to obtain three granularities including the original image. This operation also means that the computational burden increases exponentially. Moreover, according to the data in Table 4.3, the highest individual granularity of using Sqrt-E only obtains an accuracy of 89.05%, which is lower than the average accuracy of the proposed model.

Two example confusion matrices are randomly selected from experiments to show more details of classification at the category-level. The confusion matrix with using 10% of training samples has been presented in Figure 5.4. It can be found that 31 out of a total of 45 categorises achieve a classification accuracy of greater than 90%. For the two most visually similar categories (i.e., **Palace** and **Church**), it produced the substantial improvements of 6% and 11% compared with the results achieved by the Fine-tuned VGGNet-16 (Cheng, Han, & Lu, 2017), respectively. From Figure 5.5, the accuracy of the **Palace** category is 70% which is an improvements of 7% compared with the Fine-tuned VGGNet-16 reported in (Cheng, Han, & Lu, 2017). In addition, the accuracy on the category of **Church** is 86%, which exceeds MG-CAP model with Sqrt-E in Chapter 4 (S. Wang, Guan, & Shao, 2020), RTN model with VGGNet-16 (in Chapter 3) (Z. Chen et al., 2018) and D-CNN model with VGGNet-16 by 14%, 13%, 11%, respectively.

5.3.2.2 Comparison on AID dataset

AID dataset (Xia, Hu, et al., 2017) another public large-scale dataset that is also used to evaluate the effectiveness of the proposed CFE model. As shown in Table 5.2, the proposed CFE model can accomplish an accuracy of 93.15% at the training ratio of 20%, with improvements about 10% over both the Fine-tuned AlexNet (Cheng, Han, & Lu, 2017) and the transferred GoogLeNet (Cheng, Han, & Lu, 2017). Notably, with using 20% of the total number of training samples, the Two-Stream Fusion model (Y. Yu & Liu, 2018), MSCP model with VGGNet16

Table 5.2: Comparison of overall accuracy and standard deviation obtained by the proposed CFE model and previous deep learning-based methods on AID dataset (Xia, Hu, et al., 2017), where T.R. is short for the training ratio.

Methods	AID	
	T.R.=20%	T.R.=50%
Transferred AlexNet (Cheng, Han, & Lu, 2017)	83.22±0.10	91.17±0.10
Fine-tuned AlexNet (Cheng, Han, & Lu, 2017)	84.23±0.10	93.51±0.10
Transferred GoogLeNet (Cheng, Han, & Lu, 2017)	84.94±0.10	92.35±0.10
Fine-tuned GoogLeNet (Cheng, Han, & Lu, 2017)	87.51±0.11	95.27±0.10
Transferred VGGNet-16 (Xia, Hu, et al., 2017)	85.77±0.10	93.21±0.10
Fine-tuned VGGNet-16 (Cheng, Han, & Lu, 2017)	89.33±0.23	96.04±0.13
salM^3 LBP-CLM (Bian et al., 2017)	86.92±0.35	89.76±0.45
TEX-NET-LF (Anwer et al., 2018)	90.87±0.11	92.96±0.18
Two-Stream Fusion (Y. Yu & Liu, 2018)	92.32±0.41	94.58±0.25
MSCP with AlexNet (N. He et al., 2018)	88.99±0.38	92.36±0.21
MSCP with VGGNet-16 (N. He et al., 2018)	91.52±0.21	94.42±0.17
D-CNN with AlexNet (Cheng et al., 2018)	85.62±0.10	94.47±0.10
D-CNN with GoogLeNet (Cheng et al., 2018)	88.79±0.10	96.22±0.10
D-CNN with VGGNet-16 (Cheng et al., 2018)	90.82±0.16	**96.89**±0.10
RTN in Chapter 3 (Z. Chen et al., 2018)	92.75±0.21	95.09±0.16
CapsNet with VGGNet-16 (W. Zhang et al., 2019)	91.63±0.19	94.74±0.17
MG-CAP (Bilinear) in Chapter 4 (S. Wang, Guan, & Shao, 2020)	92.11±0.15	95.14±0.12
MG-CAP (Log-E) in Chapter 4 (S. Wang, Guan, & Shao, 2020)	90.17±0.19	94.85±0.16
MG-CAP (Sqrt-E) in Chapter 4 (S. Wang, Guan, & Shao, 2020)	**93.34**±0.18	96.12±0.12
the proposed CFE	93.15±0.18	95.78±0.15

(N. He et al., 2018) and RTN model (in Chapter 3) can achieve higher accuracy than the D-CNN model (Cheng et al., 2018) while the D-CNN model (Cheng et al., 2018) achieved the best classification accuracy with using 50% of training data (i.e., 96.89%). Furthermore, the proposed CFE model can outperform multi-stream based RTN method (in Chapter 3) (Z. Chen et al., 2018) by 0.71%. When using a training ratio of 50%, the accuracy of the proposed CFE model can exceed the CapsNet with VGGNet-16 (W. Zhang et al., 2019) by approximately 1.0%. The accuracy of all other methods listed is lower than the proposed CFE model or remains a large gap, except for the MG-CAP with Sqrt-E in Chapter 4 (S. Wang, Guan, & Shao, 2020) (S. Wang, Guan, & Shao, 2020).

From Figure 5.6, using 20% of the training data, it can be seen that among all

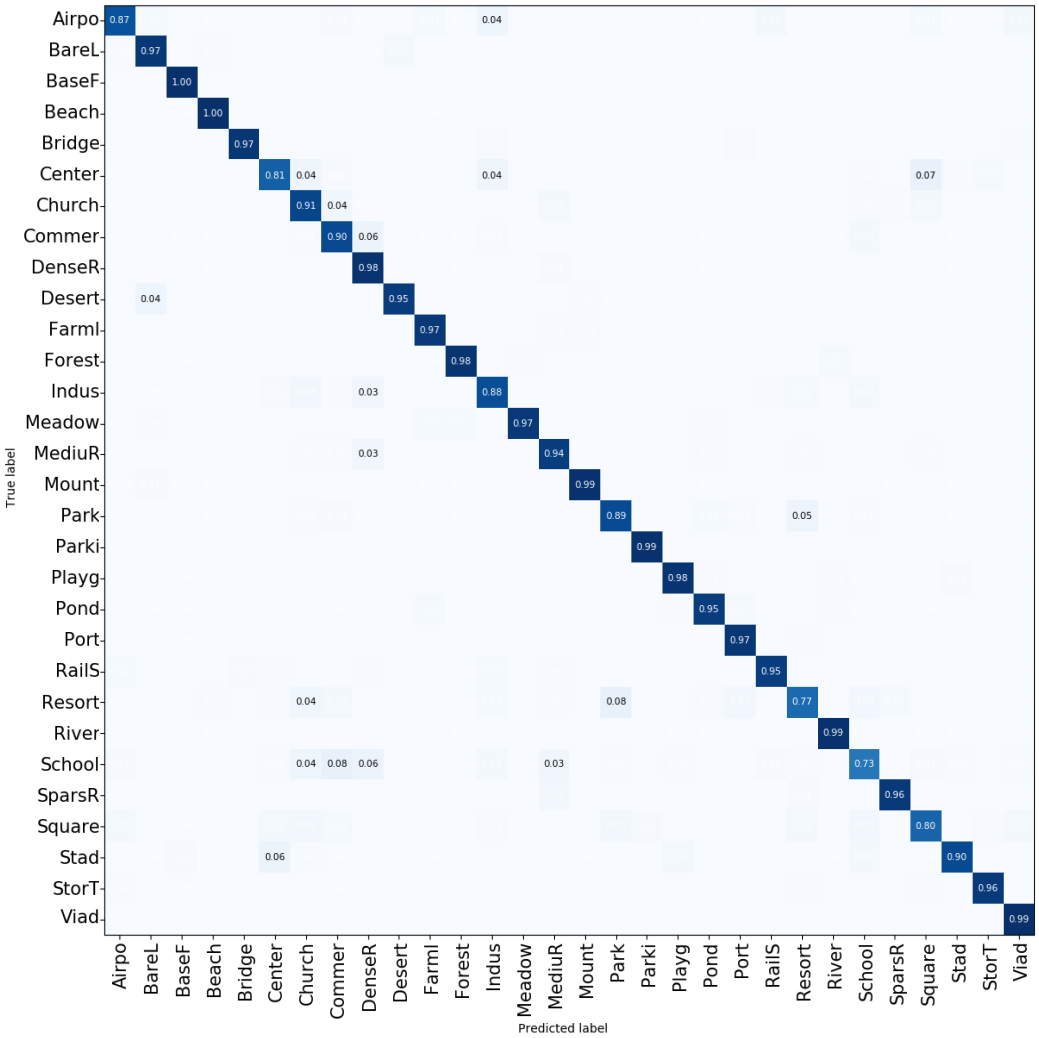


Figure 5.6: The confusion matrix on AID dataset under the training ratio of 20%. For the sake of clarity, values less than 0.03% are omitted.

30 categories, the classification accuracy of 19 categories can reach more than 95%. Especially, **Base ball field** and **Beach** categories achieve the classification accuracy of 100%. For those categories with high inter-class similarity, such as **Dense residential**, **Medium residential** and **Sparse residential**, the proposed CFE model can achieve an accuracy of 98%, 94% and 96%, respectively. In addition, the proposed CFE model can obtain the results of 77% on the **Resort** class and 73% on the **School** class, which produces great improvements compared with

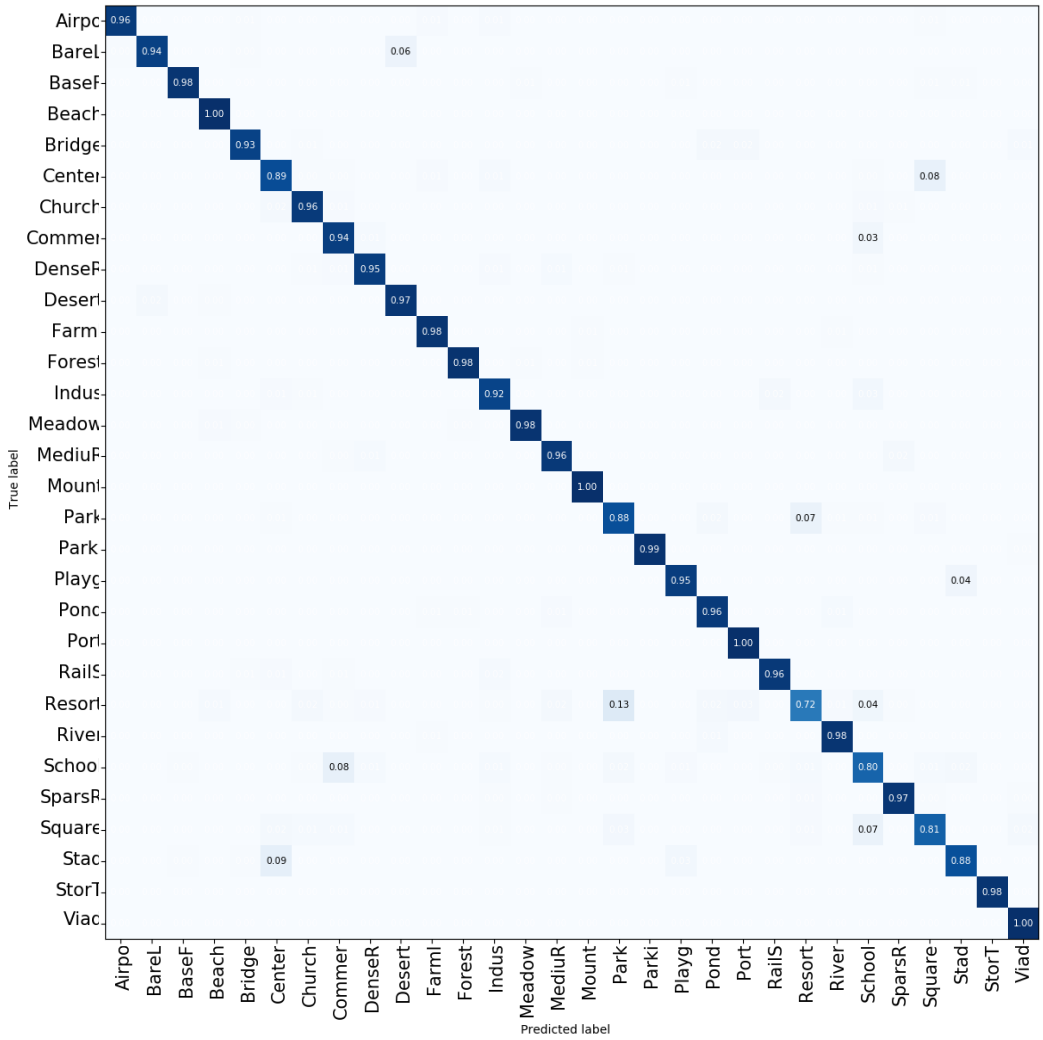


Figure 5.7: The confusion matrix on AID dataset under the training ratio of 50%. For the sake of clarity, values less than 0.03% are omitted.

the classification accuracy of 70% and 67% reported in (Xia, Hu, et al., 2017). (Note: the images in **Resort** category are usually misclassified as **Park** due to the existence of analogous objects, such as ponds and belts. Similarly, the images in **School** class are often confused with **Commercial** since they contain very similar structures, such as teaching buildings and shopping malls).

Table 5.3: Comparison of classification results (%) achieved by our CFE framework and previous methods on UC-Merced Land-Use dataset (Y. Yang & Newsam, 2010).

Methods	UC-Merced Land-Use Training Ratio=80%
Transferred AlexNet (Cheng, Han, & Lu, 2017)	94.42±0.10
Fine-tuned AlexNet (Cheng, Han, & Lu, 2017)	94.58±0.11
Transferred GoogLeNet (Cheng, Han, & Lu, 2017)	95.32±0.10
Fine-tuned GoogLeNet (Cheng, Han, & Lu, 2017)	95.82±0.20
Transferred VGGNet-16 (Cheng, Han, & Lu, 2017)	95.24±0.10
Fine-tuned VGGNet-16 (Cheng, Han, & Lu, 2017)	97.14±0.10
salM^3 LBP-CLM (Bian et al., 2017)	95.75±0.80
TEX-NET-LF (Anwer et al., 2018)	96.62±0.49
Two-Stream Fusion (Y. Yu & Liu, 2018)	98.02±1.03
GCFs+LOFs (Zeng et al., 2018)	99.00±0.35
MSCP with AlexNet (N. He et al., 2018)	97.29±0.63
MSCP with VGGNet-16 (N. He et al., 2018)	98.36±0.58
D-CNN with AlexNet (Cheng et al., 2018)	96.67±0.10
D-CNN with GoogLeNet (Cheng et al., 2018)	97.07±0.12
D-CNN with VGGNet-16 (Cheng et al., 2018)	98.93±0.10
RTN in Chapter 3 (Z. Chen et al., 2018)	98.33 ±0.71
CapsNet with VGGNet16 (W. Zhang et al., 2019)	98.81±0.22
MG-CAP (Bilinear) in Chapter 4 (S. Wang, Guan, & Shao, 2020)	98.60±0.26
MG-CAP (Log-E) in Chapter 4 (S. Wang, Guan, & Shao, 2020)	98.45±0.12
MG-CAP (Sqrt-E) in Chapter 4 (S. Wang, Guan, & Shao, 2020)	99.00±0.10
the proposed CFE	**99.19**±0.42

5.3.2.3 Comparison on UC-Merced Land-Use dataset

It also compares the proposed CFE model with state-of-the-art methods on another popular UC-Merced Land-Use dataset (Y. Yang & Newsam, 2010). Again, the proposed CFE model achieved a remarkable overall accuracy of 99.19% with using 80% of the total number of training samples. As can be seen in Table 5.3, there is an improvement of about 4-5% compared with the accuracy shown by the pure deep learning method of the proposed method (i.e, Transferred or Fine-tuned AlexNet, GoogLeNet and VGGNet-16). Compared with salM^3 LBP-CLM (Bian et al., 2017), TEX-NET-LF (Anwer et al., 2018), Two-Stream Fusion (Y. Yu &

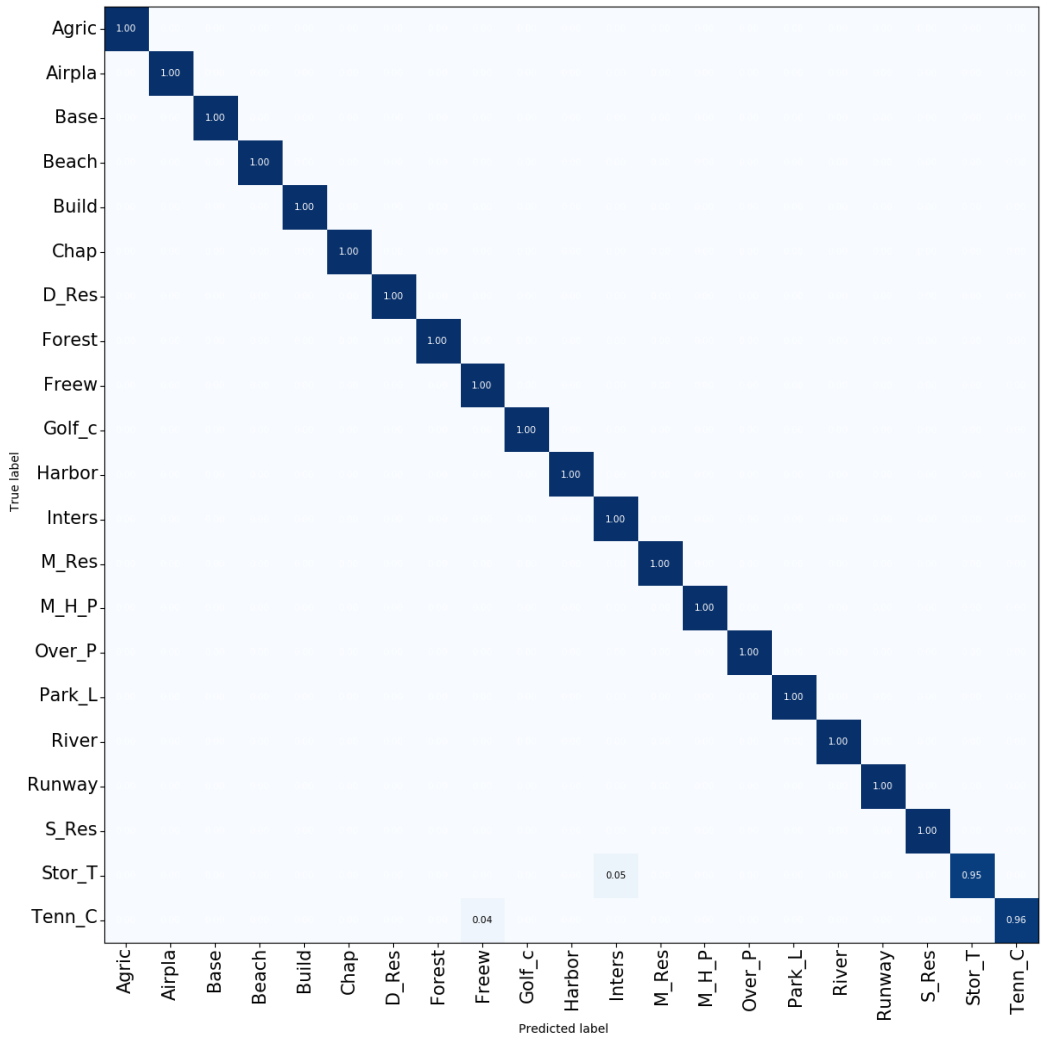


Figure 5.8: The confusion matrix on UC-Merced dataset under the training ratio of 80%. For the sake of clarity, values less than 0.03% are omitted.

Liu, 2018) and other algorithms, the proposed algorithm also has different degrees of obvious improvement. Both GCFs+LOFs model (Zeng et al., 2018) and the state-of-the-art MG-CAP model with Sqrt-E in Chapter 4 (S. Wang, Guan, & Shao, 2020) achieved 99.00% accuracy, but they are slightly lower than the accuracy obtained by the proposed algorithm, namely 0.19%.

The confusion matrix of the proposed algorithm has achieved extremely amazing

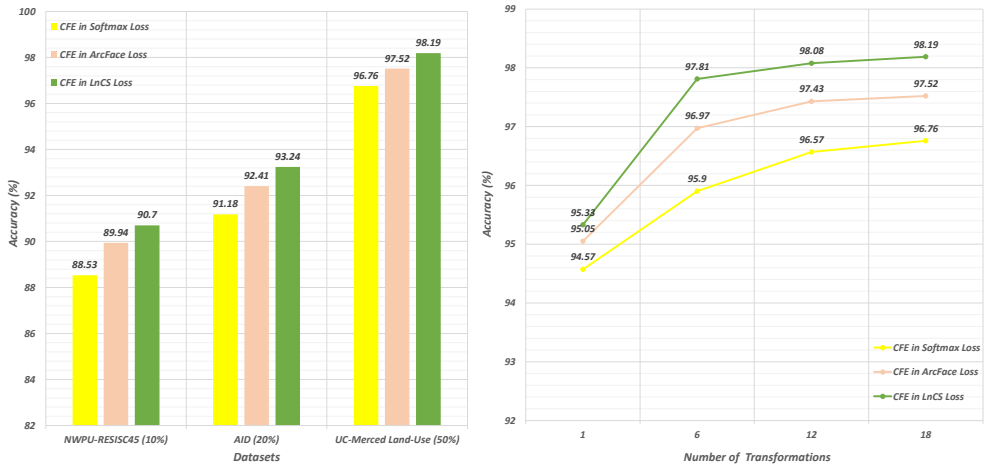
Table 5.4: Comparison of computational complexity and model size between CFE model and RTN (in Chapter 3) (Z. Chen et al., 2018), where n represents the number of streams.

	Model Complexity	Model Size (MB)
RTN (in Chapter 3) (Z. Chen et al., 2018)	*O(n(LocNet+BilinearVGG))*	68.7
CFE	*O(Cov-VGG)*	22.6

results (i.e., 99.57% overall accuracy). As shown in Figure 5.8, among these 21 categories, 19 categorises can achieve complete and accurate classification. Although subtle errors appear in categories of **Storage tanks** and **Tennis court**, their results are also very close to 1. This phenomenon not only reflects the advantages of the proposed algorithm, but also proves that under appropriate measurements, a relatively simple model can even obtain better results than complicated deep learning models on a smaller dataset.

