FINE-GRAINED OBJECT RECOGNITION IN REMOTE SENSING IMAGERY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

By
Gencer Sümbül
June 2018

FINE-GRAINED OBJECT RECOGNITION IN REMOTE SENSING IMAGERY
By Gencer Sümbül
June 2018

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Selim Aksoy(Advisor)

Ramazan Gökberk Cinbiş(Co-Advisor)

A. Aydın Alatan

Hamdi Dibeklioğlu

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

FINE-GRAINED OBJECT RECOGNITION IN REMOTE SENSING IMAGERY

Gencer Sümbül
M.S. in Computer Engineering
Advisor: Selim Aksoy
Co-Advisor: Ramazan Gökberk Cinbiş
June 2018

Fine-grained object recognition aims to determine the type of an object in domains with a large number of sub-categories. The steadily increase in spatial and spectral resolution entailing new details in remote sensing image data, and consequently more diversified target object classes having subtle differences makes it an emerging application. For the approaches using images from a single domain, widespread fully supervised algorithms do not completely fit into accomplishing this problem since target object classes tend to have low between-class variance and high within-class variance with small sample sizes. As an even more arduous task, a method for zero-shot learning (ZSL), in which identification of unseen sub-categories is tackled by associating them with previously learned seen sub-categories when there is no training example for some of the classes, is proposed. More specifically, our method learns a compatibility function between image representation obtained from a deep convolutional neural network and the semantics of target object sub-categories explained by auxiliary information gathered from complementary sources. Knowledge transfer for unseen classes is carried out by maximizing this function throughout the inference. Furthermore, benefitting from multiple image sensors can overcome the drawbacks of closely intertwined sub-categories that limits the object recognition performance. However, since multiple images may be acquired from different sensors under different conditions at different spatial and spectral resolutions, they may be geometrically unaligned correctly due to seasonal changes, different viewing geometry, acquisition noise, an imperfection of sensors, different atmospheric conditions etc. To address these challenges, a neural network model that aims to correctly align images acquired from different sources and to learn the classification rules in a unified framework simultaneously is proposed. In this network, one of the sources is used as the reference and the others are aligned with the reference image at representation

level throughout a learned weighting mechanism. At the end, classification of sub-categories is carried out with a feature-level fusion of representations from the source region and estimated multiple target regions. Experimental analysis conducted on a newly proposed data set shows that both zero-shot learning algorithm and the multi-source fine-grained object recognition algorithm give promising results.

Keywords: Fine-grained classification, zero-shot learning, multisource, remote sensing, object recognition.

ÖZET

UZAKTAN ALGILANMIŞ GÖRÜNTÜLERDE İNCE TANELİ NESNE TANIMA

Gencer Sümbül
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Danışmanı: Selim Aksoy
İkinci Tez Danışmanı: Ramazan Gökberk Cinbiş
Haziran 2018

İnce taneli nesne tanıma, çok sayıda alt kategori arasından hedef nesnenin türünü belirleme görevi ile ilgilenir. Uzaktan algılanmış görüntülerde yeni detayların ortaya çıkmasını sağlayan uzamsal ve spektral çözünürlükteki sürekli artış ve zor algılanan farklara sahip olan daha çeşitli hedef nesne sınıflarının ortaya çıkışı bunu yeni bir uygulama haline getirmektedir. Tek bir veri kaynağından alınan görüntüleri kullanan yaklaşımlarda, denetimli algoritmalar, düşük sınıflar arası değişinti ve yüksek sınıf içi değişintiye ek olarak küçük örneklem büyüklüğü nedeniyle bu problemi tam olarak çözemez. Bu sorunların yanı sıra, daha da zorlu bir görev olarak sınıfların bazıları için hiçbir eğitim örneği bulunmayan örneksiz öğrenme problemi ele alınabilir. Örneksiz öğrenme, daha önce öğrendiği alt kategorilerle, eğitim örnekleri olmayan yeni alt kategorileri ilişkilendirerek bir tanıma modeli oluşturmayı amaçlamaktadır. Bu ilişkiyi kurmak için geliştirdiğimiz yöntem, derin bir evrişimsel sinir ağından elde edilen görüntü temsili ile sınıfların anlamsal özelliklerini tanımlayan yardımcı bilgiler arasında bir uyumluluk fonksiyonu öğrenir. Eğitim örneği olmayan sınıflar için bilgi aktarımı, çıkarım esnasında bu fonksiyonun en büyüklenmesi ile gerçekleştirilir. Örneksiz öğrenmeye ek olarak çoklu veri kaynaklarından faydalanmak, nesne tanıma performansını sınırlayan alt kategorilerin benzerliğinin yarattığı olumsuz etkilerin üstesinden gelebilir. Ancak bu durum aynı zamanda yeni sorunları ortaya çıkarmaktadır. Farklı uzamsal ve spektral çözünürlüklerde, farklı koşullar altında ve farklı sensörlerden elde edilen görüntüler; mevsimsel değişiklikler, farklı görüntüleme geometrisi, edinim gürültüsü, sensörlerin kusurları, farklı atmosfer koşulları vb. nedeniyle geometrik olarak doğru şekilde çakıştırılamayabilirler. Bu çalışmada farklı kaynaklardan edinilen görüntüleri doğru bir şekilde çakıştırmayı ve sınıflandırma kurallarını aynı anda tek bir çerçevede öğrenmeyi amaçlayan bir sinir ağı modeli önerilmiştir. Bunu yapmak

için bir görüntü kaynak görüntü olarak kullanılır. Diğer görüntülerde olası bölge önerilerinin temsilleri ağırlıklandırılarak kaynak görüntü ile çakışan doğru uzamsal bölge kestirilir. Kaynak görüntüsünden çıkarılan derin özelliklerin yardımıyla gerekli ağırlıklar bulunur. Sonunda, alt kategorilerin sınıflandırılması, kaynak bölgeden ve kestirilmiş hedef bölgelerden çıkarılan temsillerin kaynaştırılması ile gerçekleştirilir. Yeni önerilen bir veri kümesi üzerinde yapılan deneysel analiz, her iki yöntemin de başarılı sonuçlar verdiğini göstermektedir.

Anahtar sözcükler: İnce taneli sınıflandırma, örneksiz öğrenme, çok kaynaklı veri, uzaktan algılama, nesne tanıma.

Acknowledgement

First, I am deeply indebted to my advisors, Assoc. Prof. Dr. Selim Aksoy and Asst. Prof. Ramazan Gökberk Cinbiş for their patience, time, encouragement and kindness from the first moment of my M.S. studies. Without their priceless guidance, this thesis would not have been possible.

I would like to thank the members of my thesis committee, Prof. Dr. A. Aydın Alatan and Asst. Prof. Hamdi Dibeklioğlu for their interest to my study, helpful feedbacks and comments.

I am grateful to my comrades from EA427, Ali Burak Ünal, Bulut Aygüneş, Caner Mercan, Iman Deznaby, Mert Bülent Sarıyıldız, Onur Taşar, Yarkın Deniz Çetin, and Yiğit Özen for their help and all the enjoyable moments.

I am also thankful to my beloved family for their support, love and most importantly their understanding.