5.3.2.4 Analysis of Model Complexity

In addition to comparing the classification accuracy of the model, the complexity of the model is also worthy of in-depth analysis. Table 1 5.4 shows the comparison between the proposed model and one of the latest models in terms of model complexity and model size. Specifically, the number of model parameters of the CFE model is significantly less than RTN (in Chapter 3) (Z. Chen et al., 2018) (i.e., three times smaller). This is because the RTN model (Z. Chen et al., 2018) requires to gradually use the localisation network and the bilinear pooling, which will increase the number of model parameters exponentially.



(a) (b)

Figure 5.9: Comparison of the classification results of the CFE model under three different loss functions. (a) The classification accuracy obtained by using different losses on three different datasets. (b) Comparison of classification accuracy obtained by using different numbers of rotation transformations.

5.3.3 Ablation Study

5.3.3.1 Loss Functions

To demonstrate the superiority of the CFE model, it evaluated the loss functions of different variants on three experimental datasets. For a fair comparison, all hyper-parameters are guaranteed to be consistent (e.g., the partition of datasets and the number of rotations). From Figure 5.9 (a), the use of the LnCS loss enables the CFE model to achieve the highest classification accuracy on the three datasets. Specifically, the CFE model based on LnCS has achieved a classification accuracy of 90.7% and 93.24% on NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017) and AID dataset (Xia, Hu, et al., 2017), respectively. Especially, on NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017), the accuracy of the CFE model with LnCS loss is 0.76% higher than the result obtained using ArcFace loss, and even 2.17% higher than using the original Softmax loss. On UC-Merced Land-

Use dataset (Y. Yang & Newsam, 2010), the difference between using different losses is less than 1%. The reason for this phenomenon may be because UC-Merced Land-Use dataset (Y. Yang & Newsam, 2010) is relatively small, so only the vectorised covariance feature is sufficient to distinguish the difference between most images.

5.3.3.2 Number of Rotations

The number of transformations is another crucial factor that affects the performance of the CFE model. As shown in Figure 5.9 (b), the overall accuracy first increases dramatically and then remains relatively stable after six transformations. With 18 rotations, all three variants of the CFE model reached their peak. The CFE model based on LnCS loss can achieve 97.81% classification accuracy with only using 6 transformations, which is even higher than the ArcFace loss and the Softmax loss using 18 transformations. Besides, it obtained an accuracy of 98.19% using 18 transformations, with improvements of 0.6% and 1.43% compared with using ArcFace and Softmax loss. Taking into account the limited computing resources and the fact that the classification accuracy increases significantly slower after the 12 transformations, the number of experimental rotations is set to 18.

5.3.4 Qualitative Visualisation and Discussion

5.3.4.1 Qualitative Analysis

The qualitative results of some experiments are shown in Figure 5.10. By comparing the successfully classified and failed images, the **School** images are easily to be misclassified as **Commercial** and **Church** images are likely confused by **Palace**. From Figure 5.10, it can be seen that the successfully classified cases of **School** category usually contain more distinguishable objects than the failed

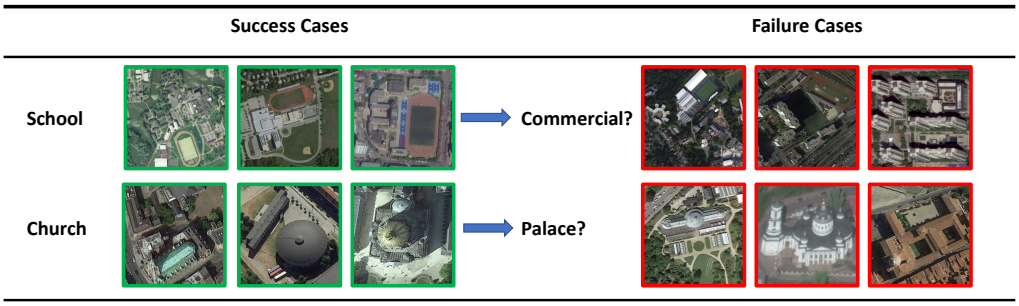


Figure 5.10: Success and Failure cases of the CFE model. School and Church images are selected from AID dataset (Xia, Hu, et al., 2017) and NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017) respectively.

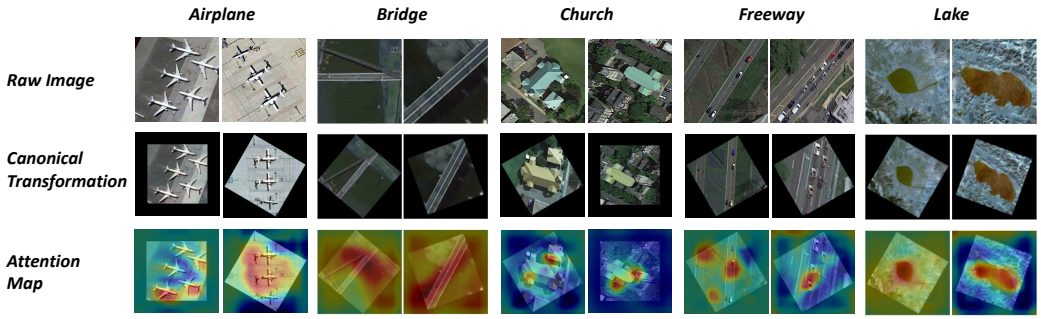


Figure 5.11: Visualised results produced by the CFE model. The first line is the raw image, the second line is the canonical transformation derived from the backpropagation, and the last line is the heat map generated using the Grad-CAM algorithm (Selvaraju et al., 2017).

cases, such as the sports field. Similarly, the algorithm is more likely to make mistakes on those **Church** images that contain tiny characteristic objects.

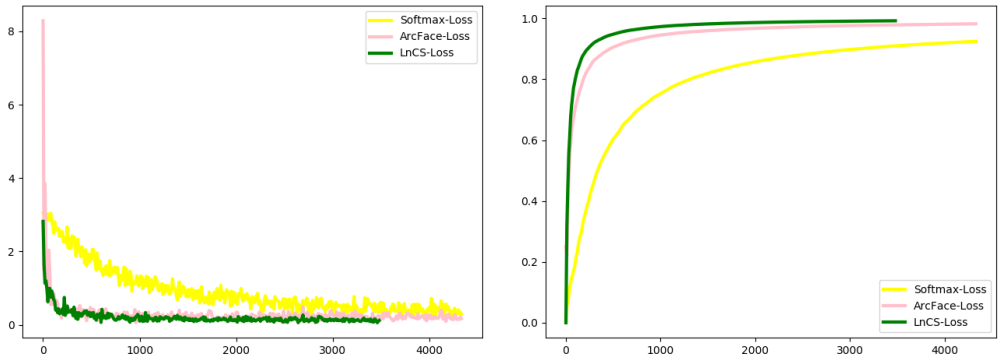
5.3.4.2 Visualisation

The interpretability of the model has always been the focus of deep learning research. In response to this problem, it provides two strategies to illustrate the interpretability of the CFE model, including how the CFE model can simultaneously learn the optimal rotation orientation and distinguished regions of the input images. For the former method, if the order of the subnetworks in the Siamese

architecture has been marked, the canonical transformation of the cropped image can naturally be derived from the back-propagation of Eq.(5.28), namely, $\phi = \underset{\phi \in \Phi}{\operatorname{argmax}} f_t(\mathcal{F}^{\phi})$. As shown in Figure 5.11, for each category, the CFE model can find that many visually similar images are oriented in roughly the same direction at certain angles. For example, **Bridge**, **Freeway** and **Airplane** in Figure 5.11. Thus, incorporating this rotation-based prior knowledge can reduce the impact of high diversity within the class caused by arbitrary rotation angles. The latter visualisation method is given by employing Gradient-weighted Class Activation Mapping (Grad-CAM) algorithm (Selvaraju et al., 2017), which relies on calculating pixel-level gradients to display the distinguished areas of the input image. From the *Attention Map* in Figure 5.11, it is not difficult to see that the CFE model can not only find the discriminative parts for images with clutter backgrounds (e.g., **Church**) but also can effectively progress the input images with irregular geometry appearances, such as **Lake**. This shows that the learned features can assist to observe the subtle differences between different images, which is believed to reduce the impact of high similarity between classes.

5.3.4.3 The Convergence Speed

A suitable measurement method can usually not only improve the classification outcome, but also make the model converge quickly, thereby saving training time and computing resources. The comparison of the convergence speed of the CFE model using different loss functions has been shown in Figure 5.12. From Figure 5.12(a), it can be found that the loss begins to converge at about 200-300 steps for LnCS loss and ArcFace loss. The convergence time is much earlier than using the softmax function, which requires about 3,000 steps to reach the same level. The original ArcFace loss has a convergence speed similar to the LnCS loss, but the proposed LnCS loss produces less vibration and is more stable, especially



(a) (b)

Figure 5.12: Comparison of the convergence speed of the CFE model under different losses.

at the beginning. From Figure 5.12(b), it can be easily seen that our LnCS loss achieves the highest classification accuracy. This is because the proposed LnCS loss allows measuring the true geodesic distance of high-dimensional data in the angular embedding space in a more accurate manner.

5.4 Conclusion

This chapter introduces a novel model named CFE, which aims to improve the discriminative ability of second-order features to solve variations in RS scene images. The proposed model uses the Siamese-style CNN architecture to learn features that are invariant to predefined transformations, and also presents a pair of complementary matrix Frobenius norms to improve the distinguishing ability of second-order statistical features. Especially, a novel low-norm cosine similarity loss is proposed to simultaneously encourage the intra-class compactness and inter-class separability by learning the angle between the vectorised covariance feature and their weights. The proposed CFE model can also be trained end-to-end using GPU. Despite the success of the proposed model, second-order features

suffer from excessively high dimensionality. In addition, it requires in-depth analysis from a theoretical perspective and explains why the features learned with Siamese-style CNNs are invariant for the predefined transformations. In the next chapter, it will commence from the perspective of group theory and finally propose a unified model to solve the above concerns.

6 | Invariant Deep Compressible Covariance Pooling

6.1 Introduction

Learning discriminative and invariant feature representation is the key to visual image categorisation. It has successively presented three different models, including the RTN model (Z. Chen et al., 2018) in Chapter 3, the MG-CAP model (S. Wang, Guan, & Shao, 2020) in Chapter 4 and the CFE model (S. Wang et al., -) in Chapter 5, respectively. The designed model can solve different challenges in the RSSC task in a targeted manner, and gradually reduce the complexity of the model while improving the classification results. The common point of these three models is that they concentrate on collecting the second-order statistics of CNN features, and then introduce different image transformations or appropriate spatial measurements. In addition to their success, there are two critical problems that need to be resolved. First, the ultra-high dimension of the second-order features contains a substantial amount of redundant information, which makes the training efficiency low. Second, the learned features are transformation-invariant, but more convincing theoretical proofs are needed. These are the motivations of this chapter, namely, to effectively reduce the feature dimension without affecting the classification accuracy, and from a theoretical point of view, prove that this idea can be regarded as a new learning paradigm to handle the problems in remote sensing scene image classification.

Compared with conventional scene images, the texture information of remote sensing images is more complicated. The main reason for sophisticated texture features is the variation in orientation, scale, and shape of objects presented in

the image. In addition to these variations, the inherent property of remote sensing images is also quite different from the ordinary scene images. Precisely, remote sensing image, as one of the most representative overhead images, has no dominant left-right or up-down relationships. To classify a typical scene image, only the presence or absence of the main object needs to be considered. However, in the aerial scene classification task, an expectation is that the model is capable of assigning the correct label for a given image regardless of its absolute orientation. This sought-after property remains strictly constant under all transformations of the input data, which is so-called *invariance*.

Invariance can be directly encoded and considered to be the most effective method to mitigate the impact of variations of the input data. However, incorporating invariant information is challenging, even for the powerful CNN architectures. Precisely, off-the-shelf CNN architectures are only endowed with the minimal internal structures due to the costly computing of the optimization. These minimal intrinsic structures are capable of handling locally minor shifts but not global transformations. Data augmentation techniques (Tanner & Wong, 1987; Perez & Wang, 2017) are widely adopted to incorporate the prior knowledge of input data, but there is no guarantee that the invariance learned in the training stage is effectively generalized for the test data. Furthermore, it is difficult to quantify the predominate transformations and lacks the interpretability of feature maps. In contrast to the redundant approaches, such as data augmentation, one of the latest research lines is toward procuring the equivariance from equivariant CNNs (T. Cohen & Welling, 2016; T. S. Cohen & Welling, 2016; Dieleman, De Fauw, & Kavukcuoglu, 2016; Henriques & Vedaldi, 2017). The basic idea of these methods is to learn the *transformation-equivariant CNN* by constructing features in a linear G-space and then derive an invariant subspace by employing the appropriate pooling method (e.g., the coset pooling). These methods can detect co-occurrences of

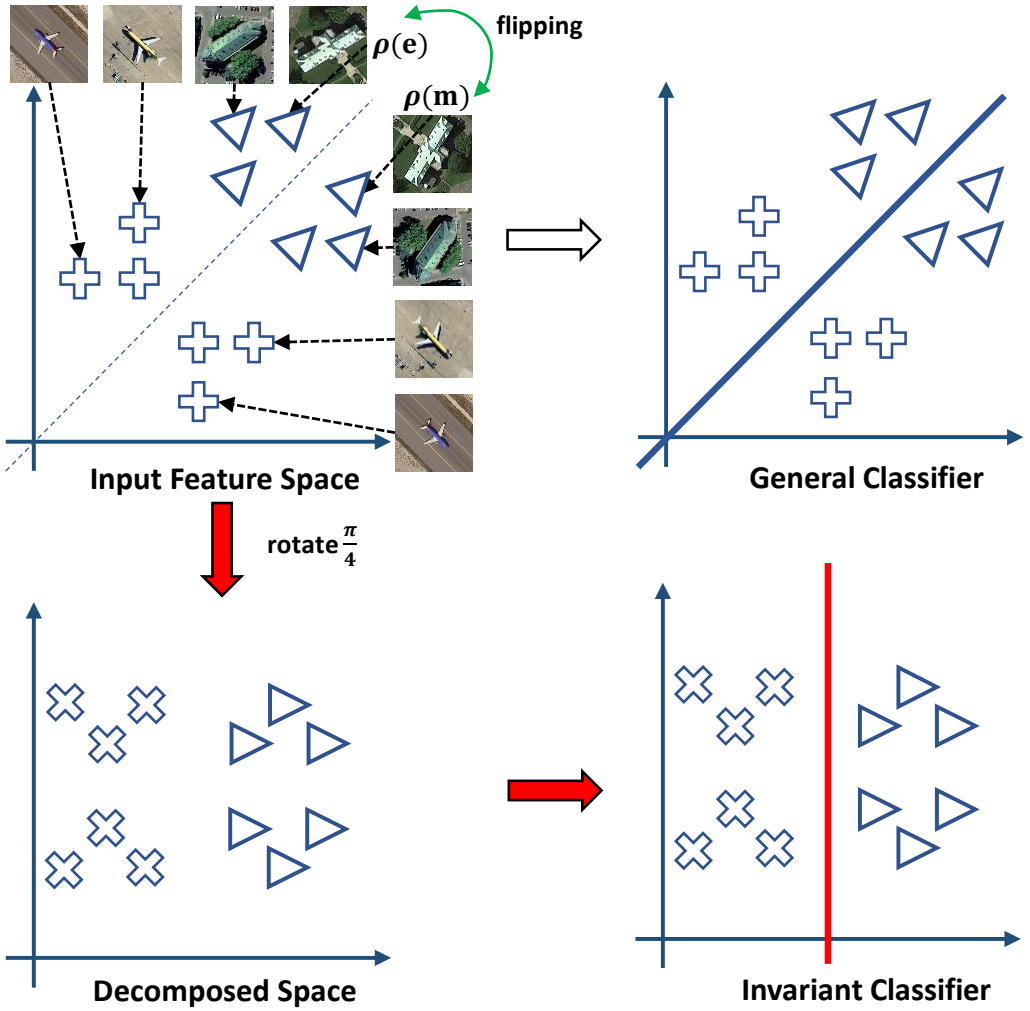


Figure 6.1: Solely flipping the input image may render conventional classifiers inoperable. Combined with the rotation transformation, a new orthogonal representation space can be formed. Then, it can generate a trivial representation from the space and leverage it to train an invariant classifier.

features at any *positions* in a standard CNN architecture, and any preferred *poses* in a G-space, but the computational cost scales dramatically with the increasing cardinality of the group.

To address the shortcomings of the aforementioned approaches, a novel framework is proposed to derive the transformation-invariant subspace from a finite

linear G-group space, which allows group actions to be directly applied to the raw image. As shown in Figure 6.1, merely flipping the local feature space can render the traditional classifier fail to work. Through looking insight into the flipping operation, it can be expressed by the permutation matrices. The expression of permutation matrices implies two primary properties: the flipping operation acts orthogonally at the local pixel and prevents images from distortion during transformation. Then, it becomes feasible to construct a transformation group G where all the decomposed spaces are orthogonal to each other (i.e., D4 group in this case). An invariant feature space can be sought through using the reducible decomposition of the representations of G-space. Namely, it allows decomposing the action of G into the direct sum of irreducible representations and results in a locally invariant subspace that serves to train an invariant classifier.

The orthogonal transformations prevent the pixel value shifting in the process of transforming but cannot avoid the changes of pixel locations. To alleviate the effect of pixel position changes, it considers the fact that the reducible decomposition of the representation conforms to the group action of G-space, thereby the tensor product of irreducible representations can be calculated to form a global representation. The tensor representation contains more discriminative information than the conventional first-order feature but suffers from the high-dimensional problem. Considering that the second-order feature representation is a covariance matrix (i.e., symmetric positive definite (SPD) matrix), the weight matrix can be forced to be a row full-rank matrix in which all elements reside on a Stiefel manifold. In this way, it can produce a compact space while maintaining the geometry of the SPD manifold. The contributions can be summarised as follows:

- It proposes a unified paradigm and proves its effectiveness in handling the challenges of RSSC tasks.

- It derives an invariant classifier from the learned weights of the trivial tensor representation with the guarantee of being invariant under the finite *G*-group actions.

- It introduces a way of imposing orthogonal constraints on the weight matrix to effectively map the high-dimensional SPD manifolds into new compact manifolds.

- Extensive experiments are conducted on four aerial scene image datasets and achieved state-of-the-art performance.

6.2 Preliminary Notions and Definitions

It will use calligraphic typeface $\mathcal{X}$ and $\mathcal{F}$ to denote the input image and the deep CNN features, respectively. A group $G = (\mathcal{X}, \bullet)$ is the pair of a set $\mathcal{X}$, together with an operation $\bullet : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{X}$ (also known as group law) that satisfies the group axioms of closure, associativity, identity and invertibility. The number of elements in a finite $\mathcal{X}$ is denoted as $|\mathcal{X}|$. A homomorphism is a map from a group *G* to the group of automorphisms of a vector space *V* that preserves group action operations, $\rho(g_1) \bullet \rho(g_2) = \rho(g_1 \bullet g_2), \forall g_1, g_2 \in G$ and exists the d-dimensional identity matrix $\rho(e) = 1_{d\times d}$. For a concrete example, $\rho : G \rightarrow \mathrm{GL}(V)$ is a homomorphism and also called a representation, where GL is the general linear space. A representation is named a trivial representation if and only if it maps all $g \in G$ to $1_{d\times d}$ (e.g., one-dimensional trivial representation is denoted as **1**). Similarly, the representation is called a unitary representation or orthogonal representation when all $\rho(g)$ are unitary matrices or orthogonal matrices. The space of intertwining operator is written as $\mathrm{Hom}_{\mathcal{X}}(\rho, \rho^{'})$ which implies that there is a linear operator $L : \mathbb{C}^d \rightarrow \mathbb{C}^{d^{'}}$ that satisfies $L \bullet \rho(g) = \rho'(g) \bullet L$. If *L* is a bijective function that satisfies $L \in \mathrm{Hom}_{\mathcal{X}}(\rho, \rho^{'})$, we will write it as $\rho \simeq \rho^{'}$. Given two representations

(ρ, V_1) and (σ, V_2) of the same group *G*, the direct sum of these two representations is given as $\rho \oplus \sigma : G \rightarrow$GL$(V_1 \oplus V_2)$ with regarding G as block-diagonal form of $G \times G$. According to Schur's Lemma, $\text{Hom}_G(\rho_1, \rho_2) = \{0\}$ if ρ_1 and ρ_2 are not isomorphic or 1-D when they are isomorphic. If ρ and σ are in tensor spaces, the tensor representation will be denoted as $\rho \otimes \sigma$. The character function $\mathcal{X}_\rho$ that maps *G* into a finite-dimensional vector space over a filed *F* is given by $\mathcal{X}_\rho(g) = \text{tr}(\rho(g))$, where tr$(\cdot)$ is the trace operation. The degree of a representation ρ is the dimension of its representation space *V* and we denote it as dim(ρ).

6.3 Method

6.3.1 Transformation-Equivariant Networks

In deep learning models, the transformation-equivariant preserves the capacity to capture various useful transformations. An example is the translation-equivariant in convolution layers, which can be exploited in any layers of the deep CNN architecture. Given an input image $\mathcal{X}$, the transformation equivariant can be regarded as seeking a unique $T_g^{'} \in G^{'}$ that satisfies:

$$\Phi(T_g(\mathcal{X})) = T_g^{'}(\Phi(\mathcal{X})) \tag{6.1}$$

where $T_g^{'}$ is an action in a group structure $G^{'}$ and Φ denotes the feature mapping function. For brevity, it is usually written as $\Phi(T_g(\mathcal{X})) = T_g(\Phi(\mathcal{X}))$ since $T_g^{'} = T_g$ and then $G^{'} = G$. However, the former format is preferred because $\Phi(\mathcal{X})$ and $T_g(\mathcal{X})$, perhaps, lie in the different domains. Two strategies can be derived from the definition to achieve the equivariance to transformations. On the one hand, $T_g^{'}(\Phi(\mathcal{X}))$ indicates an explicit way to learn equivariance of transformations by transforming kernels or feature maps extracted from the input image,

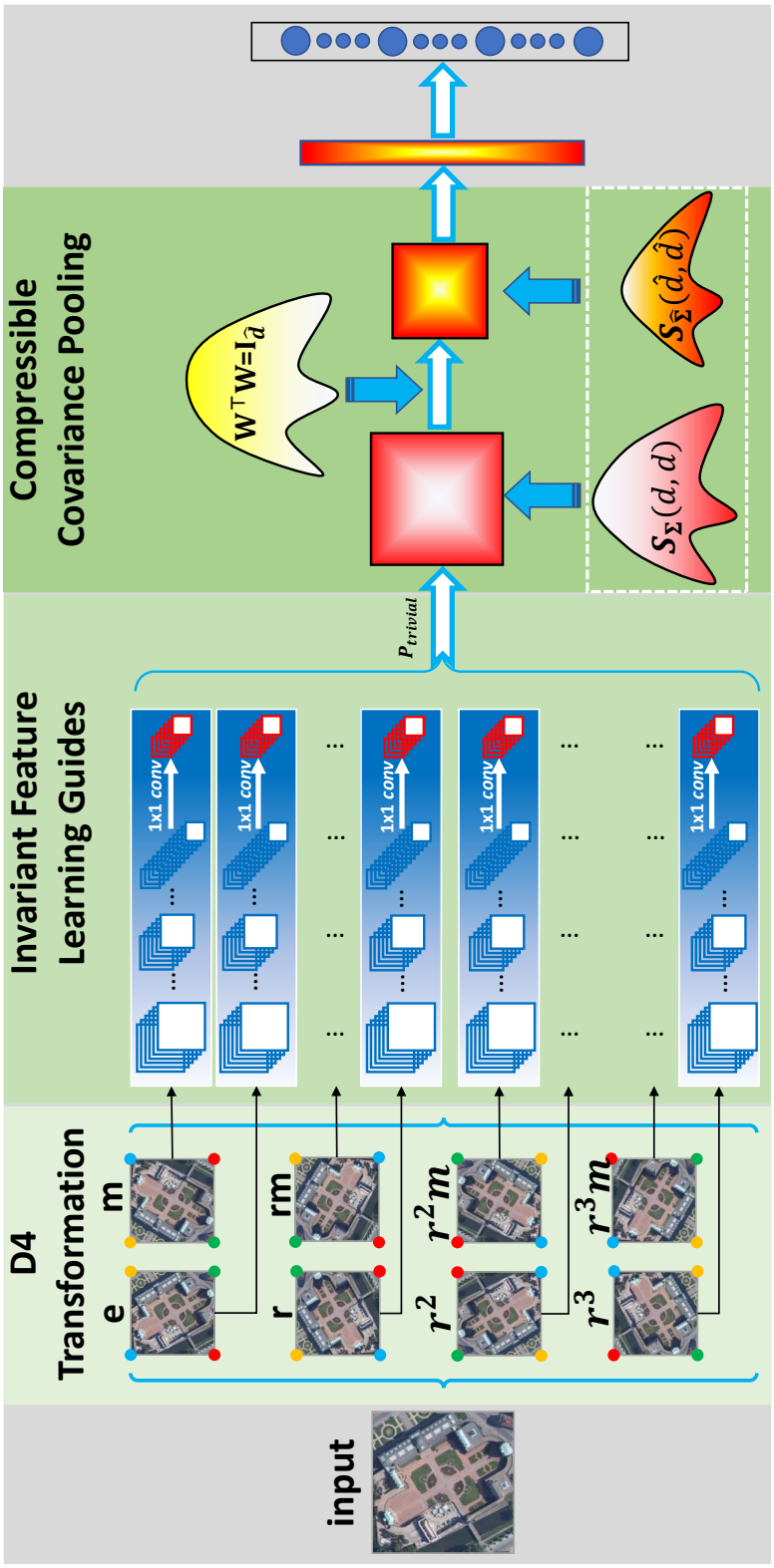


Figure 6.2: An overview of the proposed IDCCP architecture. Given an input image, it will be used to generate multiple copies according to the D4 principle. Then, each copy will be fed into a subnetwork of Siamese-style CNNs to extract feature (Note:1×1 *conv* is only adopted in the Siamese architecture with ResNet50 as the backbone). $P_{trivial}$ is the projection layer to produce a trivial representation. Subsequently, orthogonal weights are adopted to compress high-dimensional manifold $S_\Sigma(d,d)$ to a compact manifold $S_{\hat{\Sigma}}(\hat{d},\hat{d})$. The resulting features will be flattened and fed into the classifier to generate predictions.

such as (T. Cohen & Welling, 2016; Dieleman et al., 2016). However, these methods are generally inefficient because they require complicated permutations of each convolution kernel in all convolutional layers and need retraining on large-scale datasets. In addition, they neglect the manipulation of shared weights between convolution kernels, which makes them difficult to transfer or scale to new challenging tasks. $\Phi(T_g(\mathcal{X}))$, on the other hand, offers an option to achieve transformation-equivariant by transforming input image directly. However, this branch arises less attention or has been referred to data augmentation method (Perez & Wang, 2017; Tanner & Wong, 1987).

To cope with the abovementioned problems, a novel framework is proposed to achieve equivariance by directly transforming input images and extracting the corresponding features with multiple CNNs. As shown in Figure 6.2, it first transforms the input image according to a D_4 transformation group that consists of image reflections and rotations by multiples of $90°$. The main reason for choosing the D_4 group is that the group is a regular and symmetrical polygon. In other words, it implies that any actions in a D_4 group can prevent the image transformation from distortion. Once the transformed images have been obtained, it will focus on seeking for an architecture that is effective to retain the group structure during the feature extraction. The naive way is to adopt as many CNN networks as the order of the D_4 group. However, this method will exponentially increase the computational burdens. To address this problem, it exploits a Siamese-style architecture for feature extraction, which allows the weights to be shared among all subnetworks. To show how it works for preserving group structure, the following proposition and the corresponding proof are given.

Proposition 1. *Let $\mathcal{X}$ be a set of images with the structure of symmetry square dihedral D_4 group, so $D_4 = \langle r, m : r^4 = m^2 = e, rm = mr^{-1} \rangle$ and let $\Phi : Siam(\mathcal{X}) \rightarrow \mathcal{F}$ be the feature extraction function. Then, the resulting features $\mathcal{F}$ will be*

given in the structure of the D_4 group.

Proof. Let $T_g(\mathcal{X})$ be an action result of input D_4 group image and K be the convolution kernel of general CNN. The convolution operation on a 2-D image can be denoted as:

$$[T_g(\mathcal{X}) * K](i,j) = \sum_u \sum_v T_g(\mathcal{X})(u,v) K(i-u, j-v), \tag{6.2}$$

Then, it can use $u \to u+t, v \to v+t$ (i.e., the substitution does not change the summation bounds since rotation is a symmetry of the sampling grid) to prove the relationships between convolution and translation. The details are as follows:

$$\begin{aligned}
[\Phi_t T_g(\mathcal{X})] * K(i,j) =& \sum_u \sum_v T_g(\mathcal{X})(u-t, v-t) K(i-u, j-v) \\
=& \sum_u \sum_v T_g(\mathcal{X})(u,v) K(i+t-u, j+t-v) \\
=& \sum_u \sum_v T_g(\mathcal{X})(u,v) K(i-(u-t), j-(v-t)) \\
=& \Phi_t [T_g(\mathcal{X}) * K](i,j)
\end{aligned} \tag{6.3}$$

Analogically, it can derive equivariance such as reflection or flip by using the communicative: $u \to -u, v \to -v$ and write it as:

$$\begin{aligned}
[\Phi_m T_g(\mathcal{X})] * K(i,j) =& \sum_u \sum_v T_g(\mathcal{X})(-u,-v) K(i-u, j-v) \\
=& \sum_u \sum_v T_g(\mathcal{X})(u,v) K(u-i, v-j) \\
=& \Phi_m [T_g(\mathcal{X}) * \Phi_{-m} K](i,j)
\end{aligned} \tag{6.4}$$

The conventional convolution operations hold equivariant property for translation and flip, but not be equivariant to other isometric sampling methods, such as rota-

tion. To proof the rotation equivariant, it needs the definition of $F_r T_g(\mathcal{X})(u,v) = F(r_{-1}(u,v))$ and the substitution $(u,v) = r(u,v)$:

$$\begin{aligned}
[\Phi_r T_g(\mathcal{X})] * K(i,j) =& \sum_u \sum_v \Phi_r T_g(\mathcal{X})(u,v) K(i-u, j-v) \\
=& \sum_u \sum_v T_g(\mathcal{X})(r^{-1}u, r^{-1}v) K(i-u, j-v) \\
=& \sum_u \sum_v T_g(\mathcal{X})(u,v) K(r(i-u), r(j-v)) \\
=& \sum_u \sum_v T_g(\mathcal{X})(u,v) \Phi_{r^{-1}} K(i-r^{-1}u), (j-r^{-1}v)) \\
=& \Phi_r [T_g(\mathcal{X}) * \Phi_{r^{-1}} K](i,j)
\end{aligned} \tag{6.5}$$

□

A similar visual proof of the abovementioned relationships between convolution and transformations can be found in (Dieleman et al., 2016). Furthermore, the pooling function that exists in CNN architecture has been proven to be commuted with the group action (T. Cohen & Welling, 2016). Hence, if an ordinary Siamese-style CNN learns transformed copies of the input image, the stack of feature maps will attain the same group structure as the transformed copies. It must be emphasised that the orientations of rotation may appear in either clockwise or counter-clockwise depending on the implementation environment. If let T_g and $T_g^{'}$ be actions on the sets of $\mathcal{X}$ and $\mathcal{F}$ that satisfy $T_{g_1 g_2} = T_{g_1} \bullet T_{g_2}$ and $T^{'}_{g_1 g_2} = T^{'}_{g_1} \bullet T^{'}_{g_2}$, the transformations T_g and $T_g^{'}$ will induce actions $\mathbf{T}_g$ and $\mathbf{T}_g^{'}$ on the space of $\mathcal{X}$ and $\mathcal{F}$. The difference between two spaces of $\mathcal{X}$ and $\mathcal{F}$ is the space field rather than the group structure. Thus, the transformation group of the input image can be preserved by using the Siamese-style CNNs.

Table 6.1: The irreducible representations of the roto-reflection D4 group (T. S. Cohen & Welling, 2016).

Irrep.	e	r	r^2	r^3	m	mr	mr^2	mr^3
$\rho_{1,1}$	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
$\rho_{1,-1}$	[1]	[1]	[1]	[1]	[-1]	[-1]	[-1]	[-1]
$\rho_{-1,1}$	[1]	[-1]	[1]	[-1]	[1]	[-1]	[1]	[-1]
$\rho_{-1,-1}$	[1]	[-1]	[1]	[-1]	[-1]	[1]	[-1]	[1]
ρ_2	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$

6.3.2 Invariant Feature Learning Guides

Learning invariant features, as a particular case of learning equivariant features, is essential for many recognition tasks. It turns out that adopting a Siamese-style architecture can preserve the structure of the predefined transformations of inputs $\mathcal{X}$. The next step is to find the invariant subspace from the generated feature space $\mathcal{F}$. Because it assumes that $\rho(g)$ are all orthogonal representations, it means that they are also unitary representations that cannot be decomposed, thus enabling us to derive invariant subspaces from the perspective of irreducible representations. Taking the D4 group as an example, its irreducible representations have been summarised in TABLE. 6.1 where the orthogonality of the characters of representations can be verified.

Considering the fact that orthogonal representation is a real analogy of unitary representation, the whole representation space can be formed by calculating the direct sum of all irreducible representations. For example, given a representation ρ, it can be decomposed by $\rho \simeq \lambda_1\tau_1 \oplus \lambda_2\tau_2 \oplus \ldots \lambda_T\tau_T$. As the characteristic function of ρ has been defined as $\mathcal{X}_\rho(g) = \text{tr}(\rho(g))$ with the matrix form $\rho(g)$ of ρ, the corresponding coefficients can be computed by using $\lambda_t = \frac{1}{|G|}\sum_{g\in G}\overline{\mathcal{X}_\rho(g)}\mathcal{X}_{\tau_t(g)}$. The operator that projects ρ to $n_t\tau_t$ can be achieved by following $P_{\tau_t} = \text{dim}(\tau_t)\sum_{g\in G}\overline{\mathcal{X}_{\tau_t}(g)}\rho(g)$. Since $\mathcal{X}_\mathbf{1}(g) = 1$, it can obtain the trivial

representation by calculating the average of $\rho(g)$:

$$P_{trivial} = \frac{1}{|G|} \sum_{g \in G} \rho(g). \tag{6.6}$$

When it uses the above trivial representation to train the classifier, the learned weights lie in the subspace of the entire action space (i.e., the average of all $\rho(g)$ is a subspace that is invariant to T-actions). To reveal the role of learning the trivial representation, it gives the following theorem.

Theorem 1. *Given an input sample space* $\mathcal{S} = \mathcal{X} \times \mathcal{Y} = \{(x_n, y_n)\}_{n=1}^{N} \in \mathbb{R}^d$, *which is structured by a set of orthogonal transformation group G. Then the solution of minimizing the L2 regularised convex loss function:*

$$\min_{w,b} \frac{1}{N} \sum_{n=1}^{N} l\left(\left\langle w^\top x_n + b \right\rangle_{\mathbb{R}}, y_n\right) + \frac{\lambda}{2}||w||^2 \tag{6.7}$$

lies in a vector subspace that is G-invariant, and the general error of the algorithm may be up to a factor $\sqrt{T}$ *smaller than the general error of a non-invariant learning algorithm.*

Proof. The proof of G-invariant has been given by (Mukuta & Harada, 2019) from the irreducible representation in the complex space, while Sokolic *et al.* (Sokolic, Giryes, Sapiro, & Rodrigues, 2017) exploited a covering number to prove the general error of the invariant algorithm. □

For more details, it refers readers to (Mukuta & Harada, 2019) and(Sokolic et al., 2017) and reference herein. This theorem also induces essential properties of the trivial representation. Formally, for all $g \in G$, it can have:

$$\rho(g)w = w \Leftrightarrow \rho(g)w \subseteq w \; and \; P_{trivial}w = w \tag{6.8}$$

Table 6.2: Tensor product of irreducible representation of the roto-reflection D4 group (Mukuta & Harada, 2019).

Irrep.	$\rho_{1,1}$	$\rho_{1,-1}$	$\rho_{-1,1}$	$\rho_{-1,-1}$	ρ_2
$\rho_{1,1}$	$\rho_{1,1}$	$\rho_{1,-1}$	$\rho_{-1,1}$	$\rho_{-1,-1}$	ρ_2
$\rho_{1,-1}$	$\rho_{1,-1}$	$\rho_{1,1}$	$\rho_{-1,-1}$	$\rho_{-1,1}$	ρ_2
$\rho_{-1,1}$	$\rho_{-1,-1}$	$\rho_{-1,1}$	$\rho_{1,-1}$	$\rho_{1,1}$	ρ_2
$\rho_{-1,-1}$	$\rho_{-1,-1}$	$\rho_{-1,1}$	$\rho_{1,-1}$	$\rho_{1,1}$	ρ_2
ρ_2	ρ_2	ρ_2	ρ_2	ρ_2	$\rho_{1,1}\bigoplus\rho_{1,-1}\bigoplus\rho_{-1,1}\bigoplus\rho_{-1,-1}$

The aforementioned theorem proves the G-invariance of augmented space contributes to reducing the general error of the learning algorithm but neglects to handle the massive parameters of the learning algorithm and the high-dimensional feature space. Instead, it will deploy the learning algorithm to a shared-weights Siamese-style network and supply an effective compressible tensor representation in the following section.

6.3.3 Compressible Covariance Pooling

Covariance pooling, as a form of the second-order statistics feature, aims to establish the correlation between the spatial and channels of local CNN features to aggregate more distinguishing information. Suggested by (P. Li et al., 2017; Acharya et al., 2018), and (P. Li et al., 2018), it performs the second-order pooling in the form of a scatter covariance matrix:

$$\boldsymbol{\Sigma} = \frac{1}{\mathrm{hw}} \sum_{i=1}^{\mathrm{hw}} P\left((\mathbf{f}_i - \bar{\mathbf{f}})(\mathbf{f}_i - \bar{\mathbf{f}})^\top\right) = \frac{1}{\mathrm{hw}} P_{trivial}\left(\mathbf{F}\bar{\mathbf{I}}\mathbf{F}^\top\right). \tag{6.9}$$

where w and h are feature width and height. $P_{trivial}$ is projection function that it has been introduced before. $\bar{\mathbf{f}} = \frac{1}{\mathrm{hw}} \sum_{i=1}^{\mathrm{hw}} \mathbf{f}_i$ is the mean of feature vectors. $\bar{\mathbf{I}} = \mathbf{I} - \frac{1}{\mathrm{hw}}\mathbf{1}\mathbf{1}^\top \in \mathbb{R}^{\mathrm{hw}\times\mathrm{hw}}$ is the centering matrix, where $\mathbf{I}$ and $\mathbf{1}$ denote the

identity matrix and the all-ones matrix, respectively.

Since the projection function $P_{trivial}$ is employed in the tensor space, the tensor product representation needs to be given concerning the irreducible representation in the D4 group. According to the distributive property of tensor product representation (e.g., given two representations ρ and σ, it satisfies $(\rho_1 \oplus \rho_2) \otimes (\rho_3 \oplus \rho_4) = (\rho_1 \otimes \rho_3) \oplus (\rho_2 \otimes \rho_3) \oplus (\rho_1 \otimes \rho_4) \oplus (\rho_2 \otimes \rho_4)$ and $\mathcal{X}_{\rho\otimes\sigma}(g) = \mathcal{X}_{\rho}(g)\mathcal{X}_{\sigma}(g)$, it can calculate the tensor product representations of irreducible representations. Combining the fact that the tensor product of irreducible representation and 1-D representation is irreducible, it allows decomposing tensor products of D4 group and present the results in TABLE 6.2. For verifying the results, it takes two representations $\rho(e)$ and $\rho(m)$ in TABLE 6.1 as an example, and the corresponding tensor product representations become 4-D vectors such that $\rho(e) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ and $\rho(m) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$, respectively.

The obtained covariance matrix can be regarded as a form of representation, which is capable of capturing more information than the ordinary first-order statistical feature. However, its shortcomings are also obvious. The first and foremost drawback of such covariance pooling is its high dimensionality. Taking VGG architecture (Simonyan & Zisserman, 2014) as an example, the dimension of the vectorized covariance matrix generated from the last convolution layer will be 2^{18}. Rank deficiency is another weakness of covariance pooling because the number of CNN channels is much larger than the product of feature height and width.

The abovementioned reasons promote me to discover a compact form of covariance pooling. Considering that the covariance matrix is an SPD matrix, it is necessary to retain the geometry of the SPD manifold while reducing the matrix dimension. To accomplish this goal, it presents a method based on the following proposition.

Proposition 2. *Let* $\mathbf{\Sigma} \in \mathbb{R}^{d\times d}$ *be the covariance matrix generated from the last convolution layer and* $\boldsymbol{W} \in \mathbb{R}^{d\times \hat{d}}$ *be an orthogonal, row full rank matrix with* $\hat{d} < d$*. Then, the bilinear form of transformation matrix* $\boldsymbol{W}$ *maps* $\mathbf{\Sigma}$ *to a valid SPD matrix* $\hat{\mathbf{\Sigma}} \in \mathbb{R}^{\hat{d}\times\hat{d}}$.

Proof. The bilinear mapping function can be generally denoted as $\mathcal{B} : \mathbf{\Sigma} \times \mathbf{W} \rightarrow \hat{\mathbf{\Sigma}}$. In order to express it more accurately, it can be rewritten in the form of: $\hat{\mathbf{\Sigma}} = \mathbf{W}^\top \mathbf{\Sigma} \mathbf{W}$. Due to the orthogonality and row full rank of transformation matrix $\mathbf{W}$, the elements generated by transformation weights are naturally located on a non-compact Stiefel manifold $\mathcal{S}^*(\hat{d}, d) \triangleq \left\{ \mathbf{W} \in \mathbb{R}^{d\times\hat{d}} : \mathbf{W}^\top\mathbf{W} = \mathbf{I}_{\hat{d}} \right\}$ and can be transformed into a compact manifold $\mathcal{S}(\hat{d}, d)$. Then, the resulting matrix $\hat{\mathbf{\Sigma}} \in \mathbb{R}^{\hat{d}\times\hat{d}}$ is a valid but very compact SPD matrix because $\hat{d} < d$. □

The abovementioned claim and proof are trivial, but it can be regarded as guides to convert those high-dimensional SPD matrices $\mathbf{\Sigma}$ to new, low-dimensional SPD matrices $\hat{\mathbf{\Sigma}}$ with $\hat{d} < d$, $\hat{\mathbf{\Sigma}} \in Sym^+_{\hat{d}}$. Compared with most existing methods that directly map SPD manifold into the Euclidean space (Lin et al., 2015; Lin & Maji, 2017; Kong & Fowlkes, 2017; Gao et al., 2016; P. Li et al., 2017, 2018; S. Wang, Guan, & Shao, 2020; S. Wang et al., -), the proposed method can certainly preserve the inherent manifold structure of high-dimensional SPD matrices. However, given a non-compact Stiefel manifold, a matrix form of writing linearly independent column vectors (i.e., $\hat{d}$-frames), has no closed-form of geodesic curves. In other words, it is infeasible to optimize on the manifold directly (Fiori,

2010). The relatively tractable strategy is to endow non-compact Stiefel manifold with a pseudo-Riemannian manifold so that the gradient of geodesic distance can be derived from a smooth manifold and present in a closed form. To this end, the orthogonal constraints will be imposed on $\mathbf{W}$ (precisely speaking, it is semi-orthogonal matrix under this scenario). Consequently, the entities of transformation weight $\mathbf{W}$ reside on a compact Stiefel manifold $\mathcal{S}(\hat{d}, d)$, which allows us to find the optimal solutions of the weight matrix.

Furthermore, the abovementioned function for feature dimension reduction can also be regarded as an intertwining operator when it imposes orthogonal constraints $\mathbf{W}^\top \mathbf{W} = \mathbf{I}_{\hat{d}}$ on transformation weight. Recalling the introduction of intertwining in preliminaries, the produced projection space is also the representation space. Thus, the low-dimensional representation can be achieved by imposing low-rank constraints on weight $\mathbf{W}$. Specifically, it can first line up the eigenvalues of $\mathbf{\Sigma}$ by employing eigenvalue decomposition function and then find the elements with the larger variance to retain. However, matrix decomposition often requires more computational costs and time-consuming (P. Li et al., 2017; S. Wang, Guan, & Shao, 2020). Rather than using cumbersome decomposition functions, the bilinear mapping function can transform the input SPD matrix into a new, low-dimensional SPD matrix that is useful for subsequent optimisation.