Last but most important, I would like to record my sincere and profound gratitude to my wife, Kimya, for his innumerable sacrifices while casting her bread upon the waters, for all the sleepless nights, for believing my success with her whole heart, for being my best friend, editor, sounding board, muse and most significantly for her inestimable love.

This work was supported in part by the TUBITAK Grant 116E445 and in part by the BAGEP Award of the Science Academy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Statement

Contemporary cameras used for remote sensing allows capturing landcover images at very high spatial resolution with rich spectral information. Consequently, the increased resolution has exposed new details, and has enabled new object classes to be detected and recognized in aerial and satellite images. The ability to collect such imagery opens the door to making detailed observations and inferences about objects through aerial and satellite images.

Automatic object recognition has been one of the most popular problems in remote sensing image analysis where the algorithms aim to map visual characteristics observed in image data to object classes. The main goal of these algorithms is to find distinctive image features that can discriminate between different object categories. Both the traditional methods that use various hand-crafted features with classifiers such as support vector machines and random forests, and the more recent approaches that use deep neural networks that aim to learn both the features and the classification rules have been shown to achieve remarkable performance in data sets acquired from different sources [1,2]. A common characteristic of such data sets in the remote sensing literature is that they contain relatively

distinctive classes, with a balanced mixture of urban, rural, agricultural, coastal, etc., land cover/use classes and object categories, for which sufficient training data to formulate a supervised learning task are often available. For example, commonly used benchmark data sets (e.g., UC Merced and AID [2]) pose the classification problem as the assignment of a test image patch to the most relevant category among the candidates such as agricultural, beach, forest, freeway, golf course, harbor, parking lot, residential, and river. Such data sets have been beneficial in advancing the state-of-the-art by enabling objective comparisons of different approaches. However, the unconstrained variety of remotely sensed imagery still leads to many open problems.

An important problem that is enabled by enrichment in the sensor technology is *fine-grained object recognition*. A practical definition of fine-grained object recognition is object recognition in the domain of a large number of closely related categories. Figure 1.1 shows examples from the street trees data set used in this thesis. As seen from the 16 test classes among 40 types of street trees included in this data set, differentiating the sub-category can be a very difficult task even when very high spatial or varied spectral resolution image data are used. It is envisioned that the fine-grained object recognition task will gain importance in the coming years as both the diversity and the subtleness of target object classes increase with the constantly improving spatial and spectral resolution. However, it is currently not clear how the existing classification models will behave for such recognition tasks.

Fine-grained object recognition differs from other classification and recognition tasks with respect to two important aspects: small sample size and class imbalance. Remote sensing has traditionally enjoyed abundance of data, but obtaining label information has always been an important bottleneck in classification studies. The acquisition costs for spatially distributed data can make sample collection via site visits practically unfeasible when one needs to travel unpredictably long distances to find sufficient number of examples [3]. Class imbalance in training data can also cause problems during supervised learning, particularly when the label frequencies observed in training data do not necessarily reflect the distribution of the labels among future unseen test instances.

Figure 1.1: Example RGB instances for 16 classes from the fine-grained street trees data set used in this thesis. For each class, a ground-view photograph and two 25×25 pixel patches from aerial RGB imagery with 1-foot spatial resolution are shown. From left to right and top to bottom: *London Plane, Callery Pear, Horse Chestnut, Common Hawthorn, European Hornbeam, Sycamore Maple, Pacific Maple, Mountain Ash, Green Ash, Kousa Dogwood, Autumn Cherry, Douglas Fir, Orchard Apple, Apple Serviceberry, Scarlet Oak, Japanese Snowbell.*

Besides these problems, an even more extreme scenario is the *zero-shot learning* task where *no training examples* exists for some of the classes. Zero-shot learning for fine-grained object recognition has received very little attention in the remote sensing literature even though it is a highly probable scenario where new object categories can be introduced after the training phase or when no training examples exists for several rare classes that are still of interest for different applications.

Zero-shot learning aims to build a recognition model for new categories that have no training examples by relating them to categories that were previously learned [4]. It is different from the domain adaptation and supervised transfer learning tasks [5] where at least some training examples are available for the target classes or the same classes exist in the target domain. Since no training instances are available for the test categories in zero-shot learning, image data alone are not sufficient to form the association between the *unseen* and *seen* classes. Thus, it is needed to find new sources of auxiliary information that can act as an intermediate layer for building this association. Attributes [6, 7] have been the most popular source of auxiliary information in the computer vision literature where zero-shot learning has recently become a popular problem [8]. Attributes often refer to well-known common characteristics of objects, and can be acquired by human annotation. They have been successfully used in zero-shot classification tasks for the identification of different bird or dog species or indoor and outdoor scene categories in computer vision [8]. An important requirement in the design of the attributes is that the required human effort should be small because otherwise resorting to supervised or semi-supervised learning algorithms by collecting training samples can be a viable alternative. An alternative is to use automatic processing of other modalities such as text documents [9]. New relevant attributes that exploit the peculiarities of overhead imagery should be designed for target object categories of interest in remotely sensed data sets.

In addition to aforementioned difficulties, use of image data from single sensor limits the fine-grained recognition performance because low between-class and high within-class variance make classes closely intertwined. To overcome this hurdle, information from multiple sources can decrease the effect of this entanglement

that causes uncertainty for object recognition. For instance, although RGB images give spatial contextual information, using hyperspectral and multispectral images will be beneficial for spectral characteristics of categories acquired from different bands. Besides, light detection and ranging (LIDAR) based elevation models can give information about the height and shape character for classes. Therefore, using multisource images is a very popular scenario in remote sensing image analysis.

Although using multiple sources has advantages, it also brings many problems to resolve in order to gain whole information from different sources. Multiple images may be acquired from different sensors under different conditions at different spatial and spectral resolutions so that images may not be geometrically aligned correctly because of seasonal changes, different viewing geometry, acquisition noise, imperfection of sensors, different atmospheric conditions etc. Thus, correspondence of all ground truth labels in multiple images is often not possible.

At the first stage of this thesis, as the most challenging scenario which is zero-shot learning in remotely sensed images acquired from a single source, the proposed approach uses a bilinear function that models the compatibility between the visual characteristics observed in the input image data and the auxiliary information that describes the semantics of the classes of interest. The image content is modeled by features extracted using a convolutional neural network that is learned from the seen classes in the training data. The auxiliary information is gathered from three complementary domains: manually annotated attributes that reflect the domain expertise, a natural language model trained over large text corpora, and a hierarchical representation of scientific taxonomy. When the between-class variance is low and the within-class variance is high, a single source of information is often not sufficient. Thus, different representations are exploited and their effectiveness are evaluated comparatively. Additionally, how the compatibility function can be estimated from the seen classes by using the maximum likelihood principle during the learning phase, and how knowledge transfer can be performed for the unseen classes by maximizing this function during the inference phase are shown. Finally, a realistic performance evaluation in a challenging

setup by using different partitionings of the data, making sure that the zero-shot (unseen) categories are well-isolated from the rest of the classes during both learning and parameter tuning [10] are also presented.