6.3.4 Invariant Classifier Training

The compressible covariance pooling method has been described in the last section, which maps the high-dimensional manifold to a low-dimensional compact manifold. Different from the mainstream methods, the proposed algorithm deduces a rank efficient representation on manifold space while retaining the inherent manifold structure.

The elements of the resulting low-dimensional SPD matrices reside on the Riemannian manifold, which needs to be transformed into the Euclidean space so that the distance between different elements can be measured by the Euclidean operations. The natural choice is to employ the logarithm of SPD matrices since it reflects the true geodesic distance of the manifold. Furthermore, the logarithm of an SPD matrix will give rise to the matrix with a Lie group, and then, all Euclidean operations can be adopted. However, the logarithm will change the magnitude order of small eigenvalues and usually not robust in practice (P. Li et al., 2017; Lin & Maji, 2017). Instead, it will be committed to learning more robust square root normalization of matrices, which can be considered as the approximate Riemannian geometry in covariance matrices (P. Li et al., 2017).

It is well-known that any SPD matrix has a unique square root, which can be obtained by using SVD or EIG. Although SVD or EIG yield the accurate solution of the square root of a matrix, they are time-consuming and often cannot be well-supported by GPU acceleration (P. Li et al., 2017; S. Wang, Guan, & Shao, 2020). Inspired by (P. Li et al., 2018), the iSQRT-COV approach is employed, which uses a variation of the Newton method to iteratively calculate the square root of the matrix. Especially, given $\mathbf{C}_0 = \frac{\hat{\mathbf{\Sigma}}}{\text{tr}(\hat{\mathbf{\Sigma}})}$ and $\mathbf{D}_0 = \mathbf{I}$, the Newton-Schulz method (P. Li et al., 2018) allows computing the square root $\mathbf{C}$ of $\hat{\mathbf{\Sigma}}$ by using the following iterations:

$$\begin{aligned} \mathbf{C}_j &= \frac{1}{2}\mathbf{C}_{j-1}(3\mathbf{I} - \mathbf{D}_j\mathbf{C}_{j-1}), \\ \mathbf{D}_j &= \frac{1}{2}(3\mathbf{I} - \mathbf{D}_j\mathbf{C}_{j-1})\mathbf{D}_j, \end{aligned} \tag{6.10}$$

where $j = 1, \ldots, J$ is the iteration steps. With the condition of $||\hat{\mathbf{\Sigma}} - \mathbf{I}|| < \frac{1}{2}$, $\mathbf{C}_j$ and $\mathbf{D}_j$ are guaranteed to quadratically converge to $\mathbf{C}^{\frac{1}{2}}$ and $\mathbf{C}^{-\frac{1}{2}}$, respectively. Briefly, it means that $\mathbf{C}^2 = \hat{\mathbf{\Sigma}}$ and $\mathbf{C} = \mathbf{\Psi}\mathbf{\Lambda}\mathbf{\Psi}^\top$ described in EIG format, where

$\mathbf{C} = \mathbf{\Psi}$ is an orthogonal matrix and $\mathbf{\Lambda} = (\boldsymbol{\lambda}_1^{\frac{1}{2}}, \boldsymbol{\lambda}_2^{\frac{1}{2}}, \ldots, \boldsymbol{\lambda}_{d'}^{\frac{1}{2}})$ is a diagonal matrix.

Once the square-root of the SPD matrix is obtained, the Euclidean operations can be used to measure the distance of elements on the flatted Stiefel Manifold. Considering the fact that the initialisation of $\mathbf{C}_0$ has changed the magnitude of the matrix value, we then use $\hat{\mathbf{C}} = \sqrt{\text{tr}(\hat{\mathbf{\Sigma}})}\mathbf{C}_j$ to counteract such changes (P. Li et al., 2018). The resulting matrix $\hat{\mathbf{C}}$ can be used to train the classifier. Let us suppose that $\hat{\mathbf{W}}$ be the corresponding weight matrix of $\hat{\mathbf{C}}$. The objective function in **Theorem 1.** can be rewritten by substituting w with $\hat{\mathbf{W}}$, and then yield the following expression:

$$\begin{aligned} &\min_{\hat{\mathbf{W}},b} \frac{1}{N}\sum_{i=1}^{N} l\left(\left\langle \text{tr}\left(\hat{\mathbf{W}}^\top \hat{\mathbf{C}}\right) + b\right\rangle_{\mathbb{R}}, y_i\right) + \frac{\lambda}{2}||\hat{\mathbf{W}}||^2 \\ =&\min_{\hat{\mathbf{W}},b} \frac{1}{N}\sum_{i=1}^{N} l\left(\left\langle \text{tr}\left(\hat{\mathbf{W}}^{(\mathbf{1})} \hat{\mathbf{C}}^{(\mathbf{1})}\right) + b\right\rangle_{\mathbb{R}}, y_i\right) + \frac{\lambda}{2}||\hat{\mathbf{W}}||^2. \end{aligned} \tag{6.11}$$

For brevity, the transpose operator $\top$ in the last line is omitted because of $\hat{\mathbf{W}} = \hat{\mathbf{W}}^\top$. The final result highlights the key advantage of our classifier, which avoids the direct optimization on the original high-dimensional weights $\mathbf{W}$.

6.3.5 Back-propagation

Stochastic gradient descent (SGD), as one of the most popular gradient calculation algorithms, is widely adopted for training deep CNNs. In this scenario, it will employ SGD to compute the gradient of the given objective function with respect to the transformation matrix $\mathbf{W}$ and the second-order statistical feature $\mathbf{\Sigma}$. Let the derivative of $\hat{\mathbf{C}}$ be $\left(\frac{\partial l}{\partial \hat{\mathbf{C}}}\right)$ that derives from the Softmax layer. Then, it can use the

following chain rules to calculate the matrix derivatives:

$$\begin{aligned} \text{tr}\left(\left(\frac{\partial l}{\partial \hat{\mathbf{C}}}\right)^{\top} d\hat{\mathbf{C}}\right) &= \text{tr}\left(\left(\frac{\partial l}{\partial \hat{\mathbf{C}}_J}\right)^{\top} d\hat{\mathbf{C}}_J + \left(\frac{\partial l}{\partial \hat{\mathbf{\Sigma}}}\right)^{\top} d\hat{\mathbf{\Sigma}}\right), \\ \text{tr}\left(\left(\frac{\partial l}{\partial \mathbf{W}}\right)^{\top} d\mathbf{\Sigma}\right) &= \text{tr}\left(\left(\frac{\partial l}{\partial \hat{\mathbf{\Sigma}}}\right)^{\top} d\hat{\mathbf{\Sigma}} + \left(\frac{\partial l}{\partial \mathbf{W}}\right)^{\top} d\mathbf{\Sigma}\right), \end{aligned} \tag{6.12}$$

where $d\hat{\mathbf{C}}$ is the variation of $\hat{\mathbf{C}}$. According to expression at the first line, it can derive the derivative of $\hat{\mathbf{\Sigma}}$ through some manipulations. For more details, it refers readers to (P. Li et al., 2018) and reference it herein. Once $\frac{\partial l}{\partial \hat{\mathbf{\Sigma}}}$ has been obtained, it can be used to compute the gradient for updating $\mathbf{W}$.

As described in 6.3.3, it projects all elements on the Stiefel manifold $\mathcal{S}(\hat{d}, d)$ into the Euclidean space so that the Euclidean operations can be used to measure the distance between projected elements. However, directly using the back propagation rules in the Euclidean space to calculate the gradient of the Stiefel manifold cannot guarantee that the orthogonality of weights $\mathbf{W}$. To this end, it introduces the Euclidean inner product in the tangent space of the Stiefel manifold as a new strategy for updating the gradient of the covariance pooling. Therefore, the Stiefel manifold is transformed into a Riemannian manifold so that it can borrow the method of optimising the Riemannian manifold to calculate the gradient of the Stiefel manifold. To better explain this, it provides the following statement.

Lemma 2. *Let $\boldsymbol{M}_1 = \boldsymbol{S}\boldsymbol{A}_1 + \boldsymbol{S}_{\perp}\boldsymbol{B}_1, \boldsymbol{M}_2 = \boldsymbol{S}\boldsymbol{A}_2 + \boldsymbol{S}_{\perp}\boldsymbol{B}_2$ are two matrices on the tangent space of Stiefel manifold $\mathcal{S}(\hat{d}, d)$ and let $\langle \boldsymbol{M}_1, \boldsymbol{M}_2 \rangle_e = tr(\boldsymbol{M}_1^{\top}\boldsymbol{M}_1)$ be the Euclidean inner product over the ambient space $\mathbb{R}^{d\times\hat{d}}$, Then the Euclidean metric weighs the coefficients of the basis of $\boldsymbol{A}_1^{\top}\boldsymbol{A}_2$ and $\boldsymbol{B}_1^{\top}\boldsymbol{B}_2$ unequally.*

Proof. Since matrices $\mathbf{M}_1, \mathbf{M}_2$ belong to the tangent space of the Stiefel manifold $\mathcal{S}(\hat{d}, d)$, the matrices $\mathbf{A}_1, \mathbf{A}_2$ must be skew symmetric matrices of dimension $\hat{d}\times\hat{d}$,

and $\mathbf{B}_1, \mathbf{B}_2$ are arbitrary matrices with dimension $(d - \hat{d}) \times d$. For more detailed information, it refers readers to (Absil, Mahony, & Sepulchre, 2009). Then, it allows the Euclidean metric to be rewritten in the following form:

$$\begin{aligned}\langle \mathbf{M}_1, \mathbf{M}_2 \rangle_e =& \text{tr}(\mathbf{B}_1^\top \mathbf{S}_\perp^\top + \mathbf{A}_1^\top \mathbf{S}_\perp)(\mathbf{S}\mathbf{A}_2 + \mathbf{S}_\perp \mathbf{B}_2) \\ =& \text{tr}(\mathbf{B}_1^\top \mathbf{S}_\perp^\top \mathbf{S}\mathbf{A}_2 + \mathbf{B}_1^\top \mathbf{S}_\perp^\top \mathbf{S}_\perp \mathbf{B}_2 + \mathbf{A}_1^\top \mathbf{S}^\top \mathbf{S}\mathbf{A}_2 + \mathbf{A}_1^\top \mathbf{S}^\top \mathbf{S}_\perp \mathbf{B}_2) \\ =& \text{tr}(\mathbf{B}_1^\top \mathbf{B}_2 + \mathbf{A}_1^\top \mathbf{A}_2) = \text{tr}(\mathbf{A}_1^\top \mathbf{A}_2) + \text{tr}(\mathbf{B}_1^\top \mathbf{B}_2)\end{aligned} \tag{6.13}$$

where $\mathbf{S} \in \mathbb{R}^{d \times \hat{d}}, \mathbf{S}_\perp \in \mathbb{R}^{d \times (d-\hat{d})}$. Considering that the diagonal elements of $\mathbf{A}_1, \mathbf{A}_2$ and all elements of $\mathbf{B}_1, \mathbf{B}_2$ are the coefficients of the basis of $\mathbf{M}_1, \mathbf{M}_2$. Then, $\langle \mathbf{M}_1, \mathbf{M}_2 \rangle_e = \sum_{i>j} 2a^2(i,j) + \sum_{i,j} b^2(i,j)$, where it presents doubled on skew symmetric matrices. □

It can be seen that the Euclidean inner product $\langle \mathbf{M}_1, \mathbf{M}_2 \rangle_e = \text{tr}(\mathbf{M}_1^\top \mathbf{M}_1)$ cannot equally weigh the cardinality of the matrix. However, the Euclidean inner product naturally derives from the predefined Euclidean measurement method and is easy to implement in practice. When the Euclidean inner product is adopted, the corresponding gradient of the current points $\mathbf{W}^t$ on the Riemannian manifold $G_e^{\mathcal{S}} l(\mathbf{W}^t)$ can be obtained by:

$$G_e^{\mathcal{S}} l(\mathbf{W}^t) = \frac{\partial l}{\partial \mathbf{W}^t} - \mathbf{W}^t \left(\frac{\partial l}{\partial \mathbf{W}^t} \right)^\top \mathbf{W}^t, \tag{6.14}$$

where $\frac{\partial l}{\partial \mathbf{W}^t}$ is the normal component of the gradient in the Euclidean space, which can be obtained by using the second expression of Eq.(6.12) as:

$$\frac{\partial l}{\partial \mathbf{W}^t} = 2 \frac{\partial l}{\partial \hat{\mathbf{\Sigma}}} \mathbf{W}^t \mathbf{\Sigma}, \tag{6.15}$$

When it obtains the Riemannian gradient, it needs to seek the descent direction of

the gradient (i.e., the steepest gradient descent will be used in this scenario) and ensure that the new update points $\mathbf{W}^{t+1}$ are located on the Stiefel manifold. To achieve this, the QR-decomposition retraction is adopted $Z_{\mathbf{W}}(\xi) = qf(\mathbf{W} + \xi)$ which has been introduced in (Z. Huang & Van Gool, 2017; Edelman, Arias, & Smith, 1998; Absil et al., 2009). Here, $qf(\cdot)$ is the adjusted Q factors of the QR-decomposition and R factors in an upper triangular matrix with strictly positive elements on the diagonal. Thus, the decomposition is guaranteed to be unique and orthogonal. Through defining the learning rate as η, we can compute the new point by:

$$\mathbf{W}_{t+1} = qf\left(\mathbf{W}_t - \eta G_e^{\mathcal{S}} l(\mathbf{W}^t)\right), \tag{6.16}$$

Once the derivation of $\frac{\partial l}{\partial \mathbf{\Sigma}}$ has been achieved, it can derive the derivative for the input feature $\mathbf{F}$ with using:

$$\frac{\partial l}{\partial \mathbf{F}} = \left(\frac{\partial l}{\partial \mathbf{\Sigma}}\left(\frac{\partial l}{\partial \mathbf{\Sigma}}\right)^{\top}\right)\hat{\mathbf{I}}\mathbf{F}. \tag{6.17}$$

6.4 Experiments

6.4.1 Implementation Details

The proposed method is implemented using the GPU version of Tensorflow in v1.10.0. Two different types of Siamese-style architectures are used, they are VGGNet (Simonyan & Zisserman, 2014) and ResNet50 (K. He et al., 2016). All fully-connected layers are removed from the original backbone networks and then replaced by the projection layer and compressible covariance pooling layer at the same place to train the invariant classifier. The batch size is set to 32 during training. The SGD with a momentum of 0.9 and a weight decay of 0.0005 is used to optimise the gradient. The initial learning rate is set to 0.1 and becomes

0.01 when fine-tuning the entire network. The exponential decay is applied in the training process, with a decay factor of 0.9 in every 10 epochs. The five-fold cross-validation is used to reduce the influence of the randomness and obtain reliable results. When training the proposed model on UC Merced Land-Use dataset (Y. Yang & Newsam, 2010), NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017), and OPTIMAL-31 dataset (Q. Wang et al., 2018), it randomly crops patches of 224 $\times$ 224 pixels from the input image and flip them horizontally or vertically. During the test, the manipulation of central cropping is adopted to obtain patches of the same size as in training. These operations are also applied to AID (Xia, Hu, et al., 2017), but the size of patches becomes 448 $\times$ 448 pixels.

6.4.2 Comparison with State-of-the-Arts

Four variants of the IDCCP model are proposed, and their overall classification accuracy and standard deviation are presented in TABLE 6.3. It is plain to see that the proposed IDCCP models achieved extremely competitive results on all experimental datasets. In particular, the performance of the IDCCP model based on ResNet50 (K. He et al., 2016) is superior to the latest MG-CAP model in Chapter 4 (S. Wang, Guan, & Shao, 2020) on all datasets and even far exceeds baseline methods (e.g., the proposed method is improved by about 10% compared with the standard method of AlexNet + SVM on the challenging NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017)). When using VGGNet (Simonyan & Zisserman, 2014), the MG-CAP model in Chapter 4 (S. Wang, Guan, & Shao, 2020) shows strong competitiveness in classification accuracy, but even if GPU acceleration is enabled, it requires 4.5 times the number of transformations and nearly seven times in terms of inference time. When the ResNet50-based Siamese-style architecture is employed, the proposed IDCCP models can obtain accuracy rates higher than 91% and 93% under two split ratios on NWPU-RESISC45 dataset

Table 6.3: Comparison with state-of-the-art deep learning-based approaches in terms of overall accuracy and standard deviation (%). T.R. is the abbreviation of the Training Ratio.

Methods	NWPU-RESISC45		AID		UC-Merced Land-Use
	T.R.=10%	T.R.=20%	T.R.=20%	T.R.=50%	T.R.=80%
AlexNet+SVM (Cheng et al., 2018)	81.22±0.19	85.16±0.18	84.23±0.10	93.51±0.10	94.42±0.10
GoogLeNet+SVM (Cheng et al., 2018)	82.57±0.12	86.02±0.18	87.51±0.11	95.27±0.10	96.82±0.20
VGGNet+SVM (Cheng et al., 2018)	87.15±0.45	90.36±0.18	89.33±0.23	96.04±0.13	97.14±0.10
MSCP with AlexNet (N. He et al., 2018)	81.70±0.23	85.58±0.16	88.99±0.38	92.36±0.21	97.29±0.63
MSCP+MRA with AlexNet (N. He et al., 2018)	88.31±0.23	87.05±0.23	90.65±0.19	94.11±0.15	97.32±0.52
MSCP with VGGNet (N. He et al., 2018)	85.33±0.17	88.93±0.14	91.52±0.21	94.42±0.17	98.36±0.58
MSCP+MRA with VGGNet (N. He et al., 2018)	88.07±0.18	90.81±0.13	92.21±0.17	96.56±0.18	98.40±0.34
DCNN with AlexNet (Cheng et al., 2018)	85.56±0.20	87.24±0.12	85.62±0.10	94.47±0.10	96.67+0.10
DCNN with GoogLeNet (Cheng et al., 2018)	86.89±0.10	90.49±0.15	88.79±0.10	96.22±0.10	97.07+0.12
DCNN with VGGNet (Cheng et al., 2018)	89.22±0.50	91.89±0.22	90.82±0.16	96.89±0.10	98.93±0.10
RTN in Chapter 3 (Z. Chen et al., 2018)	89.53±0.21	92.20±0.34	92.75±0.21	95.09±0.16	98.33±0.71
Two-Stream Fusion (Y. Yu & Liu, 2018)	80.22±0.22	83.16±0.18	92.32±0.41	94.58±0.25	98.02±1.03
GCFs+LOFs (Zeng et al., 2018)	-	-	92.48±0.38	96.85±0.23	99.00±0.35
CapsNet with VGGNet (W. Zhang et al., 2019)	85.08±0.13	89.18 ± 0.14	91.63±0.19	94.74±0.17	98.81±0.22
MG-CAP (Bilinear) in Chapter 4 (S. Wang, Guan, & Shao, 2020)	89.42±0.19	91.72 ± 0.16	92.11±0.15	95.14±0.12	98.60±0.26
MG-CAP (Log-E) in Chapter 4 (S. Wang, Guan, & Shao, 2020)	88.35±0.23	90.94 ± 0.20	90.17±0.19	94.85±0.16	98.45±0.12
MG-CAP (Sqrt-E) in Chapter 4 (S. Wang, Guan, & Shao, 2020)	90.83±0.12	92.95 ± 0.13	93.34±0.18	96.12±0.12	99.00±0.10
CFE in Chapter 5 (S. Wang et al., -)	90.64±0.12	92.77 ± 0.13	93.15±0.18	95.78±0.15	**99.19**±0.42
the proposed IDCCP with VGGNet-512	90.88±0.18	92.80±0.10	93.58±0.24	96.33±0.12	98.45±0.12
the proposed IDCCP with VGGNet-64	89.61±0.19	91.75±0.18	92.33±0.25	94.82±0.22	97.35±0.20
the proposed IDCCP with ResNet50-512	**91.55**±0.16	**93.76**±0.12	**94.80**±0.18	**96.95**±0.13	99.05±0.20
the proposed IDCCP with ResNet50-64	91.31±0.22	93.66±0.21	94.64±0.23	96.73±0.18	98.57±0.24

Table 6.4: Comparison with state-of-the-art methods in terms of overall accuracy and standard deviation (%).

Method	OPTIMAL-31 Training Ratio=80%
Fine-tuned AlexNet (Q. Wang et al., 2018)	81.22 ± 0.19
Fine-tuned GoogLeNet (Q. Wang et al., 2018)	82.57 ± 0.12
Fine-tuned VGGNet16 (Q. Wang et al., 2018)	87.45 ± 0.45
ARCNet with Alexnet (Q. Wang et al., 2018)	85.75 ± 0.35
ARCNet with ResNet34 (Q. Wang et al., 2018)	91.28 ± 0.45
ARCNet with VGGNet16 (Q. Wang et al., 2018)	92.70 ± 0.35
the proposed IDCCP with VGG-512	93.82±0.32
the proposed IDCCP with VGG-64	92.13±0.38
the proposed IDCCP with ResNet50-512	**94.89**±0.22
the proposed IDCCP with ResNet50-64	94.54±0.28

(Cheng, Han, & Lu, 2017), respectively. On AID (Xia, Hu, et al., 2017), it can obtain 94.80±0.18 with using 20% of training samples, which exceeds the best results of MG-CAP model in Chapter 4 (S. Wang, Guan, & Shao, 2020) and DCNN model (Cheng et al., 2018) by 1.46% and 3.98%, respectively. Under the 50% training ratio, the GCFs+LOFs model (Zeng et al., 2018) presents surprisingly better than most existing methods but still below the optimal level of the proposed IDCCP model. On UC Merced Land-Use dataset (Y. Yang & Newsam, 2010), the highest accuracy among the listed algorithms is achieved by the CFE model in Chapter 5 (S. Wang et al., -), which relies on imposing the appreciate measurements on the high-dimensional vectorised covariance features and the corresponding weights. In addition, it shows the comparisons of the proposed IDCCP model with previous methods on the OPTIMAL-31 dataset (Q. Wang et al., 2018). As shown in TABLE 6.4, three variants of the proposed method can achieve higher results than the state-of-the-art ARCNet model (Q. Wang et al., 2018). Even the worst of the IDCCP model can still exceed the result of fine-tuned AlexNet by more than 10%. By using ResNet50 architecture, the proposed IDCCP model can

improve the optimal performance of ARCNet with VGGNet-16 by 1.84%. These indicate that the classification performance can be improved by incorporating the prior knowledge of the input image.