At the second stage of this thesis, a neural network model that aims to correctly align remotely sensed images acquired from different sources, to learn deep representations of them, and to learn the classification rules in one framework at the same time is proposed. To do so, one image is used as source image which is correctly aligned. For others, named as target images, correct spatial region is estimated by weighting representations of possible region proposals. Required weights are found with the help of the deep features extracted from source image. At the end, classification of sub-categories is done with respect to feature-level fusion of the deep representations coming from source region and estimated mutiple target regions. How additional image sources affect fully supervised object recognition and zero-shot learning performance is also discussed.

1.2 Contributions

Our major contributions are as follows:

- First, to the best of our knowledge, we present the first study on fine-grained object recognition with zero-shot learning in remotely sensed imagery.
- Second, we propose a new approach for zero-shot learning that uses a bilinear function for modelling the compatibility between the visual characteristics of the images and the auxiliary information describing the semantics of the classes. The image content is modeled by features extracted using a convolutional neural network. The auxiliary information is gathered from three complementary domains: manually annotated attributes, a natural language modeland a hierarchical representation of scientific taxonomy.
- Third, we present a realistic performance evaluation in a challenging setup by using different partitionings of the data, making sure that the zero-shot

(unseen) categories are well-isolated from the rest of the classes during both learning and parameter tuning.

- Fourth, we present a new data set that contains 40 different types of trees with 1 foot spatial resolution RGB images, 1.84 meter spatial resolution multispectral images and point-based ground truth. Since RGB images can be used for both zero-shot learning scenario by sparing some classes as unseen and fine-grained object recognition studies, these together with multispectral images can be employed for multisource remote sensing image analysis tasks. With the point-based ground truth that can be used during training and validation, this data set provides a challenging test bed for fine-grained multisource studies.

- Fifth, to the best of our knowledge, we present the first deep neural network model that aims to learn deep representations of multisource remote sensing images, to correctly align them and to learn the classification rules in a unified framework simultaneously.

1.3 Outline

The rest of the thesis is organized as follows. Chapter 2 gives the details of how remote sensing studies handle the fine-grained object recognition problem with zero-shot learning and multisource scenarios. Chapter 3 introduces the data set used in this thesis. Since Chapter 4 describes the details of the zero-shot learning methodology and the related experiments when images are from only a single source. Chapter 5 provides the methodology to tackle the aligment problem when multiple sources are added. Chapter 6 provides the conclusion.

Chapter 2

Literature Review

Fine-grained object recognition differs from the traditional object recognition tasks predominantly studied in remote sensing literature in at least three main ways: (*i*) differentiating among many similar categories can be much more difficult due to low between-class variance, (*ii*) the difficulty of accumulating examples for a large number of similiar categories and rareness of some target classes can greatly limit the training set sizes, and, (*iii*) class imbalance when distribution of number of labels belonging to each class is different between training and test instances. Due to these major differences, the applicability of existing object recognition methods developed based on traditional data sets is unclear. Although there are data sets like UC Merced [11] and AID [2] having more than 20 classes, categories in these dataset (e.g. agricultural, airplane, buildings, forest, industrial, beach etc.) are relatively distinctive and balanced. Thus, the development of methods and benchmark datasets for fine-grained classification is an open research problem, whose importance is likely to increase over time.

However, there is limited number of studies in the literature dealing with directly fine-grained categories. In [12], for the fine-grained categories of street trees, two methods to produce a geographic catalog of objects with the help of multi-view aerial and street-level images of each location for object detection by using different viewpoints and zoom levels for object classification are proposed.

In [13], they combine their street trees detection and classification methods into a single framework and a change detection method is proposed additionally by classifying similarity of objects by using deep representation of images. Despite the fact that these studies can be regarded as the first attempts dealing with identification of subtle sub-categories at large scale on remote sensing literature, these do not directly propose a solution for limited training data.

Common attempts for reducing the effects of limited training data include generally both statistical solutions and active learning approaches. In [14], a regularized covariance estimator of each class in quadratic maximum-likelihood classifier in order to move the problem into a lower dimensional space without loss of information when the number of training instances are limited is proposed. For this, in order to benefit from the advantages of both leave-one-out covariance estimator (LOOC) and bayesian LOOC, linear combination of all mixture matrices used in these approaches is suggested as the estimator. For the feature extraction of small sample size scenario, [15] proposes a regularized within-class scatter matrix for linear discriminant analysis (LDA) and nonparametric weighted feature extraction by using the only diagonal parts and trace of the covariance matrix, and uses genetic algorithm in order to obtain mixing parameters of the within-class scatter matrix. To do so, the effect of the singularity problem is tried to handle. For the hyperspectral image classification with few annotated samples, the proposed approach in [16] gathers together rotation forest and multiclass AdaBoost algorithms as a classifier ensemble in order to decrease model bias and variance especially in the case of high dimensionality. Additionally, acquired posterior probabilities from AdaBoost are used as the unary potentials of the conditional random field (CRF) model to associate spatial contextual information with image classification.

However, significantly low between-class variance and high within-class variance in fine-grained recognition tasks limit the use of such statistical solutions. Another approach for tackling the insufficiency of annotated samples is to use active learning for interactively collecting new examples while enhancing the classification performance via manual labelling by interaction between domain expert and machine. For instance, in [17], with the help of two known active learning

methods supported by predefined heuristics that makes the simultaneous selection of several candidates at every iteration and multiclass classification possible, increasing the adaptability and speed of active learning methods in remote sensing image classification are studied. In [18], batch-mode active learning with a query function which selects the batch while considering the uncertainty for confidence of the supervised algorithm and diversity of samples to reduce the redundancy in remote sensing image classification is proposed. However, collecting examples for a very large number of very similar object categories in fine-grained recognition by using visual inspection of image data can be very difficult even for domain experts, as can be seen in the aerial-view examples in Figure 1.1.

Considering the more extreme and realistic scenario in which there is no annotated training instances of some classes, zero-shot learning aims to eliminate the bottleneck of limited training data. Recognition of these unseen classes in training phase requires new source of auxiliary information to carry out knowledge transfer between the unseen and seen classes. As one of the commonly used auxiliary information source, attributes can be understood by humans due to their semantic meaning [19] and used as class level information by machine.

Altough, the usage of attribute-based methods in computer vision literature is varied within the scope of image description [7], [6], caption generation [20], face recognition [21], image retrieval [22], action recognition [23], and object classification [24] in addition to zero-shot learning [25], to the best of our knowledge, there is no study using attributes for zero-shot learning task in remote sensing.

In addition to attributes, different source of information such as category hierarchy or text corpora can be used instead of manually annotated attributes. For these, as the only example in the remote sensing literature, the Word2Vec model [26] that was learned from text documents in Wikipedia was used for zero-shot scene classification by selecting some of the scene classes in the UC Merced data set as unseen categories [27]. However, categories in this study can be regarded as more distinctive and balanced considering the fine-grained tasks.

In addition to the difficulties of intertwined categories, class imbalance and

small sample size for rare objects, usage of single remote sensing sensor also limits the success of fine-grained recognition methods so that information gain from different sources improve the efficacy of those methods. Thus, how to use multiple remotely sensed images is very common research problem in remote sensing literature.

To exemplify, the Bayes rule for compound classification of images in addition to joint prior estimation with expectation maximization method [28], statistical modeling based fusion with dependence trees via estimation of probability distributions [29], kernel-based information fusion by bringing a group of nonlinear classifiers together for multitemporal image classification and change detection [30], copula-based statistical model of a multiresolution graph for the classifier development by estimating multivariate probability density functions with automatically generated multivariate copulas [31], feature-selection among spectral channels with sequential forward floating selection algorithm [32], active learning with ensemble multiple kernel depend on maximum disagreement query strategy [33], feature-level fusion of vegetation index, morphological building index, texture and connected component analysis by merging pixels having similiar pixel intensity values [34], a two-branch convolutional neural network by using both 1-D and 2-D kernels for feature-level fusion [35], feature stacking and graph-based feature fusion of extinction profiles of height, area, volume, diagonal of the bounding box, and standard deviation followed by a convolutional neural network [36], two convolutional neural networks for multispectral and LIDAR feature extraction followed by feature-level fusion [37], a decision-level fusion of a fully-convolutional neural network and a logistic regression as a linear classifier while combining them into a higher-order conditional random field (CRF) in which graph cut inference is applied for the estimation [38], multi-level ensemble of convolutional neural Network, random forest and gradient boosting machine classifiers via selection of prediction maps with the average entropy of the multinomial class distribution and posterior averaging [39] and a neural network model in which different convolutional branches are used for the image representations of different sources followed by a single convolutional fusion branch [40] are proposed for image classification.

Altough the research of the usage of multiple remote sensing sources mainly focus on image registration tasks, there have been very limited studies on other tasks which recevied more attention in the single source scnearios such as object detection, recognition etc. in the remote sensing literature. As an example, multi-scale sliding window with two-branch convolutional neural network to carry out late fusion [41] is suggested for object localization. Besides, a decision-level fusion of multiview contexual information [42] for object recognition are proposed. In this study, after object classification based on generating segmented regions properties with respect to spectral and textural characteristics together with the structural features (size, shape, and height of these regions) is carried out, general visibility map by adding all of the individual visibility maps together is created and the map is used with a higher level context aware approach for the decision-level fusion of the classified regions within the multiview perspective. For a detailed review of the classification methods for multisource remote sensing images, [43] gives the analysis of approaches in the basis of multimodality.

Despite the fact that benefitting from multiple sources gives an additional information about the object recognition task, using different sources can bring out new research problems to overcome. For instance, different remote sensing images may be obtained from diverse sources that can be on diverse conditions and can give different spatial and spectral resolution imagery. Thus, the geometrical alignment of resulting acquired images may not be appropriate due to seasonal changes, various viewing geometry, acquisition noise, imperfection of sensors, different atmospheric conditions etc. Furthermore, consistency of all ground truth labels in images from different sources is often not possible.

In order to tackle these problem, [44] benefitting from annotated samples from all remote sensing domains in order to gather them together to keep their manifolds more closer while trying to make the inherent structure of them unchanged with the help of proximity graphs created from unlabeled samples. Thus, semisupervised manifold alignment method of the study that contains the constraints of local manifold geometry into the alignment space is proposed for image registration. As an extension to the previous study, [45] imitates the similiarity of object categories with the help of weak labels which are obtained via common

objects (tie points) in the remote sensing images so that weakly supervised manifold alignment of different image domains can be applied. Additionally, their method widen the input domains for each different sources via adding features of Gaussian distances between samples and a center of interest domain with radial basis funciton kernel. In [46], nonlinear variant of semisupervised manifold alignment for remote sensing image registration problem via kernelization is proposed. In this study, kernelization is applied with the help of mapping the images into a Hilbert space having higher dimension with a mapping function so that semisupervised manifold alignment problem becomes a generalized eigenvalue problem and becomes more suited for high dimensional datasets.

In addition to manifold alignment approaches, [47] proposes a multiagent system with case-based reasoning (CBR) to simulate expert reasoning and rule-based reasoning (RBR) to support the CBR via a similarity measurement to model image imperfections such as imprecision and uncertainty. In [48], in order to identify registration noise among multitemporal and multisensor remote sensing images, a registration-noise estimation method with edge information assuming that generally object boundaries retains registration noise on themselves is proposed. The method of this study includes generating high edge magnitude pixels on images from mutiple domains and estimation of registration noise by determining the pixels which do not exist on border regions with the difference of Gaussian filters. [49] suggests a mid-level feature representation in which spatial distribution of image regions' spectral neighbors is encoded. Those representations with a Markov Random Field, in which altough edges in the same domain promote smoothness and matches of short distances, edges between different domains promote superpixels matching with similiar representations, are used for finding nonlinear mis-registrations and matches between remote sensing images from various sensors. Finally, [50] proposes a joint registration and change detection framework in which registration problem is handled with grid-based free-form deformation strategy associated with detection labels of an interpolation-based approach. Within the scope of a decomposed interconnected graphical model formulation, a Markov Random Field over change detection and registration graphs enables that the relaxation of registration similarity constraints is carried out in

the presence of change detection. The study benefits from linear programming and duality concepts so as to optimize a joint solution space.

However, interest in object recognition has been very limited where existing studies have generally focused an image classification with a small number of classes with distinct differences. Thus, the applicability of such methods to fine-grained classification is not clear. Additionally, although using deep neural networks has been provided a significant contribution to remote sensing literature, to the best of our knowledge, there has not been any study trying to overcome aforementioned problems of multi-domain remote sensing images by benefitting deep representations of images and deep neural network models proposed to handle those problems.

Chapter 3

Data Set

There was no publicly available remote sensing data set that contains a large number of classes with high within-class and low between-class variance. Thus, we created a new data set [1] that provides a challenging test bed for fine-grained object recognition research. We have gathered the data set from three main sources.

The first part corresponds to point GIS data for street trees provided by the Seattle Department of Transportation in Washington State, USA [51]. In addition to location information in terms of latitude and longitude, the GIS data contain the scientific name and the common name for each tree. The second part was obtained from the Washington State Geospatial Data Archive's Puget Sound orthophotography collection [52]. This part corresponds to 1 foot spatial resolution aerial RGB images that we mosaiced over the area covered by the GIS data. Among the total of 126,149 samples provided for 674 tree categories, we chose the top 40 categories that contain the highest number of instances. We also carefully went through every single one of the samples, and made sure that the provided coordinate actually coincides with a tree. Some samples had to be removed during this process due to mismatches with the aerial data, probably

[1] Available with RGB image patches and point GIS data at `http://www.cs.bilkent.edu.tr/~saksoy/publications.html`.

because of seasonal and temporal differences between ground truth collection and aerial data acquisition. Finally, each tree is represented as a 25×25 pixel patch that is centered at the point ground truth coordinate where the patch size was chosen as 25 to cover the largest tree. Overall, the resulting data set contains a total of 48,063 trees from 40 different categories.

In addition to the 1 foot spatial resolution aerial RGB images, we also use 1.84 meter spatial resolution WorldView-2 satellite multispectral images (WorldView-2 © 2011, DigitalGlobe, Inc.) having 8 spectral bands for multisource experiments. RGB images are taken as reference images correctly aligned with ground truth labels since correspondence of every single one of the samples with labels was checked. Multispectral images are regarded as target images that need to be aligned with correct spatial object regions since registration of them with corresponding RGB images is improper. Thus, although the corresponding patch size is 4×4 pixel considering the relative spatial resolution of the aerial and multispectral data, 12×12 pixel patch, that is centered at the point ground truth coordinate, is used assuming that correct spatial region is located somewhere in this window.