Generalisation ability is vitally important for measuring the effectiveness of deep learning models. By analysing the data listed in Table 6.3, it is not difficult to see that the variants of the proposed IDCCP model can always bring relatively stable benefits to different datasets. Concretely, using different proportions of training data on the NWPU-RESISC45 (Cheng, Han, & Lu, 2017) (i.e., 10% versus 20% training ratios), the difference between the proposed IDCCP model is about 2%, but this gap is significantly enlarged on other models (e.g., about 4% by CapsNet with VGGNet-16 (W. Zhang et al., 2019) and about 3% by MSCP with AlexNet or VGGNet-16 (N. He et al., 2018)). A similar degree of gain is also reflected in the AID (Xia, Hu, et al., 2017) with different partitions. However, most of the existing methods are not stable enough under different partitions, including DCNN (Cheng et al., 2018) (about 6%-9%), GCFs+LOFs (Zeng et al., 2018) (about 4%), and SVM-based methods (Cheng et al., 2018) (about 7%-9%). It is worth noting that the actual number of samples corresponding to different training ratios on two different datasets (10% and 20% on NWPU-RESISC45 (Cheng, Han, & Lu, 2017) versus 20% and 50% on AID (Xia, Hu, et al., 2017)) is in the same order of magnitude (3,150 on NWPU-RESISC45 (Cheng, Han, & Lu, 2017) versus 3,000 on AID (Xia, Hu, et al., 2017)). Therefore, similar gains in different scenarios also reflect that the robustness of the proposed IDCCP model.

Through comparing the variants of our IDCCP model, it can be found that the IDCCP model based on VGGNet (Simonyan & Zisserman, 2014) can achieved very competitive results on all experimental datasets. Especially, using VGGNet (Simonyan & Zisserman, 2014), the proposed IDCCP model can obtain comparable results to the similar methods, such as RTN (in Chapter 3) (Z. Chen et al.,

2018) and MG-CAP model in Chapter 4 (S. Wang, Guan, & Shao, 2020), and even significantly better than MSCP (N. He et al., 2018). The full-rank IDCCP model (i.e., the VGGNet-512) obtained a classification accuracy rate of about 10% higher than the two-stream fusion model (Y. Yu & Liu, 2018) on the NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017). Furthermore, at the expense of the accuracy of the tolerable range (i.e., about 1%-2%), it allows compressing the model parameters to 1/64 of the original second-order features. The performance gap between IDCCP models based on full-rank and low-rank is rarely small, and some of them are even only 0.1%. For example, using ResNet50 to train the proposed model on the NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017) (under the 20% training ratio) can achieve 93.76% and 93.66% accuracy. Apart from the advanced structure of ResNet50, the proposed IDCCP model also benefits from the orthogonal feature reduction layer and the projection layer (i.e., 1×1 convolution layer), which can effectively remove the redundant feature information.

In addition to comparing overall accuracy, it also presents examples of confusion matrices to show category-level details. For the demonstration, an experiment result is randomly selected from each experiment scenario and shown in Figure 6.3 to Figure 6.7. It can be clearly seen that the darkest colour blocks appear on the diagonals of all confusion matrices. On NWPU-RESISC45 dataset 6.3 (under the 10% training ratio), there exists 35 categories among all 45 categories obtain a classification accuracy rate higher than 90%. Compared with the experimental results obtained by the Fine-tuned VGGNet-16 in (Cheng, Han, & Lu, 2017), the proposed model brings 9% and 16% improvements in the two most confusing categories (i.e., the **Church** category and the **Palace** category), respectively. By using 20% of training samples (shown in Figure 6.4), the proposed IDCCP model with ResNet50-512 can achieve 82% and 77% accuracy on the confusing **Church** and **Palace** categories. Substantial improvements have been

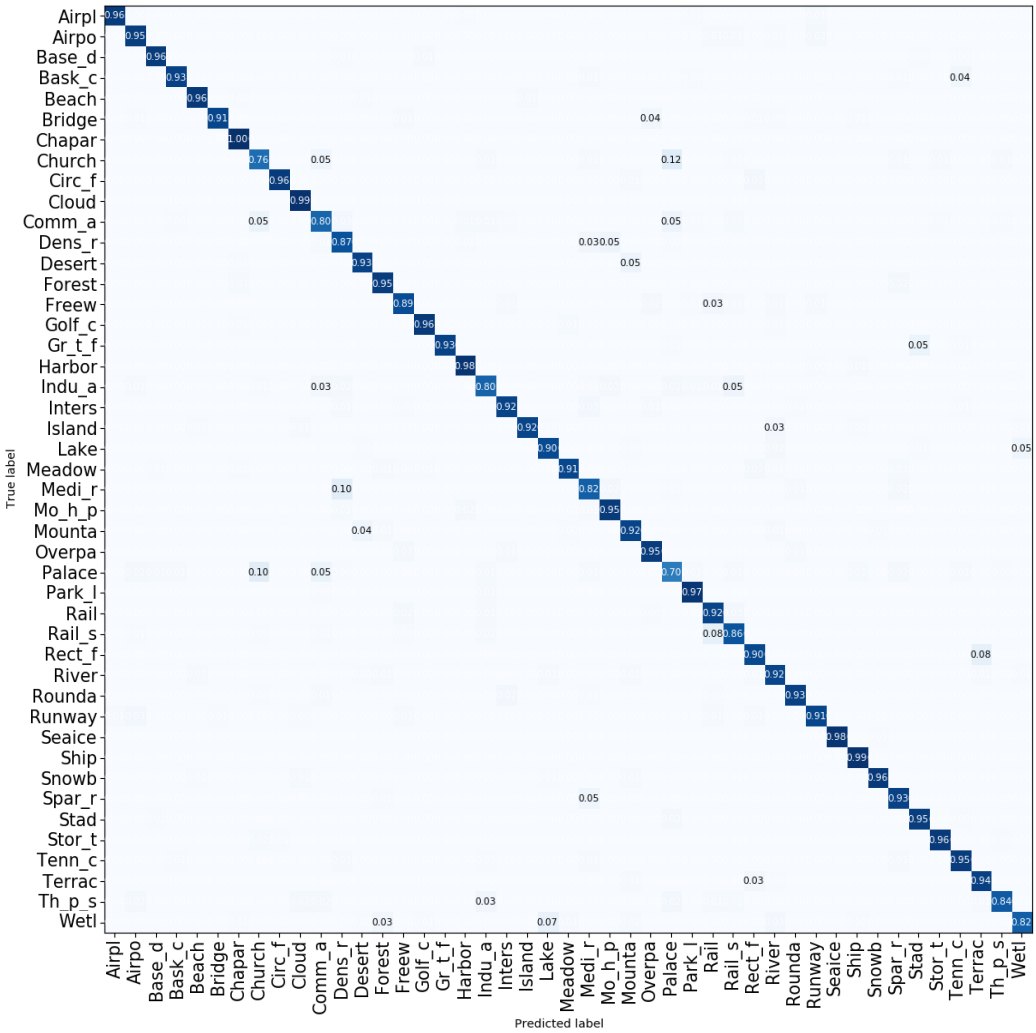


Figure 6.3: The confusion matrix on NWPU-RESISC45 dataset under the training ratio of 10%. For the sake of clarity, values less than 0.03% are omitted.

made compared with the benchmark methods described in (Cheng, Han, & Lu, 2017) (i.e., with improvements of 24% and 25%, 7% and 11% compared with the transferred VGGNet-16 (Simonyan & Zisserman, 2014) and the Fine-tuned VGGNet (Simonyan & Zisserman, 2014), respectively.) Even compared with the latest methods like the D-CNN (Cheng et al., 2018) and the RTN (in Chapter 3) (Z. Chen et al., 2018), the classification results of these two categories are improved by 3% and 4%, and 2% and 1%, respectively. On AID dataset 6.5, the

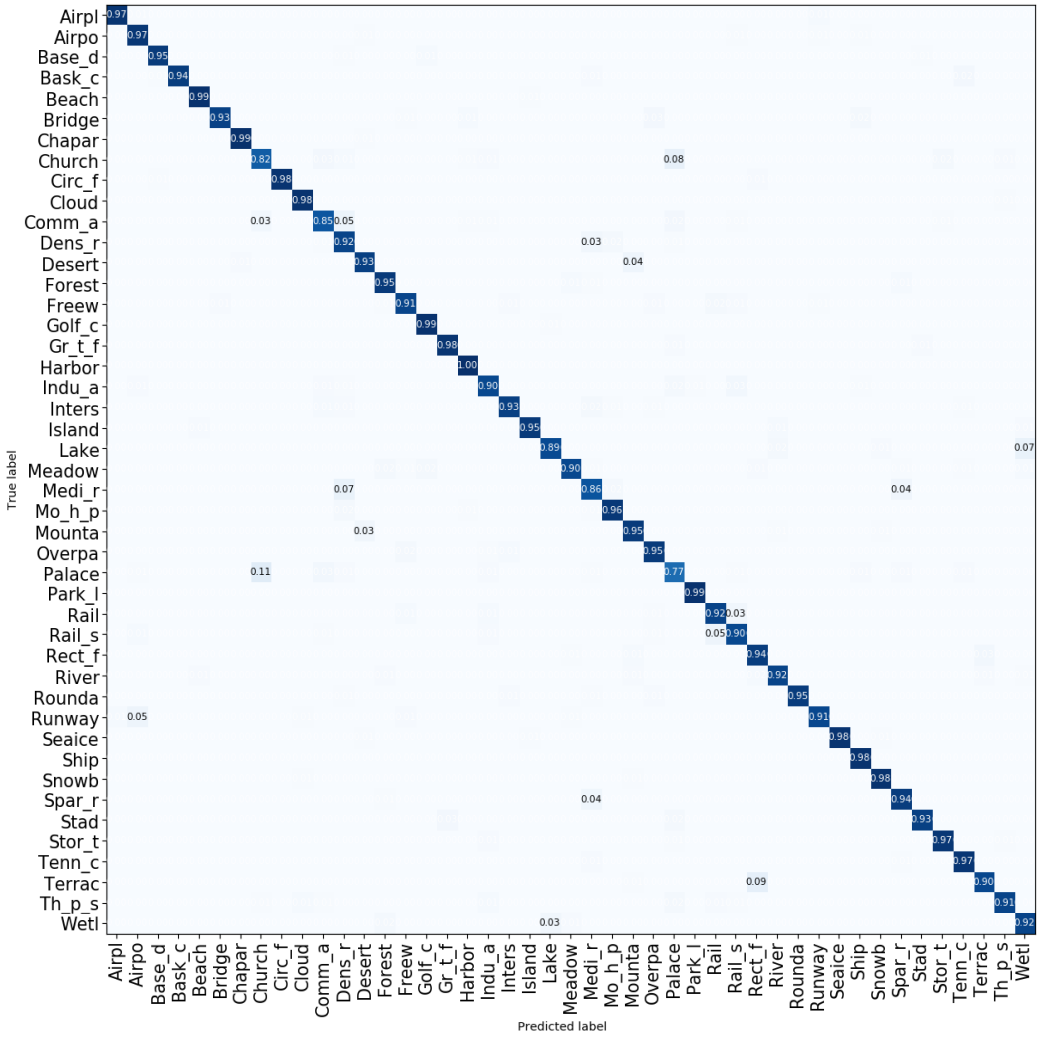


Figure 6.4: The confusion matrix on NWPU-RESISC45 dataset under the training ratio of 20%. For the sake of clarity, values less than 0.03% are omitted.

reported accuracy of **School** is 85%, which exceeds the algorithm introduced in (Xia, Hu, et al., 2017) by 18%. Due to the high similarity, 16% of **Resort** images are misclassified as **Park**. The classification results of the easily confused **Dense residential**, **Medium residential** and **Sparse residential** are 96%, 93% and 99%, respectively. When using 50% of the training samples, only the two categories of **Resort** and **Square** have an accuracy below 90% (totally 30 categories). The sparsest confusion matrix 6.7 is obtained by evaluating the proposed framework

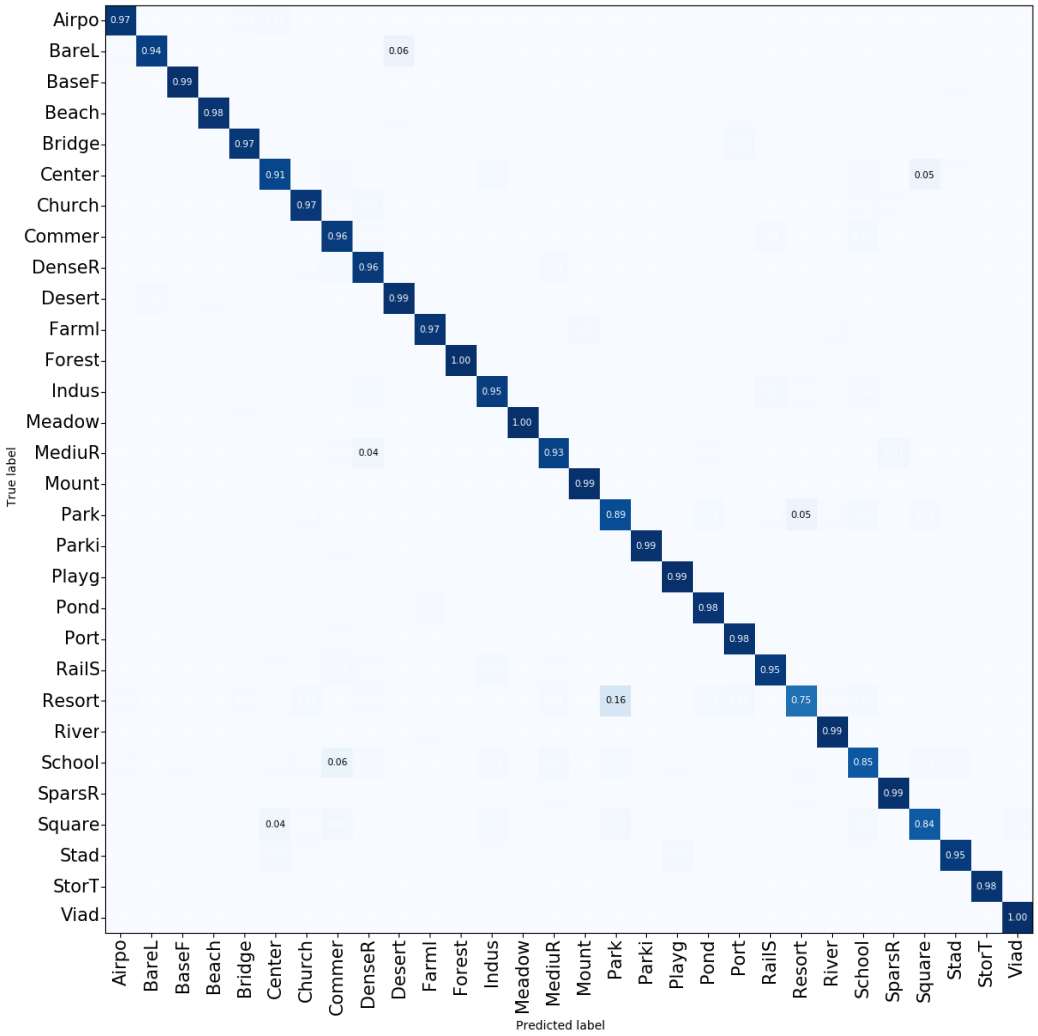


Figure 6.5: The confusion matrix on AID dataset under the training ratio of 20%. For the sake of clarity, values less than 0.03% are omitted.

on UC-Merced Land-Use dataset (Y. Yang & Newsam, 2010) with the training ratio of 80%.Due to the relatively small size of the dataset, the accuracy of all categories has reached more than 90%. On OPTIMAL-31 dataset (Q. Wang et al., 2018), there are 20 categories of test data that can be classified 100% correctly 6.8. Especially, for those categories that are difficult to distinguish, including **Church**, **Industrial area** and **Island**, the proposed IDCCP model can bring significant improvements of 21%, 25% and 25%, respectively.

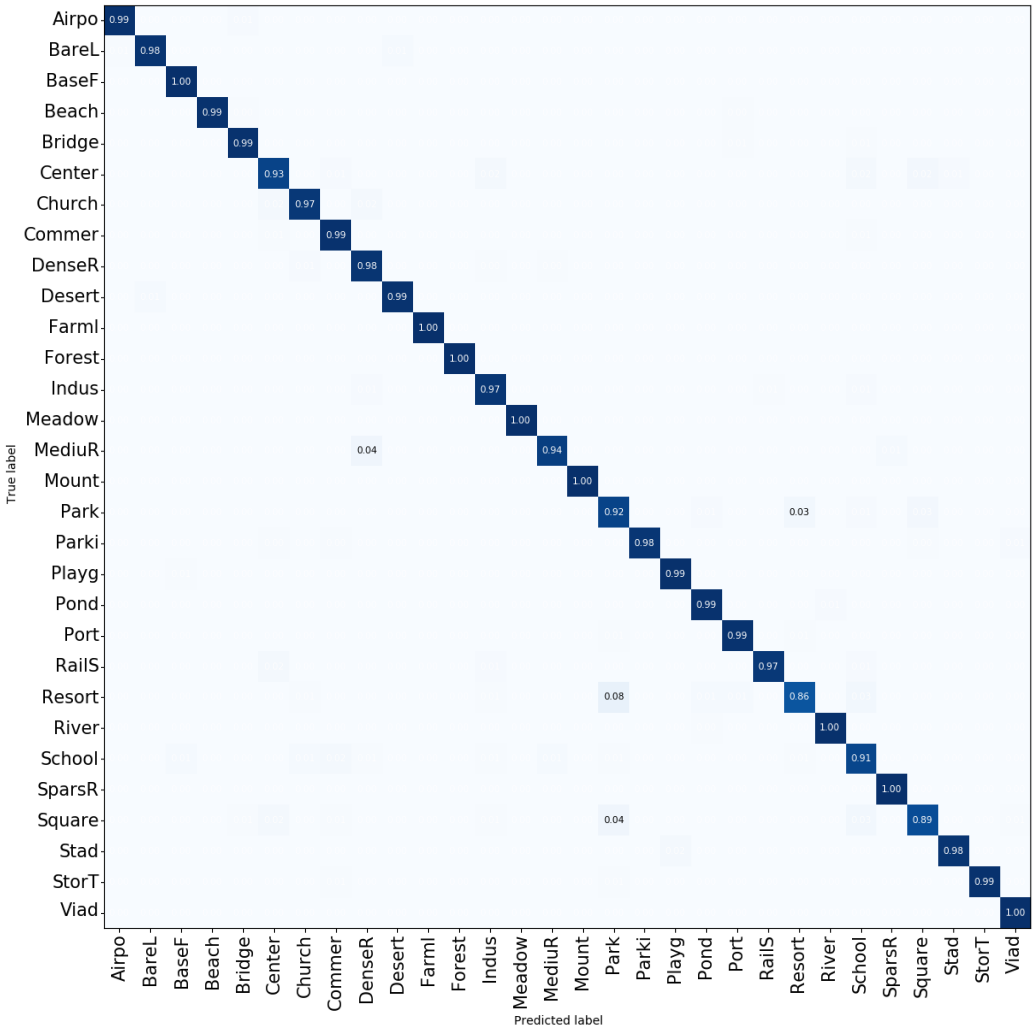


Figure 6.6: The confusion matrix on AID dataset under the training ratio of 50%. For the sake of clarity, values less than 0.03% are omitted.

6.4.3 Analysis of Model Complexity

In view of the success of bilinear pooling (Lin et al., 2015) and its relevance to the proposed method, it will compare the differences of two models in various aspects and list the results in TABLE 6.5. Especially, the aspects of comparison include input feature dimension, complexity and corresponding parameter size, classifier complexity and its parameter size, and overall model parameters. In

Table 6.5: Comparison with the Bilinear pooling method (Lin et al., 2015) in terms of feature dimensionality, computational complexity and the number of parameters (ResNet50 (K. He et al., 2016)-based Siamese-style architecture). Where $d_p = 512$ and K denote the projection layer and the number of categories, respectively. (w/ and w/o indicate with projection layer and without projection layer, respectively.)

Methods	Feature Dim.	Feature Comp.	Classifier Comp.	Feature Param.	Classifier Param.	Model Param.
Bilinear pooling (Lin et al., 2015) (w/o d_p)	d^2 [4,194K]	$O(hwd^2)$	$O(Kd^2)$	0	Kd^2 [K·16MB]	[118MB]
Our IDCCP-512 (w/ d_p)	$\hat{d}_1^2$ [256K]	$O(hwd_pd + hw\hat{d}_1^2)$	$O(Kd_1^2)$	dd_p [4MB]	Kd_1^2 [K·1MB]	[30MB]
Our IDCCP-64 (w/ d_p)	$\hat{d}_2^2$ [4K]	$O(hwd_pd + hw\hat{d}_2^2)$	$O(Kd_2^2)$	dd_p [4MB]	Kd_2^2 [K·16KB]	[25MB]

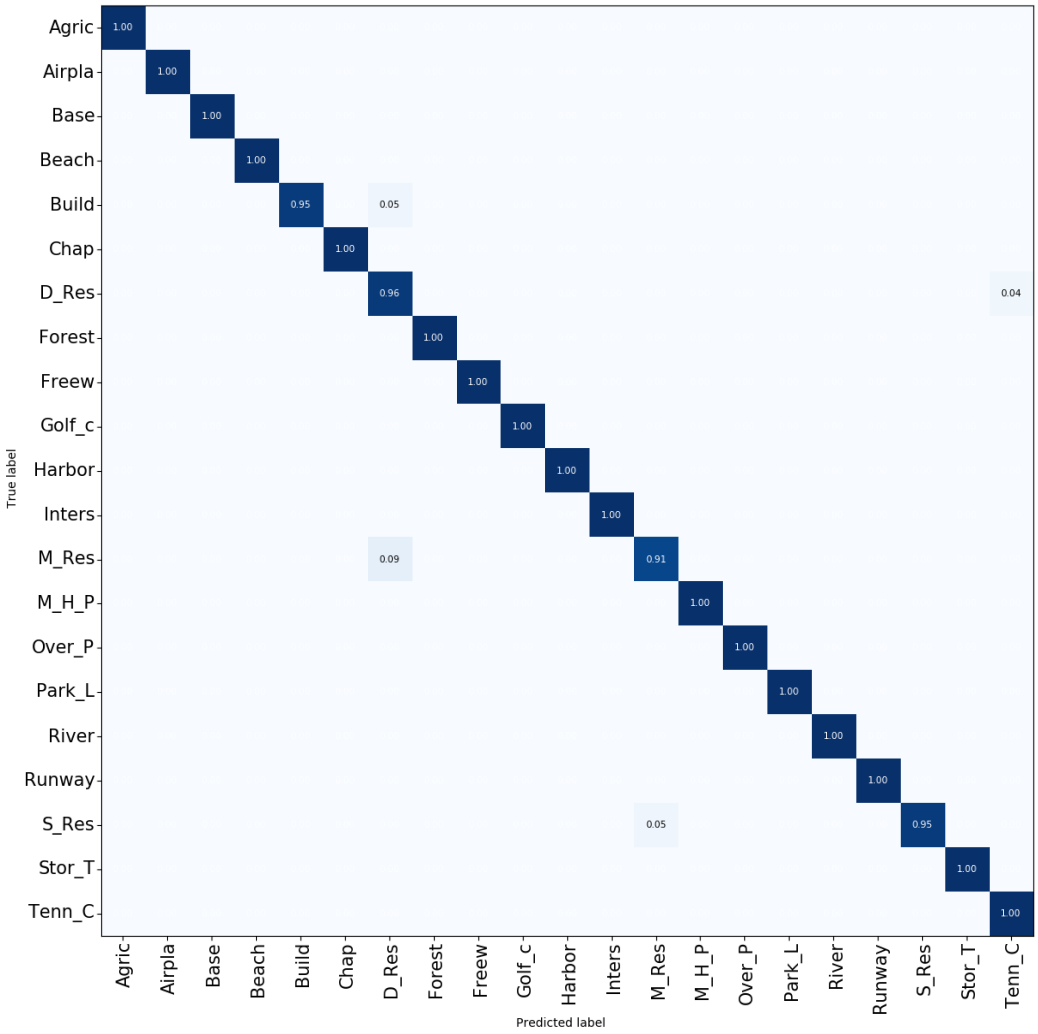


Figure 6.7: The confusion matrix on UC-Merced dataset under the training ratio of 80%. For the sake of clarity, values less than 0.03% are omitted.

order to show the function of the projection layer, all results are obtained by employing the Siamese-style architecture based on ResNet50 (K. He et al., 2016). With using the projection layer, the feature dimension can be reduced to the same par with the last convolution layer in VGGNet (Simonyan & Zisserman, 2014). As shown in TABLE 6.5, the invariant deep compressible covariance pooling (ID-CCP) model requires an additional 4-MB feature parameter compared to the bilinear pooling model (Lin et al., 2015). However, this operation is more conducive

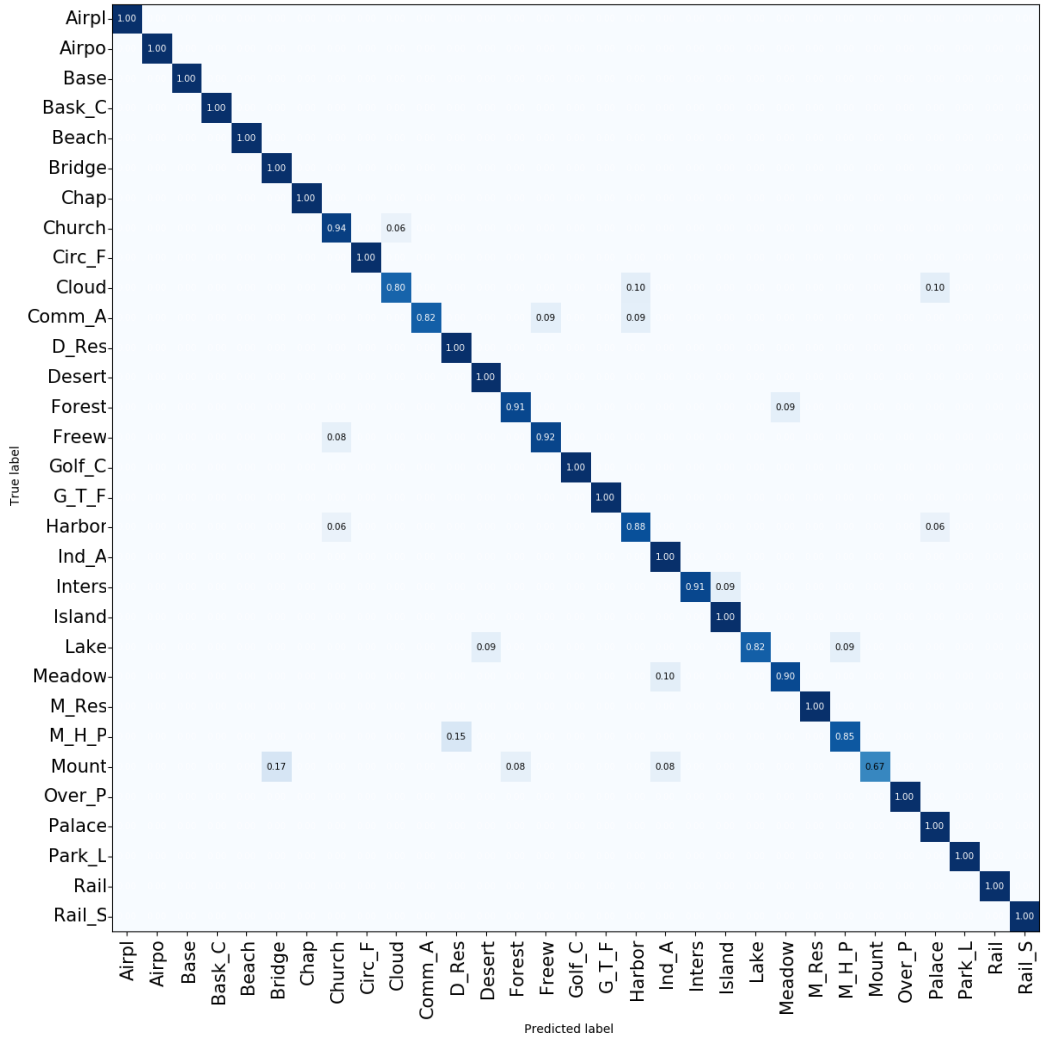


Figure 6.8: The confusion matrix on Optimal-31 dataset under the training ratio of 80%. For the sake of clarity, values less than 0.03% are omitted.

to reducing the feature dimension and, thus, greatly reducing the number of classifier parameters. Namely, the IDCCP model not only learns compressible feature representations but also trains more compact classifiers.

Table 6.6: Comparison of classification accuracy and single image inference time. Experiments were conducted on NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017) with using 10% training samples.

Networks	Feature Dim.	Accuracy (%)		Time (sec/per image)	
		w/ D4	w/o D4	w/ D4	w/o D4
ResNet50 (K. He et al., 2016)	2048	-	90.02	-	0.0219
	512	91.64	90.05	0.0768	0.0105
	64	91.26	89.94	0.0744	0.0093
	16	90.78	89.83	0.0721	0.0087
VGGNet (Simonyan & Zisserman, 2014)	512	91.11	89.44	0.0324	0.0063
	64	89.78	88.34	0.0322	0.0059
	16	88.62	87.21	0.0317	0.0052

6.4.4 Ablation Study and Analysis

6.4.4.1 Compactness and Effectiveness

In Table 6.6, extensive results are listed to show the effect of feature dimensionality and D4 transformation group on classification accuracy and a single image inference time. For a fair comparison, it ensures that all hyperparameters are consistent and then obtain the interaction time of a single image by calculating the ratio of the total test duration to the number of test samples. When the feature size is reduced, the gap in classification accuracy will not be significantly enlarged. For example, with ResNet50 architecture (K. He et al., 2016), the accuracy only decreases by 0.86% even if it compresses the feature space to 1/64 of the original feature space. It is worth noting that the IDCCP model allows features to be compressed into a very compact space (i.e., 16×16) without sacrificing too much accuracy. Interestingly, the classification accuracy is slightly improved when 1 $\times$ 1 convolution layer is used to map the CNN feature to a lower feature space. The reason for this phenomenon is that 1 $\times$ 1 convolution can reduce the diversity and redundancy of feature maps, thereby improving the discriminative power of

Table 6.7: A Comparison of using $P_{trivial}$ and P_{maxout} on NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017) with 10% training samples.

Proj.	VGGNet-512	VGGNet-64	ResNet50-512	ResNet50-64
$P_{trivial}$	**90.88±0.18**	**89.61±0.19**	91.55±0.16	**91.31±0.22**
P_{maxout}	90.00±0.17	89.10±0.10	**91.79±0.13**	90.71±0.19

learned feature (Wei et al., 2018). Due to the limited capability of the PC, the accuracy of equipping the D4 transformation group has been omitted. However, this hardly affects the effectiveness of investigating the D4 transformation group. At the feature size of 16×16, the IDCCP model based on ResNet50 (K. He et al., 2016) achieved 89.9% accuracy, which can exceed the full-rank constrained VGGNet model. It not only influenced by the superior structure of ResNet50 (K. He et al., 2016) but also reflects the effectiveness of the projection layer. In addition, ResNet50 (K. He et al., 2016)-based IDCCP model, a single image inference time, only needs about 0.07 and 0.01 seconds for equipping or not equipping the D4 group, respectively. Due to the relatively shallow CNN structure, the inference time reduce by half when using VGGNet (Simonyan & Zisserman, 2014) architecture.

To evaluate the efficiency of using $P_{trivial}$ and P_{maxout}, experiments are conducted on NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017) using 10% training data and show the corresponding results in Table 6.7. It is obvious that learning trivial representation usually performs better compared with learning maxout feature. Interestingly, in the case of using ResNet50-51, the maxout operation produces the accuracy of 91.79%, which exceeds the trivial operation by 0.24%. The reason for this phenomenon is that the maxout operation always learns the maximum response from the transformation group. Furthermore, a prerequisite of using maxout operation is that it requires the feature has been well-compressed (e.g., using 1×1 convolution kernel to remove redundant information). Through comparing

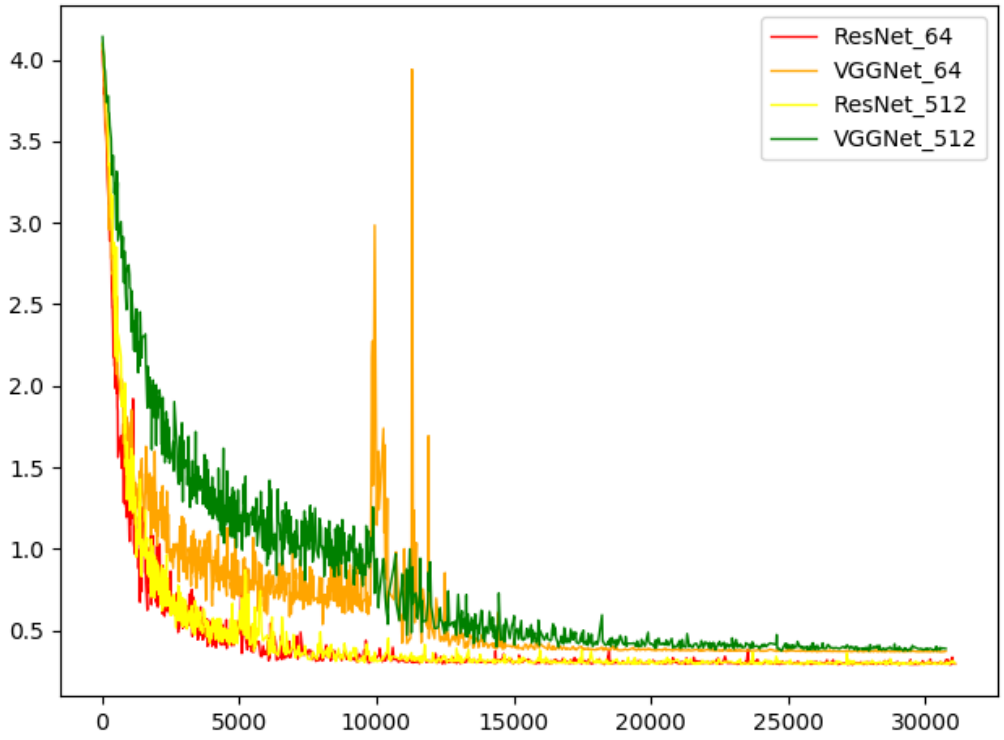


Figure 6.9: Comparison of loss convergence.

the results between ResNet50-512 and ResNet50-64, it is easy to find that the accuracy decreases over 1% by using P_{maxout} while it only change slightly using $P_{trivial}$ (*i.e.,* 0.24%). It reveals that $P_{trivial}$ performs more robust than P_{maxout}.

6.4.4.2 Convergence Speed

The convergence speed of loss is a key indicator for evaluating the effectiveness of deep learning models. In Figure 6.9, it displays the curve of loss change in four different scenarios (i.e., on both ResNet50 (K. He et al., 2016) and VGGNet (Simonyan & Zisserman, 2014) based architectures). Compared with VGG (Simonyan & Zisserman, 2014) based architecture, the ResNet50 (K. He et al., 2016) based architecture converges faster. Except for the advanced structure of the residual unit in ResNet50 (K. He et al., 2016), it also indirectly reveals the

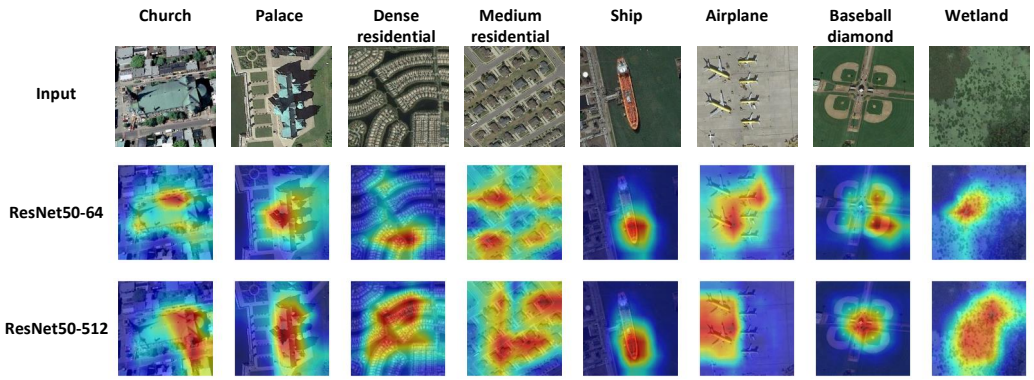


Figure 6.10: Selected images for qualitative visualisation.

superiority of using the projection layer before computing covariance matrix. The attractive point is that the loss of using 64 $\times$ 64-dimensional feature appears more smooth than using 512 $\times$ 512-dimensional features. The difference expands evidently on VGGNet (Simonyan & Zisserman, 2014) when the projection layer between the CNN feature and covariance matrix has been removed. In addition, it found that VGGNet (Simonyan & Zisserman, 2014) is more difficult to training in practice, even though it trained the classification layer in more epochs than ResNet50 (K. He et al., 2016). This may be caused by the redundant features in the deeper CNN structure. It suggests that it is necessary to propose the projection layer when using ResNet50 (K. He et al., 2016).

6.4.4.3 Qualitative Visualisation & Failure Cases

Through the comparison of the above experiments, it shows that the overall accuracy of the compressed model can be kept at the same par with the uncompressed model. Then, it promises to seek evidence from the interpretability of the model. As shown in Figure 6.10, example images are selected from NWPU-RESISC45 dataset (Cheng, Han, & Lu, 2017) and the corresponding heatmaps are shown by using the Grad-Cam algorithm (Selvaraju et al., 2017). When using ResNet50-64

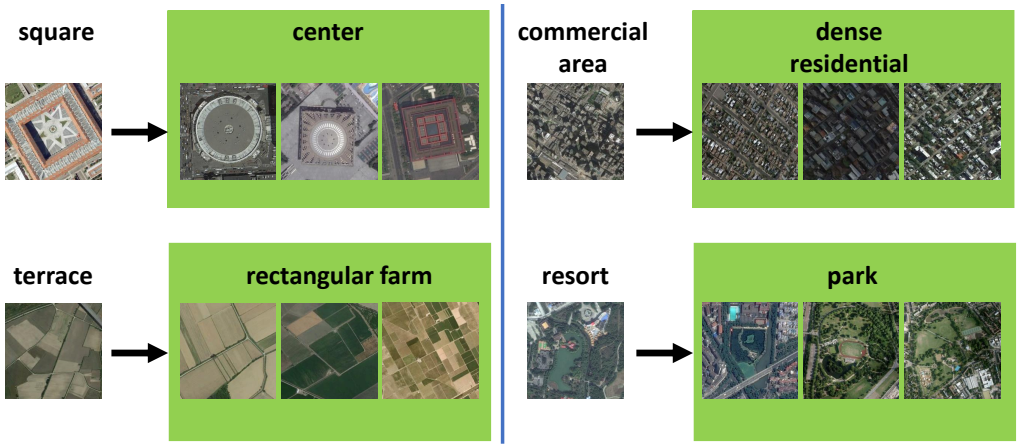


Figure 6.11: The cases of misclassification. The images to the left of the arrow were misclassified into categories to the right of the arrow. Images in green blocks are actual images.

architecture, it presents that the proposed model can focus on small patches that benefit to distinguish subtle differences between visually similar images, such as **Church** and **Palace**, **Dense residential** and **Medium residential**. Compared with ResNet50-64, the area of attention map is significantly expanded when using ResNet50-512 model. Namely, it allows the model to capture more texture information and could be the reason why ResNet50-512 performs slightly better than ResNet50-64 model.

Demonstrating failure cases helps to better understand the proposed model. Four misclassified images are chosen and displayed in Figure 6.11. It can be seen from Figure 6.11 that the proposed algorithm may misclassify images that show subtle texture differences or contain similar distinguished objects.

6.5 Conclusion

In this chapter, it has proposed a unified IDCCP model that can be regarded as a paramedian to handle the visual-semantic discrepancy and nuisance variations

in the classification of aerial scene images. The model benefits from the use of Siamese CNNs to learn the trivial representation of the predefined transformation group. The obtained representation can be deployed to the scenarios of the second-order representation. Meanwhile, it attempted to endow the weight matrix with the form of Stiefel manifold and employed it to reduce the dimensions of the SPD manifold. Finally, the generated features are flattened to train the invariant classifier in a compact space and obtain state-of-the-art results.

7 | Conclusion and Future Work

7.1 Discussion and Conclusion

The ultimate goal of my thesis is to learn transformation-invariant deep tensor features to improve the performance of RSSC tasks. For this reason, I explored the methods of learning robust second-order features and employed different strategies to expand the feature capabilities of prior knowledge, thus realising that the learned features are invariant to the test data. I brought these works to fruition in four papers and named them as RTN in Chapter 3 (Z. Chen et al., 2018), MG-CAP in Chapter 4 (S. Wang, Guan, & Shao, 2020), CFE in Chapter 5 (S. Wang et al., -) and IDCCP in Chapter 6 (S. Wang, Ren, et al., 2020), respectively. The listed models are optimised for classification accuracy or model complexity in the order in which the chapters appear, and finally, a paradigm with favourable theoretical support is proposed for the RSSC task. For clarity, the achievements of these four models are summarised in the following subsections.

7.1.1 Transformation-invariant Feature Representation

The key to image classification is whether the learned model is capable of perceiving the offset caused by the image transformation between the training domain and the test domain. The invariance of features, in this case, has always attracted the attention of researchers. The early exploration of invariance features can be traced back to the era when feature designers explicitly coded the inherent invariance characteristics of a given image (e.g., SIFT feature (Lowe, 2004) and SURF feature (Bay et al., 2006)). These hard-coded methods can capture the invariant information of the image, but can only take captive a limited number of explicit features, depending on the prior knowledge of the feature engineer.

Fusion of features with different characteristics may improve the performance, but it is extremely complicated to determine hyperparameters that can reasonably weight different features.

The emergence of CNN allows the model to learn features that are accommodating for classification from the pixel-level, which greatly relieves researchers from the burden of manual feature design. Owing to the unique recursive hierarchical structure, CNNs can learn a certain degree of invariance by incorporating pooling functions, but the limited receptive domain makes the learned invariance only reflected locally. This is why merely flipping the same image may completely invalidate the powerful CNN model. In order to compensate for the lack of global transformation of the model, parameterised affine transformation (RTN (Z. Chen et al., 2018) in Chapter 3), predefined hierarchical transformation (MG-CAP (S. Wang, Guan, & Shao, 2020) in Chapter 4) and symmetric group transformation (IDCCP model (S. Wang, Ren, et al., 2020) in Chapter 6) are proposed to increase the diversity of image level transformation. The learned transformation can be retained in a given standard CNN structure by leveraging a particular network structure, namely, Siamese-style network. Followed by the pixel-wise maximum operation, the most favorable features for the classification function can be obtained. In addition, from the spatial transformer-based attention mechanism to the granular transformation, and then to the asymmetric D4 transformation group, our methods can learn the invariant deep learning features in a more effective and efficient way.

7.1.2 Second-order Statistical Pooling

Pooling operation is an essential component of CNN and it is often occurs after one or more stacked convolutions to decrease the size of feature maps and then reduce the number of model parameters. Common pooling methods like average pooling and maximum pooling can learn a certain degree of invariance for the

model, such as translation and zooming, along with the continuous shrinking of the convolution maps. In addition to the local pooling method, the global pooling methods (i.e., GAP layer or GMP layer) can also selectively substituted for the fully connected layer in the image classification task because there is no need to optimise additional parameters and degrade the risk of overfitting. However, either the local pooling method or the global pooling method only summarises the spatial information of an individual channel, so there are limitations in statistical modeling and model generalisation capabilities.

Compared with the above-mentioned first-order pooling, the second-order pooling methods establish firm correlations between feature spatial locations and channels through the matrix product (i.e., outer product or Kronecker product). The outer product of the matrix undoubtedly enlarges the magnitude of the eigenvalues, thus it is prone to visual burstiness problems. Namely, the second-order statistics are more sensitive to changes in the magnitude local CNN feature elements. In order to alleviate the impact of this problem, the fast and stable solution of the square root of the high-dimensional matrix has become imperative. In particular, schemes based on eigenvalue decomposition and equipped with different matrix norms are proposed, which allows the use of GPU acceleration. In addition, considering the unique geometric structure of the covariance matrix, an orthogonal constraint based on the Stiefel manifold is introduced to effectively reduce the dimensionality of the high-dimensional Riemannian manifold (IDCCP model (S. Wang, Ren, et al., 2020) in Chapter 6).