The list of the tree categories along with the number of instances in each category is given in Table 4.2. We use different splits of this imbalanced data set for a fair and objective evaluation of fine-grained object recognition with zero-shot learning as suggested in [10] and one of the splits are used for multisource fine-gained object recognition with our weight estimation framework that are presented in Section 4.5 and Section 5.4. Figure 1.1 illustrates the RGB and ground-view images of 16 tree categories that are used as the unseen classes for the zero-shot learning experiments.

Chapter 4

Single Source Fine-Grained Object Recognition

In this chapter, the mathematical formulation of proposed zero-shot learning (ZSL) approach and the image and class representations utilized for describing the aerial objects and fine-grained object classes will be described. At the end of the chapter, detailed experimental analysis of our approach will be presented. Parts of this chapter was previously published as [53].

4.1 Zero-shot Learning Model

The goal is to learn a discriminator function that maps a given image $x \in \mathcal{X}$ to one of the target classes $y \in \mathcal{Y}$ where $\mathcal{X}$ is the space of all images and $\mathcal{Y}$ is the set of all object classes. By definition of zero-shot learning, training examples are available only for a subset of the classes, $\mathcal{Y}_{\text{tr}} \subset \mathcal{Y}$, which are called the *seen classes*. Therefore, it is not possible to directly use traditional supervised methods, like decision trees, to build a model that can recognize the *unseen classes*, $\mathcal{Y}_{\text{te}} \subset \mathcal{Y}$, i.e., those with no training samples, when $\mathcal{Y}_{\text{tr}} \cap \mathcal{Y}_{\text{te}} = \emptyset$.

To overcome this difficulty, it is firstly assume that a vector-space representation, called *class embedding*, is available for each class. Each class embedding vector is expected to depict (visual) characteristics of the class such that classification knowledge can be transferred from seen to unseen classes.

To carry out this knowledge transfer, we utilize a compatibility function $F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$, which is a mapping from a given image-class pair (x,y) to a scalar value. This value represents the confidence in assigning the image x to class y.

Since examples only from the seen classes are available for learning the compatibility function, which will be utilized for recognizing instances of the unseen classes, $F(x, y)$ should employ a class-agnostic model. For this purpose, following the recent work on ZSL [10], we define the compatibility function in a bilinear form, as follows:

$$F(x, y) = \phi(x)^\top W \psi(y). \tag{4.1}$$

In this equation, $\phi(x)$ is a d-dimensional image representation, called image embedding, $\psi(y)$ is an m-dimensional class embedding vector, and W is a $d \times m$ matrix. This compatibility function can be considered as a class-agnostic model of a cross-domain relationship between the image representations and class embeddings. See Figure 4.1 for an illustration of the compatibility function. The formulation itself is also not specific to any type image representation or class embedding.

A number of empirical loss minimization schemes have been proposed for learning such ZSL compatibility functions in recent years. A detailed evaluation of these schemes can be found in [10]. In our preliminary experiments, we have investigated the state-of-the-art approaches of [9] and [4], and observed that an intuitive alternative formulation based on an adaptation of multi-class logistic regression classifier yields comparable to or better results than the others. In our approach, we define the class posterior probability distribution as the *softmax* of compatibility scores:

$$p(y|x) = \frac{\exp\left(F(x, y)\right)}{\sum_{y' \in \mathcal{Y}_{\text{tr}}} \exp\left(F(x, y')\right)} \tag{4.2}$$

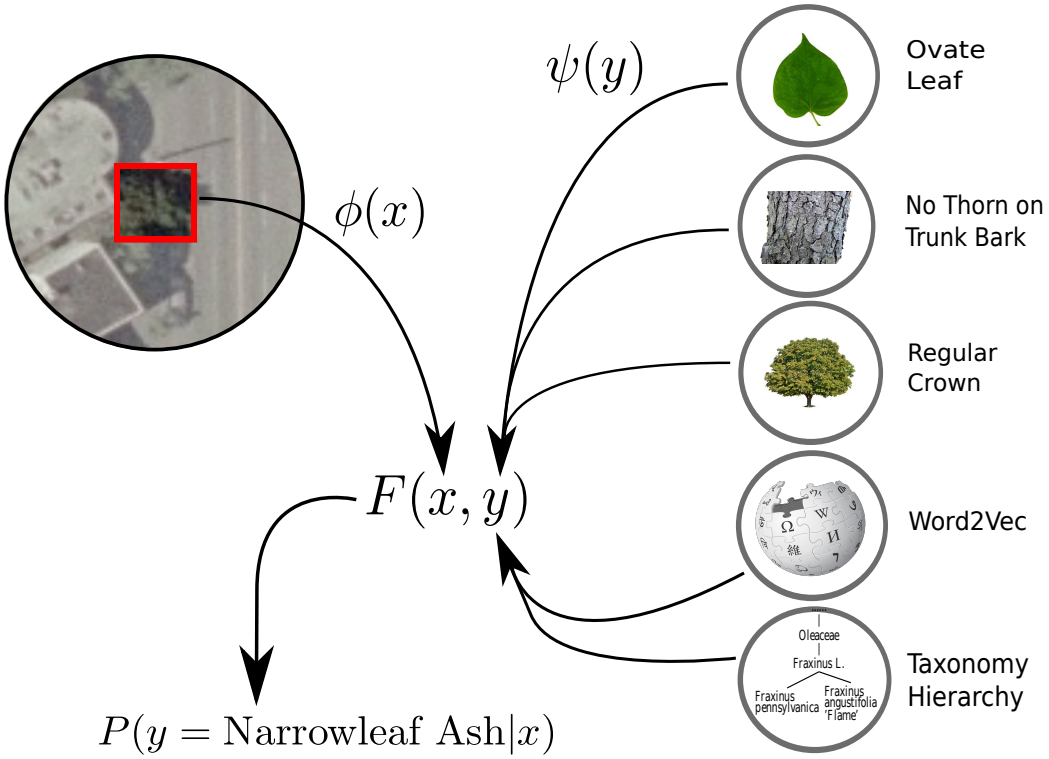

Figure 4.1: Our proposed framework learns the compatibility function $F(x, y)$ between image embedding $\phi(x)$ and class embeddings $\psi(y)$ based on attributes, word-embeddings from a natural language model, and a hierarchical scientific taxonomy. The learned compatibility function is then used in recognizing instances of unseen classes by leveraging their class embedding vectors.

where $\mathcal{Y}_{\text{tr}} \subset \mathcal{Y}$ is the set of seen (training) classes. Then, given N_{tr} training examples, we aim to learn $F(x, y)$ using the maximum likelihood principle. Assuming that the data set contains independent and identically distributed samples, the label likelihood is given by

$$\underset{W \in \mathbb{R}^{d \times m}}{\text{maximize}} \prod_{i=1}^{N_{\text{tr}}} p(y_i|x_i). \tag{4.3}$$