7.1.3 Low-norm Cosine Similarity Loss

Vectorisation is a inevitable operation before convolutional features are passed to the fully connected layers or the global pooling layers. Vectorisation gives the features a certain degree of spatial invariance, but it also produces a high-dimensional

space. Taking into account the commonly utilised distance measurements based on L2 normalisation in high-dimensional space will result in the ratio of the closest distance to the farthest distance to a given target point is very approaching to 1, a more effective measurement method is desired to accurately measure the spatial distance between different points because vectorised second-order features are accompanied by an exponential increase in feature dimensions. Therefore, a vector-based low-norm measurement function is proposed to estimate the angle between the vectorised covariance matrix and the corresponding weight in the angle space, and then a novel low-norm cosine similarity loss function is obtained (CFE model (S. Wang et al., -) in Chapter 5).

7.2 Future Work

A total of four models have been proposed to solve the challenges of the RSSC task, and they have shown substantial improvements compared to previous baseline methods. For example, on one of the experimental datasets-UC-Merced Land-Use dataset (Y. Yang & Newsam, 2010), the three proposed models can achieve 99% classification accuracy or even beyond. However, from a long-term perspective, how to reduce the dependence on manually annotated large-scale data and how to effectively integrate multi-modal data to serve applications related to geospatial systems (such as the prediction and assessment of natural disasters).

7.2.1 Multi-modality Remote Sensing Data Fusion

Remote sensing data indicates the physical characteristics of a certain area on the Earth. Placing high-definition cameras or remote sensors at a certain distance above the earth, remote sensing data then can be gathered by taking images or measuring the reflected or emitted radiation. The data obtained can be roughly

classified into two categories: optical data and non-optical data. Optical remote sensing images usually contain valuable spatial information, which is one of the main reasons for the in-depth investigation of optical images categorisation in this thesis. However, these images are usually at the mercy of weather conditions, dark clouds and night.

As an increasing number of small satellites and UAVs plan to carry radar and hyperspectral image sensors (e.g., SARs and radiometers), a huge amount of non-optical remote sensing images with the same quality images are produced regardless of day and night, and different weather conditions. Furthermore, remote sensing sensors like SAR are highly sensitive to the roughness, wetness and movements of objects. Understanding and utilising these characteristics affords us with the opportunity to gain insights into certain practical applications, such as the crop cover type, the marine pollution sources and the post-earthquake assessments. Therefore, how to effectively employ multi-modal remote sensing image data such as the crowd-sourced geographic data to enhance human's understanding of the earth will become an important direction of my future research.

7.2.2 Weakly-supervised and Unsupervised Learning

Supervised learning algorithms have always dominated machine learning, and have become increasingly attractive in recent years with the rise of deep learning. In addition to bringing incomparable performance that traditional machine learning algorithms and even human, the criticism of deep learning is also widely known, namely, the need of well-annotated large-scale datasets. Because the acquisition of the manually tagged data is subjective and time-consuming, and it becomes extremely expensive when domain expertise is involved.

Based on the above reasons, I will try to explore how to effectively extend the

current idea of learning transformation invariant second-order features to weakly supervised or unsupervised learning scenarios. Weakly supervised learning, also called bootstrapping or self-training, trains the classifier from a few training samples and then utilises thought-to-be positive samples that yielded by the classifier for retraining. Compared with supervised learning, weakly supervised learning in this way can dramatically decrease the demand for labels. In addition to weakly supervised learning methods, unsupervised learning methods are also considered in future work. In an unsupervised learning scenario, it is assumed that unlabelled images will be applied to train the model so that the learned function can capture the inherent structure from the raw data.

7.2.3 Generative Model

As mentioned in the thesis, a tremendous amount of remote sensing data is out there and can now be accessed. In the field of remote sensing, most of the existing work is proposed through the use of discriminative methods, which tend to learn from human predetermined goals while ignoring the capture of intrinsc structure in the data. Therefore, the treasure trove of remote sensing data is required to be analysed and understood by developing powerful models.

The generative model is an effective way to learn the distribution of input data using unsupervised learning. In the past few years, it has achieved remarkable success because of its potential to intelligently understand the data space. In my opinion, the most prominent advantage of the generative model is that it can mimic the data distribution of the input data. Namely, compared to learning input data, learning data distribution is more practical. This is because the number of parameters is enormously less than the amount of data we use to train the model. The latest developments in generative models, especially generative adversarial networks (GAN), have been reported to be capable of generating high-resolution

natural images (Brock, Donahue, & Simonyan, 2018). This also motivates me to attempt to train advanced GAN models for high-resolution remote sensing image synthesis in the future.

7.2.4 Zero-shot Learning

With the continuous update of different types of remote sensing datasets, it is not difficult to employ deep learning to train task-specific models. However, at this stage, it is unrealistic to collect a remote sensing dataset containing all land-use and land-cover types with accurate labels. This inevitably leads to the emergence of new problem, namely, what if the query image comes from a class that has not been seen in the training? Is it possible to train a model that can depict and classify unseen images based on what has been learned like humans?

It is worth noting that this problem has been discussed in the general field of computer vision, and a novel task called transfer learning has been derived, which is a classification task based on image attributes. In the transfer learning family, zero-shot learning (Lampert, Nickisch, & Harmeling, 2009) is viewed as the most challenging task and has attracted the attention of many researchers. Give an ordinary machine learning model an image from an unseen category, it will return an outrageously wrong result to a large extent because no clear correlation mapping is established during the training process. However, zero-shot learning can give reasonable results based on the transfer of attribute knowledge used to describe image content. These attributes can be annotated manually, or they can be learned straightly from the training set of the seen images. Although it is a risky task to extend zero-shot learning to remote sensing schemes due to large visual-semantic discrepancies and nuisance variations in RS images, it can better imitate human learning progress and move towards an intelligent visual classification system.

Bibliography

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... others (2016). Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)* (pp. 265–283). 115, 204

Absil, P.-A., Mahony, R., & Sepulchre, R. (2009). *Optimization algorithms on matrix manifolds*. Princeton University Press. 152, 153

Acharya, D., Huang, Z., Pani Paudel, D., & Van Gool, L. (2018). Covariance pooling for facial expression recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition workshops* (pp. 367–374). 39, 67, 71, 73, 74, 81, 103, 105, 108, 145

Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory* (pp. 420–434). 108, 110

Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis & Machine Intelligence*(12), 2037–2041. 3, 24

Anwer, R. M., Khan, F. S., van de Weijer, J., Molinier, M., & Laaksonen, J. (2018). Binary patterns encoded convolutional neural networks for texture recognition and remote sensing scene classification. *ISPRS journal of photogrammetry and remote sensing*, *138*, 74–85. 24, 121, 124

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*. 31, 32

Bahmanyar, R., Cui, S., & Datcu, M. (2015). A comparative study of bag-of-words and bag-of-topics models of eo image patches. *IEEE Geoscience and Remote Sensing Letters*, *12*(6), 1357–1361. 28

Bashmal, L., Bazi, Y., AlHichri, H., AlRahhal, M. M., Ammour, N., & Alajlan, N. (2018). Siamese-gan: Learning invariant representations for aerial vehicle image categorization. *Remote Sensing*, *10*(2), 351. 34, 35, 36

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). Surf: Speeded up robust features. In *European conference on computer vision* (pp. 404–417). 3, 22, 41, 172

Benz, U. C., Hofmann, P., Willhauck, G., Lingenfelder, I., & Heynen, M. (2004). Multi-resolution, object-oriented fuzzy analysis of remote sensing data for gis-ready information. *ISPRS Journal of photogrammetry and remote sensing*, *58*(3-4), 239–258. 15

Bhagavathy, S., & Manjunath, B. S. (2006). Modeling and detection of geospatial objects using texture motifs. *IEEE Transactions on Geoscience and Remote Sensing*, *44*(12), 3706–3715. 23

Bian, X., Chen, C., Tian, L., & Du, Q. (2017). Fusing local and global features for high-resolution scene classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *10*(6), 2889–2901. 34, 121, 124

Blaschke, T. (2001). What's wrong with pixels? some recent developments interfacing remote sensing and gis. *GeoBIT/GIS*, *6*, 12–17. 14

Blaschke, T. (2010). Object based image analysis for remote sensing. *ISPRS journal of photogrammetry and remote sensing*, *65*(1), 2–16. 1, 14, 15

Blaschke, T., Burnett, C., & Pekkarinen, A. (2004). New contextual approaches using image segmentation for objectbased classification. *Remote sensing image analysis: Including the spatial domain/Ed. De Meer, F. & de Jong, S.*. 14

Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*. 178

Bromley, J., Bentz, J. W., Bottou, L., Guyon, I., LeCun, Y., Moore, C., ... Shah,

R. (1993). Signature verification using a "siamese" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, *7*(04), 669–688. v, 35, 64, 69, 98, 102

Burnett, C., & Blaschke, T. (2003). A multi-scale segmentation/object relationship modelling methodology for landscape analysis. *Ecological modelling*, *168*(3), 233–249. 14

Castelluccio, M., Poggi, G., Sansone, C., & Verdoliva, L. (2015). Land use classification in remote sensing images by convolutional neural networks. *arXiv preprint arXiv:1508.00092*. 33

Chaib, S., Gu, Y., & Yao, H. (2016). An informative feature selection method based on sparse pca for vhr scene classification. *IEEE Geoscience and Remote Sensing Letters*, *13*(2), 147–151. 26

Chaib, S., Liu, H., Gu, Y., & Yao, H. (2017). Deep feature fusion for vhr remote sensing scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(8), 4775–4784. 33

Chan, T.-H., Jia, K., Gao, S., Lu, J., Zeng, Z., & Ma, Y. (2015). Pcanet: A simple deep learning baseline for image classification? *IEEE Transactions on Image Processing*, *24*(12), 5017–5032. 26

Chaudhuri, U., Banerjee, B., & Bhattacharya, A. (2019). Siamese graph convolutional network for content based remote sensing image retrieval. *Computer Vision and Image Understanding*, *184*, 22–30. 36

Chen, C., Zhang, B., Su, H., Li, W., & Wang, L. (2016). Land-use scene classification using multi-scale completed local binary patterns. *Signal, image and video processing*, *10*(4), 745–752. 24, 37

Chen, C., Zhou, L., Guo, J., Li, W., Su, H., & Guo, F. (2015). Gabor-filtering-based completed local binary patterns for land-use scene classification. In *2015 ieee international conference on multimedia big data* (pp. 324–329).

24

Chen, G., Zhang, X., Tan, X., Cheng, Y., Dai, F., Zhu, K., ... Wang, Q. (2018). Training small networks for scene classification of remote sensing images via knowledge distillation. *Remote Sensing*, *10*(5), 719. 34

Chen, J., Wang, C., Ma, Z., Chen, J., He, D., & Ackland, S. (2018). Remote sensing scene classification based on convolutional neural networks pre-trained using attention-guided sparse filters. *Remote Sensing*, *10*(2), 290. 37

Chen, S., & Tian, Y. (2014). Pyramid of spatial relatons for scene-level land use classification. *IEEE Transactions on Geoscience and Remote Sensing*, *53*(4), 1947–1957. 28

Chen, Z., Wang, S., Hou, X., & Shao, L. (2018). Recurrent transformer network for remote sensing scene categorisation. In *British machine vision conference*. ii, xiii, 9, 10, 11, 40, 62, 83, 84, 85, 94, 95, 117, 119, 120, 121, 124, 126, 133, 155, 157, 158, 159, 172, 173

Cheng, G., & Han, J. (2016). A survey on object detection in optical remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing*, *117*, 11–28. 15

Cheng, G., Han, J., Guo, L., & Liu, T. (2015). Learning coarse-to-fine sparselets for efficient object detection and scene classification. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1173–1181). 22

Cheng, G., Han, J., Guo, L., Liu, Z., Bu, S., & Ren, J. (2015). Effective and efficient midlevel visual elements-oriented land-use classification using vhr remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, *53*(8), 4238–4249. 22

Cheng, G., Han, J., Guo, L., Qian, X., Zhou, P., Yao, X., & Hu, X. (2013).

Object detection in remote sensing imagery using a discriminatively trained mixture model. *ISPRS Journal of Photogrammetry and Remote Sensing*, *85*, 32–43. 22

Cheng, G., Han, J., & Lu, X. (2017). Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, *105*(10), 1865–1883. v, vii, viii, ix, xii, xiii, xiv, 1, 7, 16, 17, 18, 19, 39, 42, 52, 53, 55, 56, 57, 59, 60, 63, 81, 82, 84, 87, 91, 92, 93, 99, 116, 117, 120, 121, 124, 127, 129, 154, 156, 157, 158, 159, 166, 167, 169

Cheng, G., Li, Z., Yao, X., Guo, L., & Wei, Z. (2017). Remote sensing image scene classification using bag of convolutional features. *IEEE Geoscience and Remote Sensing Letters*, *14*(10), 1735–1739. 28, 33, 81, 82, 88, 89, 117, 118

Cheng, G., Ma, C., Zhou, P., Yao, X., & Han, J. (2016). Scene classification of high resolution remote sensing images using convolutional neural networks. In *2016 ieee international geoscience and remote sensing symposium (igarss)* (pp. 767–770). 34

Cheng, G., Yang, C., Yao, X., Guo, L., & Han, J. (2018). When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative cnns. *IEEE Transactions on Geoscience and Remote Sensing*. 33, 34, 39, 52, 53, 54, 57, 81, 82, 83, 84, 85, 117, 119, 121, 124, 155, 156, 157, 159

Cheriyadat, A. M. (2014). Unsupervised feature learning for aerial scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, *52*(1), 439–451. 26

Clinton, N., Holt, A., Scarborough, J., Yan, L., Gong, P., et al. (2010). Accuracy assessment measures for object-based image segmentation goodness. *Photogramm. Eng. Remote Sens*, *76*(3), 289–299. 15

Cohen, T., & Welling, M. (2016). Group equivariant convolutional networks. In *International conference on machine learning* (pp. 2990–2999). 134, 140, 142

Cohen, T. S., & Welling, M. (2016). Steerable cnns. *arXiv preprint arXiv:1612.08498*. xiii, 134, 143

Congalton, R. G., & Green, K. (2002). *Assessing the accuracy of remotely sensed data: principles and practices*. CRC press. 15

Croft, W., & Cruse, D. A. (2004). *Cognitive linguistics*. Cambridge University Press. 64

Csurka, G., Dance, C., Fan, L., Willamowski, J., & Bray, C. (2004). Visual categorization with bags of keypoints. In (Vol. 1, p. 1-2). 3, 27

Cui, Y., Zhou, F., Wang, J., Liu, X., Lin, Y., & Belongie, S. (2017). Kernel pooling for convolutional neural networks. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 2921–2930). 39, 105

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection.. 3, 22

Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 4690–4699). 98, 108, 109, 110, 111, 116

Dhodhi, M. K., Saghri, J. A., Ahmad, I., & Ul-Mustafa, R. (1999). D-isodata: A distributed algorithm for unsupervised classification of remotely sensed data on network of workstations. *Journal of Parallel and Distributed Computing*, *59*(2), 280–301. 14

Dieleman, S., De Fauw, J., & Kavukcuoglu, K. (2016). Exploiting cyclic symmetry in convolutional neural networks. *arXiv preprint arXiv:1602.02660*. 134, 140, 142

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, *10*(7), 1895–1923. 21

Dietterich, T. G., Lathrop, R. H., & Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, *89*(1-2), 31–71. 69

dos Santos, J. A., & Penatti, O. A. B. (2010). Evaluating the potential of texture and color descriptors for remote sensing image retrieval and classification. In *In visapp (2)* (pp. 203–208). 22

Drăguţ, L., Csillik, O., Eisank, C., & Tiede, D. (2014). Automated parameterisation for multi-scale image segmentation on multiple layers. *ISPRS Journal of photogrammetry and Remote Sensing*, *88*, 119–127. 15

Draguţ, L., Tiede, D., & Levick, S. R. (2010). Esp: a tool to estimate scale parameter for multiresolution image segmentation of remotely sensed data. *International Journal of Geographical Information Science*, *24*(6), 859–871. 15

Dwivedi, R., Kandrika, S., & Ramana, K. (2004). Comparison of classifiers of remote-sensing data for land-use/land-cover mapping. *Current Science*, 328–335. 13

Edelman, A., Arias, T. A., & Smith, S. T. (1998). The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, *20*(2), 303–353. 153

Feitosa, R. Q., da Costa, G. A. O. P., Mota, G. L. A., & Feijó, B. (2011). Modeling alternatives for fuzzy markov chain-based classification of multitemporal remote sensing data. *Pattern Recognition Letters*, *32*(7), 927–940. 15

Fiori, S. (2010). Learning by natural gradient on noncompact matrix-type pseudo-riemannian manifolds. *IEEE transactions on neural networks*, *21*(5), 841–852. 147, 148

Friedl, M. A., & Brodley, C. E. (1997). Decision tree classification of land cover from remotely sensed data. *Remote sensing of environment*, *61*(3), 399–409. 13

Gao, Y., Beijbom, O., Zhang, N., & Darrell, T. (2016). Compact bilinear pooling. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 317–326). 39, 103, 147

Gebejes, A., & Huertas, R. (2013). Texture characterization based on grey-level co-occurrence matrix. *databases*, *9*, 10. 23

Gislason, P. O., Benediktsson, J. A., & Sveinsson, J. R. (2006). Random forests for land cover classification. *Pattern Recognition Letters*, *27*(4), 294–300. 13, 14

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (pp. 315–323). 31, 74, 107

Goncalves, M., Netto, M., Costa, J., & Zullo Junior, J. (2008). An unsupervised method of classifying remotely sensed images using kohonen self-organizing maps and agglomerative hierarchical clustering methods. *International Journal of Remote Sensing*, *29*(11), 3171–3207. 14

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. (http://www.deeplearningbook.org) 30, 43

Gu, Y., Wang, Y., & Li, Y. (2019). A survey on deep learning-driven remote sensing image scene understanding: Scene classification, scene retrieval and scene-guided object detection. *Applied Sciences*, *9*(10), 2110. 15

Han, J., Zhang, D., Cheng, G., Guo, L., & Ren, J. (2014). Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning. *IEEE Transactions on Geoscience and Remote Sensing*, *53*(6), 3325–3337. 16

Han, J., Zhou, P., Zhang, D., Cheng, G., Guo, L., Liu, Z., ... Wu, J. (2014). Efficient, simultaneous detection of multi-class geospatial targets based on visual saliency modeling and discriminative learning of sparse coding. *ISPRS Journal of Photogrammetry and Remote Sensing*, *89*, 37–48. 27, 37

Hay, G., Marceau, D., Dube, P., & Bouchard, A. (2001). A multiscale framework for landscape analysis: object-specific analysis and upscaling. *Landscape Ecology*, *16*(6), 471–490. 15

Hay, G. J., & Castilla, G. (2008). Geographic object-based image analysis (geobia): A new name for a new discipline. In *Object-based image analysis* (pp. 75–89). Springer. 15

He, H., Chen, M., Chen, T., & Li, D. (2018). Matching of remote sensing images with complex background variations via siamese convolutional neural network. *Remote Sensing*, *10*(2), 355. 36

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778). xiv, 30, 153, 154, 163, 164, 166, 167, 168, 169

He, N., Fang, L., Li, S., Plaza, A., & Plaza, J. (2018). Remote sensing scene classification using multilayer stacked covariance pooling. *IEEE Transactions on Geoscience and Remote Sensing*(99), 1–12. 36, 39, 83, 84, 85, 117, 119, 121, 124, 155, 157, 158

He, N., Fang, L., Li, S., Plaza, J., & Plaza, A. (2019). Skip-connected covariance network for remote sensing scene classification. *IEEE transactions on neural networks and learning systems*, *31*(5), 1461–1474. 39

Henriques, J. F., & Vedaldi, A. (2017). Warped convolutions: Efficient invariance to spatial transformations. In *Proceedings of the 34th international conference on machine learning-volume 70* (pp. 1461–1469). 134

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, *313*(5786), 504–507. 26, 27, 28, 29, 41

Hu, F., Xia, G.-S., Hu, J., & Zhang, L. (2015). Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sensing*, *7*(11), 14680–14707. 33, 34

Huang, C., Davis, L., & Townshend, J. (2002). An assessment of support vector machines for land cover classification. *International Journal of remote sensing*, *23*(4), 725–749. 14

Huang, J., Kumar, S. R., Mitra, M., Zhu, W.-J., & Zabih, R. (1997). Image indexing using color correlograms. In *Proceedings of ieee computer society conference on computer vision and pattern recognition* (pp. 762–768). 3, 23

Huang, L., Chen, C., Li, W., & Du, Q. (2016). Remote sensing image scene classification using multi-scale completed local binary patterns and fisher vectors. *Remote Sensing*, *8*(6), 483. 24, 37

Huang, Z., & Van Gool, L. (2017). A riemannian network for spd matrix learning. In *Thirty-first aaai conference on artificial intelligence*. 107, 108, 153

Huang, Z., Wang, R., Shan, S., Li, X., & Chen, X. (2015). Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification. In *International conference on machine learning* (pp. 720–729). 71

Hughes, L. H., Schmitt, M., Mou, L., Wang, Y., & Zhu, X. X. (2018). Identifying corresponding patches in sar and optical images with a pseudo-siamese cnn. *IEEE Geoscience and Remote Sensing Letters*, *15*(5), 784–788. 36

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*. 31, 32

Ionescu, C., Vantzos, O., & Sminchisescu, C. (2015). Matrix backpropagation for deep networks with structured layers. In *Proceedings of the ieee international conference on computer vision* (pp. 2965–2973). 38, 67, 75, 77, 112, 114

Itti, L., Koch, C., & Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, *20*(11), 1254–1259. 37

Jaderberg, M., Simonyan, K., Zisserman, A., et al. (2015). Spatial transformer networks. In *Advances in neural information processing systems* (pp. 2017–2025). 43, 44, 46, 49, 51, 61, 62

Jain, A. K., Ratha, N. K., & Lakshmanan, S. (1997). Object detection using gabor filters. *Pattern recognition*, *30*(2), 295–309. 22, 23, 41

Jegou, H., Perronnin, F., Douze, M., Sánchez, J., Perez, P., & Schmid, C. (2011). Aggregating local image descriptors into compact codes. *IEEE transactions on pattern analysis and machine intelligence*, *34*(9), 1704–1716. 3

Jolliffe, I. (2011). Principal component analysis. In *International encyclopedia of statistical science* (pp. 1094–1096). Springer. 3, 26, 41

Kim, M., Madden, M., & Warner, T. (2008). Estimation of optimal image object size for the segmentation of forest stands with multispectral ikonos imagery. In *Object-based image analysis* (pp. 291–307). Springer. 15

Koch, G., Zemel, R., & Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In *Icml deep learning workshop* (Vol. 2). 36