The optimization problem can be interpreted as finding the W matrix that maximizes the predicted true class probabilities of training examples, on average. Equivalently, the parameters can be found by minimizing the negative log-likelihood:

$$\underset{W \in \mathbb{R}^{d \times m}}{\text{minimize}} \sum_{i=1}^{N_{\text{tr}}} -\log p(y_i|x_i). \tag{4.4}$$

To find a local optimum solution, we use stochastic gradient descent (SGD) based optimization. The main idea in SGD is to iteratively sample a batch of training examples, compute approximate gradient over the batch, and update the model parameters using the approximate gradient. In our case, at SGD iteration t, the gradient matrix G_t over a batch B_t of training examples can be computed as follows:

$$G_t = -\sum_{i \in B_t} \nabla_W \log p(y_i|x_i)$$

where the gradient of the log-likelihood term for the i-th sample is given by

$$\nabla_W \log p(y_i|x_i) = \phi(x_i)\psi(y_i)^\top - \sum_{y \in \mathcal{Y}_{\text{tr}}} p(y|x_i)\phi(x_i)\psi(y)^\top.$$

Given the approximate gradient, the plain SGD algorithm works by subtracting a matrix proportional to G_t, from the model parameters:

$$W_t \leftarrow W_{t-1} - \alpha G_t \tag{4.5}$$

where W_t denotes the updated model parameters, and the *learning rate* α determines the rate of updates over the SGD iterations. It is often observed that the learning rate needs to be tuned carefully in order to avoid too large or too small parameter updates, which is necessary to maintain a stable and steady progress

over the iterations. However, not only finding the right learning rate is an uneasy task, but also the optimal rate may vary across dimensions and over the iterations [54].

In order to minimize the manual effort for finding a well-performing learning rate policy, we resort to adaptive learning rate techniques which are provided by recent progress in stochastic optimization techniques. In particular, we utilize the *Adam* technique [55], which estimates the learning rate for each model parameter based on the first and second moment estimates of the gradient matrix. For this purpose, we calculate the running averages of the moments at each iteration:

$$M_t = \beta_1 M_{t-1} + (1 - \beta_1) G_t$$
$$V_t = \beta_2 V_{t-1} + (1 - \beta_2) G_t^2$$

where M_t and V_t are the first and second moment estimates, β_1 and β_2 are the corresponding exponential decay rates, and G_t^2 is the element-wise square of G_t. Then, the SGD update step is modified as follows:

$$W_t \leftarrow W_{t-1} - \alpha \ \hat{M}_t / (\sqrt{\hat{V}_t} + \epsilon)$$

where $\hat{M}_t = M_t/(1 - \beta_1^t)$ and $\hat{V}_t = V_t/(1 - \beta_2^t)$ are the bias-corrected first and second moment estimates. These estimates remove the inherent bias towards zero due to zero-initialization of M_t and V_t at $t = 0$, which is particularly important in early iterations. Overall, $\hat{M}_t$ provides a momentum-based approximation to the true gradient based on the approximate gradients over batches, and $\hat{V}_t$ provides a per-dimension learning rate adaptation based on an approximation to diagonal Fisher information matrix.

Finally, we should also note that we do not use an explicit regularization term on W in our training formulation. Instead, we use *early stopping* as a regularizer. For this, we track the performance of the ZSL model on an independent validation set over optimization steps, and choose the best performing iteration. Additional details are provided in Section 4.5.

Once the compatibility function (i.e., the W matrix) is learned, zero-shot recognition of unseen test classes is achieved by assigning the input image to the class

y^* whose vector-space embedding yields the highest score as

$$y^* = \underset{y \in \mathcal{Y}_{\text{te}}}{\operatorname{argmax}} F(x, y). \tag{4.6}$$

In the next two sections, we explain the details of our image representation and class embeddings, which have central importance in ZSL performance.

4.2 Image Embedding

We employ a deep convolutional neural network (CNN) to learn and extract region representations for aerial images. The motivation for using a CNN is to be able to exploit both the pixel-based spectral information and the spatial texture content. Spectral information available in the three visible bands is not expected to be sufficiently discriminative for fine-grained object recognition, and the learned texture representations are empirically found to be superior to hand-crafted filters.

For this purpose, based on our preliminary experiments using only the 18 seen classes from our data set, we have developed an architecture that contains three convolutional layers with 5×5, 5×5, and 3×3 dimensional filters, respectively, and two fully-connected layers that map the output of the last convolutional layer to the 18 different class scores. In designing our CNN architecture, we have aimed to use filters that are large-enough for learning patterns of tree textures and shapes. We use a stride of 1 in all convolutional layers to avoid information loss, and keep the spatial dimensionality over convolutional layers via zero-padding. While choosing the number of filters (64 filters per convolutional layer), we have aimed to strike the right balance between having sufficient model capacity and avoiding overfitting. We use max-pooling layers to achieve partial translation invariance [56]. Finally, we have also investigated a number of similar deeper and wider architectures, yet obtained the best performance with the presented network. Additional details of the architecture can be found in Figure 4.2. While Figure 4.2 shows an input with 3 channels, the architecture can easily be adapted

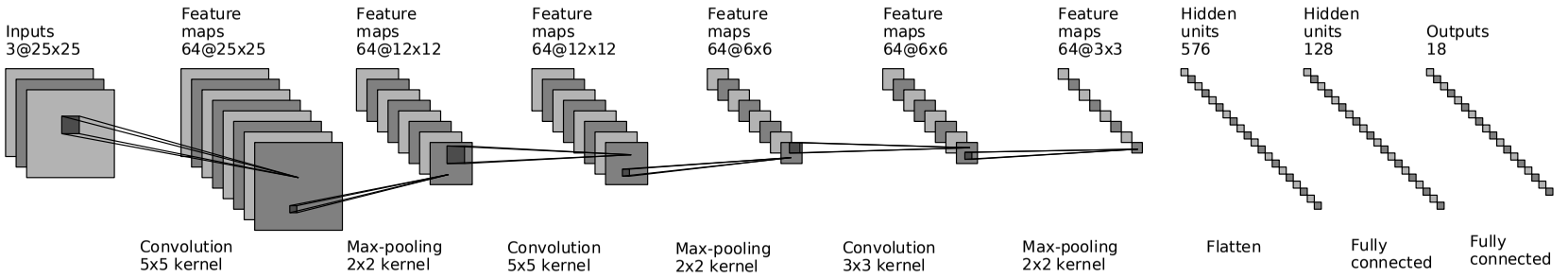

Figure 4.2: Proposed deep convolutional neural network architecture with three convolutional layers containing 64 filters each with sizes 5×5, 5×5, and 3×3, respectively, followed by two fully-connected layers containing 128 and 18 neurons, respectively. We apply max-pooling after each convolutional layer. The feature map sizes are stated at the top of each layer.

to any number of input spectral bands. In general, for an input with B bands, one can simply use kernels of shape $5\times5\times B$ in the first layer.

We train the CNN model over the seen classes using cross-entropy loss, which corresponds to maximizing the label log-likelihood in the training set. To improve training, we employ Dropout regularization [57] (with 0.9 keep probability) and Batch Normalization [58] throughout the network, excluding the last layer. Once the network is trained, we use the output of the first fully connected layer, i.e., the 128-dimensional vector shown in Figure 4.2, as our image embedding $\phi(x)$ for the ZSL model. We additionally ℓ_2-normalize this vector, which is a common practice for CNN-based descriptors [59].

Finally, we note that one can consider pre-training the CNN model on external large-scale data sets like ImageNet and fine-tuning it to the target problem. While such an approach is likely to improve the recognition accuracy, it may also lead to biased results due to potential overlaps between the classes in our ZSL test set and the classes in the data set used during pre-training that will violate the zero-shot assumption and will hinder the objectiveness of the performance evaluation [10]. Therefore, we opt to train the CNN model solely using our own training data set.

Additional CNN training details and an empirical comparison of our CNN model to other contemporary classifiers are provided in Section 4.5.

4.3 Class Embedding

Class embeddings are the source of information for transferring knowledge that is relevant to classification from seen to unseen classes. Therefore, the embeddings need to capture the visual characteristics of the classes. For this purpose, following the recent work on using multiple embeddings in computer vision problems [9], we use a combination of three different class embedding methods: (i) manually annotated attributes that we collect from the target domain, (ii) text

Table 4.1: Attributes for fine-grained tree categories

Attribute type	Possible values
Height (feet)	{10-15, 15-20, 20-25, 25-30, 30-40, 40-50, 50-60, 60-75}
Spread (feet)	{10-15, 15-25, 25-35, 35-40, 40-50}
Crown uniformity	{irregular outline, regular outline}
Crown density	{open, moderate, dense}
Growth rate	{medium, fast}
Texture	{coarse, medium, fine}
Leaf arrangement	{opposite/subopposite, alternate}
Leaf shape	{ovate, star-shaped}
Leaf venation	{palmate, pinnate}
Leaf blade length	{0-2, 2-4, 4-8}
Leaf color	{green, purple}
Fall color	{green, yellow, purple, red, orange}
Fall characteristics	{not showy, showy}
Flower color	{brown, pink, green, red, white, yellow}
Flower characteristics	{not showy, showy}
Fruit shape	{round, elongated}
Fruit length	{0-0.25, 0.25-0.50, 0.5-1.5, 1-3}
Fruit covering	{dry-hard, fleshy}
Fruit color	{brown, purple, green, red}
Fruit characteristics	{not showy, showy}
Trunk bark branches	{no thorns, thorns}
Pruning requirement	{little, moderate}
Breakage	{not resistant, resistant}
Light requirement	{not part sun, part sun}
Drought tolerance	{moderate, high}

embeddings generated using unsupervised language models, and, (iii) a hierarchical embedding based on a scientific taxonomy.

Visual attributes are obtained by determining visually distinctive features of objects, such as their parts, textures, and shapes. Since they provide a high-level description of object categories and their fine-grained properties, as perceived by humans, attributes stand out as an outstanding class embedding method for zero-shot learning [8]. In order to utilize attributes in our work, we have collected 25 attributes for tree species, based on the Florida Trees Fact-Sheet [60]. We list the names and possible values of these attributes in Table 4.1. These values are

encoded as binary variables in a vector.

Although attributes provide powerful class embeddings, they are typically not comprehensive in capturing characteristics of object categories, since attributes are defined in a manual way based on domain expertise. Our second method that complements attributes is based on unsupervised word embedding models trained over large textual corpora. For this purpose, we utilize the Word2Vec approach [26], which models the relationship between words and their contexts. Since closely related words usually appear in similar contexts, the resulting word vectors are known to implicitly encode semantic relationships. That is, words with similar meanings typically correspond to nearby locations in the embedding space. Our main goal here is to leverage the semantic relationships encoded by Word2Vec to help the ZSL model in inferring models of unseen classes. For this purpose, we use a 1000-dimensional embedding model trained on Wikipedia articles, and extract word embeddings of common names of tree species (given in Table 4.2). For categories with multiple words, we take the average of the per-word embedding vectors.

The third and the last type of class embedding that we use aims to capture the similarities across tree species based on their scientific classification. The scientific taxonomy of species in our data set is presented in Figure 4.3. Since the genetics of tree species directly affect their phenotype, the taxonomic positions of trees can be informative about the visual similarity across the species. In order to capture the position and ancestors of tree species in the taxonomy tree, we apply the tree-to-vector conversion scheme described in [62]. The embedding vector corresponding to a given tree species is obtained by defining a binary value for each node in the taxonomy tree, and turning on only the values that correspond to the nodes that appear on the path from the root to the leaf node of interest. As a result, we obtain an embedding vector of length equivalent to the number of nodes in the taxonomy.

We form the final embedding vector by concatenating the vectors produced by these three embedding methods.