Kong, S., & Fowlkes, C. (2017). Low-rank bilinear pooling for fine-grained classification. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 365–374). 39, 47, 70, 103, 147

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information*

processing systems (pp. 1097–1105). 30, 53, 84, 117

Lampert, C. H., Nickisch, H., & Harmeling, S. (2009). Learning to detect unseen object classes by between-class attribute transfer. In *2009 ieee conference on computer vision and pattern recognition* (pp. 951–958). 178

Lang, S., & Tiede, D. (2007). Definiens developer–snapshot. *GIS. Business-GeoBIT*, *2007*(9), 34–37. 15

Laptev, D., Savinov, N., Buhmann, J. M., & Pollefeys, M. (2016). Ti-pooling: transformation-invariant pooling for feature learning in convolutional neural networks. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 289–297). 69

LeCun, Y., & Cortes, C. (2010). MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/. Retrieved 2016-01-14 14:24:11, from http://yann.lecun.com/exdb/mnist/ viii, 111

Li, E., Xia, J., Du, P., Lin, C., & Samat, A. (2017). Integrating multilayer features of convolutional neural networks for remote sensing scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(10), 5653–5665. 36

Li, H., Gu, H., Han, Y., & Yang, J. (2010). Object-oriented classification of high-resolution remote sensing imagery based on an improved colour structure code and a support vector machine. *International journal of remote sensing*, *31*(6), 1453–1470. 23

Li, M., Zang, S., Zhang, B., Li, S., & Wu, C. (2014). A review of remote sensing image classification techniques: The role of spatio-contextual information. *European Journal of Remote Sensing*, *47*(1), 389–411. 13

Li, P., Xie, J., Wang, Q., & Gao, Z. (2018). Towards faster training of global covariance pooling networks by iterative matrix square root normalization. In *Proceedings of the ieee conference on computer vision and pattern recog-*

nition (pp. 947–955). 39, 105, 145, 147, 149, 150, 151

Li, P., Xie, J., Wang, Q., & Zuo, W. (2017). Is second-order information helpful for large-scale visual recognition? In *Proceedings of the ieee international conference on computer vision* (pp. 2070–2078). 38, 39, 75, 84, 103, 104, 105, 106, 112, 145, 147, 148, 149

Li, W., & Du, Q. (2014). Gabor-filtering-based nearest regularized subspace for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *7*(4), 1012–1022. 24

Li, W., Fu, H., Yu, L., Gong, P., Feng, D., Li, C., & Clinton, N. (2016). Stacked autoencoder-based deep learning for remote-sensing image classification: a case study of african land-cover mapping. *International journal of remote sensing*, *37*(23), 5632–5646. 27

Li, Z., & Itti, L. (2010). Saliency and gist features for target detection in satellite images. *IEEE Transactions on Image Processing*, *20*(7), 2017–2029. 23, 37

Lillesand, T., Kiefer, R. W., & Chipman, J. (2015). *Remote sensing and image interpretation*. John Wiley & Sons. 13

Lin, T.-Y., & Maji, S. (2017). Improved bilinear pooling with cnns. *arXiv preprint arXiv:1707.06772*. 38, 67, 75, 105, 112, 147, 149

Lin, T.-Y., RoyChowdhury, A., & Maji, S. (2015). Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the ieee international conference on computer vision* (pp. 1449–1457). v, xiv, 38, 47, 50, 67, 103, 147, 162, 163, 164

Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., & Song, L. (2017). Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 212–220). 98, 108, 109, 110

Liu, X., Zhou, Y., Zhao, J., Yao, R., Liu, B., & Zheng, Y. (2019). Siamese convolutional neural networks for remote sensing scene classification. *IEEE Geoscience and Remote Sensing Letters*, *16*(8), 1200–1204. 36

Liu, Y., & Huang, C. (2017). Scene classification via triplet networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *11*(1), 220–237. 117

Liu, Y., Suen, C. Y., Liu, Y., & Ding, L. (2018). Scene classification using hierarchical wasserstein cnn. *IEEE Transactions on Geoscience and Remote Sensing*, *57*(5), 2494–2509. 34

Lizarazo, I. (2014). Accuracy assessment of object-based image classification: another step. *International Journal of Remote Sensing*, *35*(16), 6135–6156. 15

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, *60*(2), 91–110. 3, 21, 22, 41, 172

Lu, X., Sun, H., & Zheng, X. (2019). A feature aggregation convolutional neural network for remote sensing scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, *57*(10), 7894–7906. 33

Ma, X., Wang, H., & Geng, J. (2016). Spectral–spatial classification of hyperspectral image based on deep auto-encoder. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *9*(9), 4073–4085. 27

MacLean, M. G., & Congalton, R. G. (2012). Map accuracy assessment issues when using an object-oriented approach. In (pp. 1–5). 15

MacQueen, J., et al. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, *1*(14). 3, 26, 27, 41

Marceau, D. J., Howarth, P. J., Dubois, J.-M. M., Gratton, D. J., et al. (1990). Evaluation of the grey-level co-occurrence matrix method for land-cover

classification using spot imagery. *IEEE Transactions on Geoscience and Remote Sensing*, *28*(4), 513–519. 23

Marconcini, M., Camps-Valls, G., & Bruzzone, L. (2009). A composite semisupervised svm for classification of hyperspectral images. *IEEE Geoscience and Remote Sensing Letters*, *6*(2), 234–238. 14

Marmanis, D., Datcu, M., Esch, T., & Stilla, U. (2015). Deep learning earth observation classification using imagenet pretrained networks. *IEEE Geoscience and Remote Sensing Letters*, *13*(1), 105–109. 33, 34

McCabe, M. F., Rodell, M., Alsdorf, D. E., Miralles, D. G., Uijlenhoet, R., Wagner, W., ... others (2017). The future of earth observation in hydrology. *Hydrology and earth system sciences*, *21*(7), 3879. 1

McIver, D., & Friedl, M. (2002). Using prior probabilities in decision-tree classification of remotely sensed data. *Remote sensing of Environment*, *81*(2-3), 253–261. 13

Mekhalfi, M. L., Melgani, F., Bazi, Y., & Alajlan, N. (2015). Land-use classification with compressive sensing multifeature fusion. *IEEE Geoscience and Remote Sensing Letters*, *12*(10), 2155–2159. 26

Minetto, R., Segundo, M. P., & Sarkar, S. (2019). Hydra: An ensemble of convolutional neural networks for geospatial land classification. *IEEE Transactions on Geoscience and Remote Sensing*, *57*(9), 6530–6541. 34

Mukuta, Y., & Harada, T. (2019). Invariant tensor feature coding. *arXiv preprint arXiv:1906.01857*. xiii, 144, 145

Musci, M., Feitosa, R. Q., Costa, G. A., & Velloso, M. L. F. (2013). Assessment of binary coding techniques for texture characterization in remote sensing imagery. *IEEE Geoscience and Remote Sensing Letters*, *10*(6), 1607–1611. 23, 24

Oliva, A., & Torralba, A. (2001). Modeling the shape of the scene: A holistic

representation of the spatial envelope. *International journal of computer vision*, *42*(3), 145–175. 3, 23, 41

Olshausen, B. A., & Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, *37*(23), 3311–3325. 26, 41

Othman, E., Bazi, Y., Alajlan, N., Alhichri, H., & Melgani, F. (2016). Using convolutional features and a sparse autoencoder for land-use scene classification. *International Journal of Remote Sensing*, *37*(10), 2149–2167. 27

Othman, E., Bazi, Y., Melgani, F., Alhichri, H., Alajlan, N., & Zuair, M. (2017). Domain adaptation network for cross-scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(8), 4441–4456. 34

Pal, M., & Mather, P. (2005). Support vector machines for classification in remote sensing. *International Journal of Remote Sensing*, *26*(5), 1007–1011. 14

Penatti, O. A., Nogueira, K., & dos Santos, J. A. (2015). Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In *Proceedings of the ieee conference on computer vision and pattern recognition workshops* (pp. 44–51). 23, 33

Penatti, O. A., Valle, E., & Torres, R. d. S. (2012). Comparative study of global color and texture descriptors for web image retrieval. *Journal of visual communication and image representation*, *23*(2), 359–380. 23

Perez, L., & Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*. 43, 134, 140

Qi, K., Xiaochun, Z., Baiyan, W., & Wu, H. (2016). Sparse coding-based correlaton model for land-use scene classification in high-resolution remote-sensing images. *Journal of Applied Remote Sensing*, *10*(4), 042005. 27

Radoux, J., Bogaert, P., Fasbender, D., & Defourny, P. (2011). Thematic accuracy

assessment of geographic object-based image classification. *International Journal of Geographical Information Science*, *25*(6), 895–911. 15

Risojević, V., & Babić, Z. (2012). Fusion of global and local descriptors for remote sensing image classification. *IEEE Geoscience and Remote Sensing Letters*, *10*(4), 836–840. 22, 24

Risojević, V., Momić, S., & Babić, Z. (2011). Gabor descriptors for aerial image classification. In *International conference on adaptive and natural computing algorithms* (pp. 51–60). 23

Rollet, R., Benie, G., Li, W., Wang, S., & Boucher, J. (1998). Image classification algorithm based on the rbf neural network and k-means. *International Journal of Remote Sensing*, *19*(15), 3003–3009. 14

Sánchez, J., Perronnin, F., Mensink, T., & Verbeek, J. (2013). Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, *105*(3), 222–245. 3

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the ieee international conference on computer vision* (pp. 618–626). ix, 92, 129, 130, 169

Settle, J., & Briggs, S. (1987). Fast maximum likelihood classification of remotely-sensed imagery. *International Journal of Remote Sensing*, *8*(5), 723–734. 13

Shahriari, M., & Bergevin, R. (2017). Land-use scene classification: a comparative study on bag of visual word framework. *Multimedia Tools and Applications*, *76*(21), 23059–23075. 28

Sheng, G., Yang, W., Xu, T., & Sun, H. (2012). High-resolution satellite scene classification using a sparse coding based multiple feature combination. *International journal of remote sensing*, *33*(8), 2395–2412. 26

Shi, Z., Yu, X., Jiang, Z., & Li, B. (2013). Ship detection in high-resolution optical imagery based on anomaly detector and local shape feature. *IEEE Transactions on Geoscience and Remote Sensing*, *52*(8), 4511–4523. 22

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. 30, 48, 50, 51, 53, 80, 84, 85, 115, 117, 118, 146, 153, 154, 157, 159, 164, 166, 167, 168, 169

Sokolic, J., Giryes, R., Sapiro, G., & Rodrigues, M. (2017). Generalization error of invariant classifiers. In *Artificial intelligence and statistics* (pp. 1094–1103). 144

Stehling, R. O., Nascimento, M. A., & Falcão, A. X. (2002). A compact and efficient image retrieval approach based on border/interior pixel classification. In *Proceedings of the eleventh international conference on information and knowledge management* (pp. 102–109). 3, 23

Sun, H., Li, S., Zheng, X., & Lu, X. (2019). Remote sensing scene classification by gated bidirectional network. *IEEE Transactions on Geoscience and Remote Sensing*, *58*(1), 82–96. 34

Swain, M. J., & Ballard, D. H. (1991). Color indexing. *International journal of computer vision*, *7*(1), 11–32. 3, 22, 41

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1–9). 30, 53, 84, 117

Tang, J., Wang, L., & Myint, S. (2007). Improving urban classification through fuzzy supervised classification and spectral mixture analysis. *International Journal of Remote Sensing*, *28*(18), 4047–4063. 14

Tanner, M. A., & Wong, W. H. (1987). The calculation of posterior distribu-

tions by data augmentation. *Journal of the American statistical Association*, *82*(398), 528–540. 43, 134, 140

Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2016). Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*. 32

Ungerer, F., & Schmid, H.-J. (2013). *An introduction to cognitive linguistics*. Routledge. 64

Varior, R. R., Haloi, M., & Wang, G. (2016). Gated siamese convolutional neural network architecture for human re-identification. In *European conference on computer vision* (pp. 791–808). 36

Wang, F., Cheng, J., Liu, W., & Liu, H. (2018). Additive margin softmax for face verification. *IEEE Signal Processing Letters*, *25*(7), 926–930. 98, 108, 109, 110

Wang, F., Xiang, X., Cheng, J., & Yuille, A. L. (2017). Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th acm international conference on multimedia* (pp. 1041–1049). 97, 108, 109, 110

Wang, G., Fan, B., Xiang, S., & Pan, C. (2017). Aggregating rich hierarchical features for scene classification in remote sensing imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *10*(9), 4104–4115. 36

Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., ... Liu, W. (2018). Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 5265–5274). 98, 108, 109, 110

Wang, J., Shen, L., Qiao, W., Dai, Y., & Li, Z. (2019). Deep feature fusion with integration of residual connection and attention model for classification of vhr remote sensing images. *Remote Sensing*, *11*(13), 1617. 37, 38

Wang, Q., Liu, S., Chanussot, J., & Li, X. (2018). Scene classification with

recurrent attention of vhr remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, *57*(2), 1155–1167. 17, 19, 20, 21, 38, 154, 156, 161

Wang, S., Guan, Y., & Shao, L. (2020). Multi-granularity canonical appearance pooling for remote sensing scene classification. *IEEE Transactions on Image Processing*, *29*, 5396–5407. ii, 10, 11, 21, 40, 105, 106, 107, 117, 119, 120, 121, 124, 125, 133, 147, 148, 149, 154, 155, 156, 158, 172, 173

Wang, S., Long, Y., Guan, Y., & Shao, L. (-). Covaraince feature embedding for remote sensing scene image classification. *in submission of IEEE Transactions on Geoscience and Remote Sensing*. ii, 10, 11, 21, 40, 133, 147, 155, 156, 172, 175

Wang, S., Ren, Y., Parr, G., Guan, Y., & Shao, L. (2020). Invariant deep compressible covariance pooling for aerial scene categorization. *IEEE Transactions on Geoscience and Remote Sensing*. ii, 11, 21, 40, 172, 173, 174

Watkins, D. S. (2004). *Fundamentals of matrix computations* (Vol. 64). John Wiley & Sons. 106

Wei, X., Zhang, Y., Gong, Y., Zhang, J., & Zheng, N. (2018). Grassmann pooling as compact homogeneous bilinear pooling for fine-grained visual classification. In *Proceedings of the european conference on computer vision (eccv)* (pp. 355–370). 39, 105, 167

Wen, Y., Zhang, K., Li, Z., & Qiao, Y. (2016). A discriminative feature learning approach for deep face recognition. In *European conference on computer vision* (pp. 499–515). 98

Wu, C. (2004). Normalized spectral mixture analysis for monitoring urban composition using etm+ imagery. *Remote Sensing of Environment*, *93*(4), 480–492. 14

Wu, C., & Murray, A. T. (2003). Estimating impervious surface distribution by

spectral mixture analysis. *Remote sensing of Environment*, *84*(4), 493–505. 14

Wu, J., Yu, Y., Huang, C., & Yu, K. (2015). Deep multiple instance learning for image classification and auto-annotation. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 3460–3469). 69

Wu, Y., & He, K. (2018). Group normalization. In *Proceedings of the european conference on computer vision (eccv)* (pp. 3–19). 32

Xia, G.-S., Hu, J., Hu, F., Shi, B., Bai, X., Zhong, Y., ... Lu, X. (2017). Aid: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(7), 3965–3981. ix, xiii, 7, 17, 18, 19, 39, 52, 53, 54, 64, 84, 86, 116, 120, 121, 123, 127, 129, 154, 156, 157, 160

Xia, G.-S., Tong, X.-Y., Hu, F., Zhong, Y., Datcu, M., & Zhang, L. (2017). Exploiting deep features for remote sensing image retrieval: A systematic investigation. *arXiv preprint arXiv:1707.07321*, *2*. 16

Xie, J., He, N., Fang, L., & Plaza, A. (2019). Scale-free convolutional neural network for remote sensing scene classification. *IEEE Transactions on Geoscience and Remote Sensing*. 34

Xu, R., Tao, Y., Lu, Z., & Zhong, Y. (2018). Attention-mechanism-containing neural networks for high-resolution remote sensing image classification. *Remote Sensing*, *10*(10), 1602. 37

Yang, C.-C., Prasher, S. O., Enright, P., Madramootoo, C., Burgess, M., Goel, P. K., & Callum, I. (2003). Application of decision tree technology for image classification using remote sensing data. *Agricultural Systems*, *76*(3), 1101–1117. 14

Yang, Y., & Newsam, S. (2010). Bag-of-visual-words and spatial extensions for

land-use classification. In *Proceedings of the 18th sigspatial international conference on advances in geographic information systems* (pp. 270–279). xiii, 17, 19, 20, 21, 28, 53, 85, 116, 124, 128, 154, 156, 161, 175

Yu, X., Wu, X., Luo, C., & Ren, P. (2017). Deep learning in remote sensing scene classification: a data augmentation enhanced convolutional neural network framework. *GIScience & Remote Sensing*, *54*(5), 741–758. 34

Yu, Y., & Liu, F. (2018). A two-stream deep fusion framework for high-resolution aerial scene classification. *Computational intelligence and neuroscience*, *2018*. 34, 117, 118, 120, 121, 124, 125, 155, 158

Yuan, F., Sawaya, K. E., Loeffelholz, B. C., & Bauer, M. E. (2005). Land cover classification and change analysis of the twin cities (minnesota) metropolitan area by multitemporal landsat remote sensing. *Remote sensing of Environment*, *98*(2-3), 317–328. 14

Zeng, D., Chen, S., Chen, B., & Li, S. (2018). Improving remote sensing scene classification by integrating global-context and local-object features. *Remote Sensing*, *10*(5), 734. 34, 124, 125, 155, 156, 157

Zhang, B., Zhang, Y., & Wang, S. (2019). A lightweight and discriminative model for remote sensing scene classification with multidilation pooling module. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *12*(8), 2636–2653. 34

Zhang, F., Du, B., & Zhang, L. (2015a). Saliency-guided unsupervised feature learning for scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, *53*(4), 2175–2184. 27, 37

Zhang, F., Du, B., & Zhang, L. (2015b). Scene classification via a gradient boosting random convolutional network framework. *IEEE Transactions on Geoscience and Remote Sensing*, *54*(3), 1793–1802. 34

Zhang, J., & Foody, G. (1998). A fuzzy classification of sub-urban land cover

from remotely sensed imagery. *International journal of remote sensing*, *19*(14), 2721–2738. 14

Zhang, J., Li, T., Lu, X., & Cheng, Z. (2016). Semantic classification of high-resolution remote-sensing images based on mid-level features. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *9*(6), 2343–2353. 28

Zhang, L., Zhang, L., & Du, B. (2016). Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geoscience and Remote Sensing Magazine*, *4*(2), 22–40. 33

Zhang, W., Tang, P., & Zhao, L. (2019). Remote sensing image scene classification using cnn-capsnet. *Remote Sensing*, *11*(5), 494. 117, 118, 121, 124, 155, 157

Zhang, Y., Sun, X., Wang, H., & Fu, K. (2013). High-resolution remote-sensing image classification via an approximate earth mover's distance-based bag-of-features model. *IEEE Geoscience and Remote Sensing Letters*, *10*(5), 1055–1059. 28

Zhang, Z., Vosselman, G., Gerke, M., Tuia, D., & Yang, M. Y. (2018). Change detection between multimodal remote sensing data using siamese cnn. *arXiv preprint arXiv:1807.09562*. 36

Zhao, B., Zhong, Y., & Zhang, L. (2016). A spectral–structural bag-of-features scene classifier for very high spatial resolution remote sensing imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, *116*, 73–85. 28

Zhao, L., Tang, P., & Huo, L. (2014). A 2-d wavelet decomposition-based bag-of-visual-words model for land-use scene classification. *International Journal of Remote Sensing*, *35*(6), 2296–2310. 28

Zhao, L., Tang, P., & Huo, L. (2016). Feature significance-based multi bag-of-visual-words model for remote sensing image scene classification. *Journal*

of Applied Remote Sensing, *10*(3), 035004. 28

Zhao, L.-J., Tang, P., & Huo, L.-Z. (2014). Land-use scene classification using a concentric circle-structured multiscale bag-of-visual-words model. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *7*(12), 4620–4631. 28, 37

Zhao, W., & Du, S. (2016). Scene classification using multi-scale deeply described visual words. *International Journal of Remote Sensing*, *37*(17), 4119–4131. 36

Zheng, X., Sun, X., Fu, K., & Wang, H. (2012). Automatic annotation of satellite images via multifeature joint sparse coding with spatial relation constraint. *IEEE Geoscience and Remote Sensing Letters*, *10*(4), 652–656. 26

Zheng, X., Yuan, Y., & Lu, X. (2019). A deep scene representation for aerial scene classification. *IEEE Transactions on Geoscience and Remote Sensing*. 36

Zhu, H., & Basir, O. (2005). An adaptive fuzzy evidential nearest neighbor formulation for classifying remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, *43*(8), 1874–1889. 13

Zhu, Q., Zhong, Y., Zhao, B., Xia, G.-S., & Zhang, L. (2016). Bag-of-visual-words scene classifier with local and global features for high spatial resolution remote sensing imagery. *IEEE Geoscience and Remote Sensing Letters*, *13*(6), 747–751. 24, 28

Zhu, X. X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., & Fraundorfer, F. (2017). Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, *5*(4), 8–36. 33

Zou, J., Li, W., Chen, C., & Du, Q. (2016). Scene classification using local and global features with collaborative representation fusion. *Information Sciences*, *348*, 209–226. 24, 26, 28

A | Appendix-Deep Learning Toolbox

All frameworks incorporated in this these are developed with using GPU version of TensorFlow (Abadi et al., 2016). Tensorflow is an open-source software library which is one of the most successful deep learning toolbox for both research and industry. The software is sourced by Google and involves a broad range of applications. Examples include the recognition of images and speech, the processing of natural language, and etc.

The main reason for the popularity of TensorFlow is that it supports multi-interface with other popular programming languages such as C++ and Python. After the announcement of TensorFlow 1.0, more interfaces will be supported. For example, the interfaces of R and Java. TensorFlow 2.0 has been announced by Google in January 2019 and becomes available in September 2019.

Additionally, several distinct advantages need to be noted. TensorFlow supports to be trained by Google Cloud, where the powerful GPUs are available. TensorFlow is flexible for users to construct their frameworks from either scratch or using any off-the-shelf architectures. TensorFlow provides an intuitive visualisation that is extremely helpful for debugging the complicated graphs and monitoring the training process. More details can be found on the official site with https://www.tensorflow.org/.