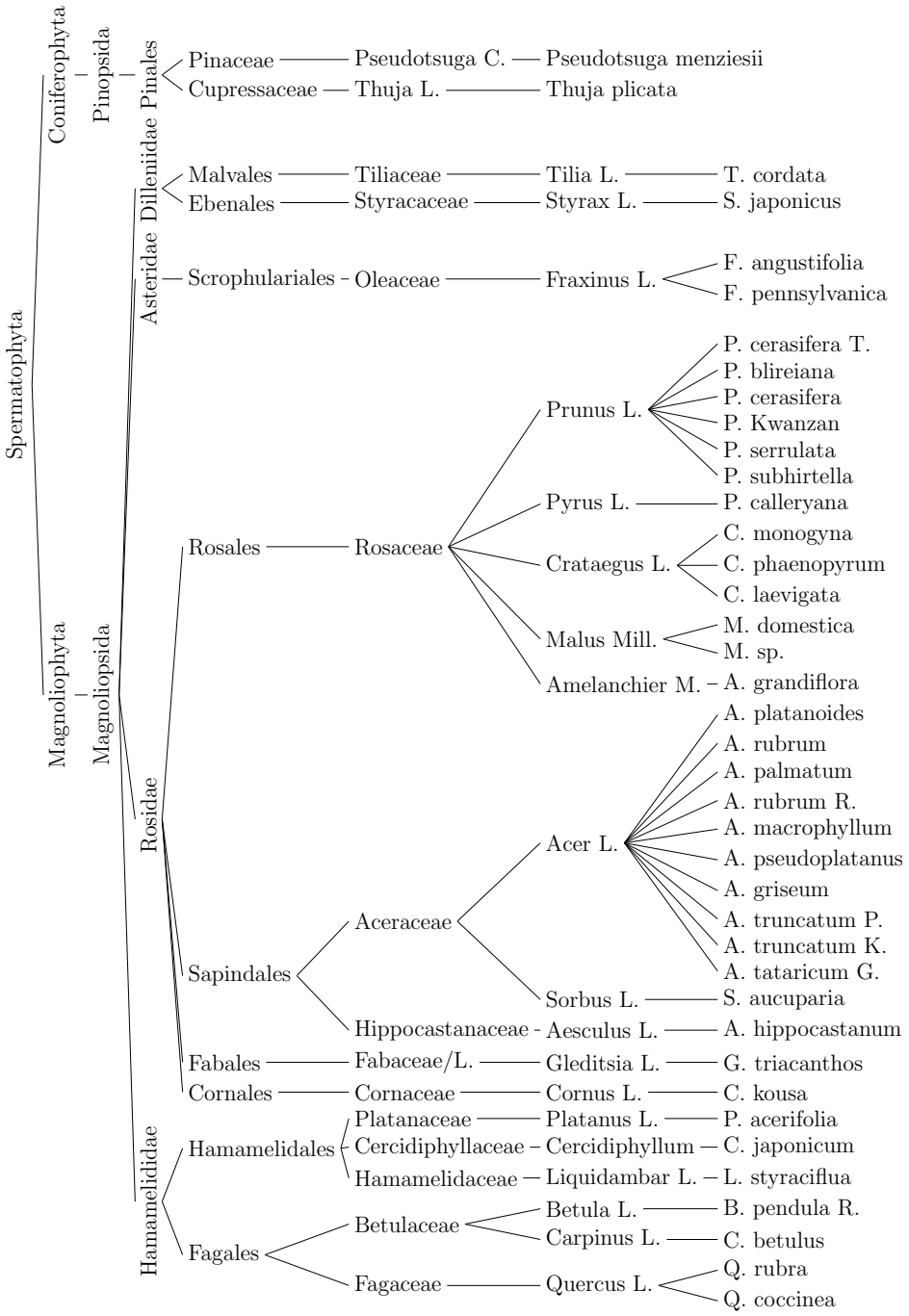

Figure 4.3: Hierarchy embeddings are based on scientific classification of tree species. This part of plant classification represents taxonomy of our tree classes that starts with Spermatophyta superdivision and continues with the names of division, class, subclass, order, family, genus, and species in order. At each level, scientific names are written instead of common names. Classification of each tree is taken from the Natural Resources Conservation Service of the United States Department of Agriculture [61].

4.4 Joint Bilinear and Linear Model

The bilinear model specified in (4.1) can be interpreted as learning a weighted sum over all products of input and class embedding pairs. That is, the compatibility function can be equivalently written in the following way:

$$F(x, y) = \sum_{u=1}^{d} \sum_{v=1}^{m} W_{uv} [\phi(x)]_u [\psi(y)]_v \tag{4.7}$$

where $[\phi(x)]_u$ and $[\psi(y)]_v$ denote the u-th and v-th dimensions of input and class embeddings, respectively. From this interpretation we can see that the approach can learn relations between input and class embeddings, but may not be able to evaluate the information provided by them individually. To address this shortcoming, we propose to extend the bilinear model by adding embedding-specific linear terms:

$$F_e(x, y) = \phi(x)^\top W \psi(y) + w_x^\top \phi(x) + w_y^\top \psi(y) + b \tag{4.8}$$

where F_e is the *extended* compatibility function, w_x is the linear model over the input embeddings, w_y is the linear model over the class embeddings, and b is a bias term.

The advantage of having input and class embedding specific linear terms can be understood via the following examples: using the term $w_x^\top \phi(x)$, the model may adjust the entropy of the posterior probability distribution, i.e., the confidence in predicting a particular class, by increasing or decreasing all class scores depending on the clarity of object characteristics in the image. Similarly, using the term $w_y^\top \psi(y)$, the model can estimate a class prior based on its embedding. Therefore, these extensions are likely to improve the recognition model. Finally, we note that the bias term has no effect on the estimated class posteriors given by (4.2), yet it simplifies the derivation below.

We incorporate the linear terms of the model in a practical way by simply

adding constant dimensions to both the input embedding and the class embedding. More specifically, we extend the input and class embeddings as follows:

$$\phi_e(x) = [\phi(x)^\top \ 1]^\top \tag{4.9}$$

$$\psi_e(y) = [\psi(y)^\top \ 1]^\top \tag{4.10}$$

where $\phi_e(x)$ and $\psi_e(y)$ denote the extended embedding vectors. Similarly, we define the extended compatibility matrix W_e as:

$$W_e = \begin{bmatrix} W & w_x \\ w_y & b \end{bmatrix}. \tag{4.11}$$

It is easy to show that the bi-linear product $\phi_e(x)^\top W_e \psi_e(y)$ is equivalent to the extended compatibility function $F_e(x, y)$, given by (4.8). Therefore, the linear terms can simply be introduced by adding bias dimensions to the embeddings.

4.5 Experiments

In this section, we first describe our experimental setup for the single source scenario of zero-shot learning and fully-supervised object recognition. We then present an evaluation of our CNN model in a supervised classification setting, followed by the evaluation of our zero-shot learning approach. Finally, we experimentally analyze our model, compare it to important baselines, and discuss our findings.

4.5.1 Experimental Setup

In our experiments, we need to train and evaluate our approach in supervised and zero-shot learning settings. Therefore, in order to obtain unbiased evaluation results, we need to define a principled way for tuning the model hyper-parameters. This is particularly important in zero-shot learning because of the expectation that the separation between the seen and unseen classes is clear. We follow the guidance given in [10]: (i) ZSL should be evaluated mainly on least populated

classes as it is hard to obtain labeled data for fine-grained classes of rare objects, (ii) hyper-parameters must be tuned on a validation class split that is different training and test classes, and (iii) extracting image features via a pre-trained deep neural network on a large data set should not involve zero-shot classes for training the network.

Following these guidelines, we split the 40 classes from our Seattle Trees data set into three disjoint sets (with no class overlap): 18 classes as the *supervised-set*, 6 classes as the *ZSL-validation* set, and the remaining 16 classes as the *ZSL-test* set. The list of classes in each split is shown in Table 4.2. We have arranged the splits roughly based on the number of examples in each class: we mostly allocated the largest classes to the supervised-set, the smallest classes to ZSL-validation, and the remaining ones to ZSL-test to have a reliable performance for ZSL accuracy evaluation. Additionally, putting classes, which have closer positions in scientific taxonomy, into different class splits is also considered when we arranged the splits for the reliable performance evaluation.

considered that having closer position in scientific taxonomy tree

We use the supervised-set for two purposes: (i) to evaluate the CNN model in a supervised classification setting, and (ii) to train the ZSL model using the supervised classes. For the supervised classification experiments, we use only the classes inside the supervised-set, and we split the images belonging to these classes into *supervised-train* (60%), *supervised-validation* (20%) and *supervised-test* (20%) subsets. We emphasize that these three subsets contain images belonging to the 18 supervised-set classes, and they do not contain any images belonging to a class from the ZSL-validation set or the ZSL-test set. We aim to maximize the performance on the supervised-validation set when choosing the hyper-parameters of the supervised classifiers.

In ZSL experiments, we train the ZSL model using all images from the supervised-set. We use the zero-shot recognition accuracy in the ZSL-validation set for tuning the hyper-parameters of the ZSL model. We evaluate the final model on the ZSL-test set, which contains the unseen classes. In this manner, we

Table 4.2: Class separation used for the data set and the number of instances

Supervised-set	ZSL-validation	ZSL-test
Flame Ash (679)	Flame Amur Maple (242)	Japanese Snowbell (460)
Western Red Cedar (720)	Norwegian Maple (372)	Scarlet Oak (489)
Honey Locust (875)	Katsura (383)	Apple Serviceberry (552)
Bigleaf Maple (885)	Paperbark Maple (467)	Orchard Apple (583)
Red Maple (1086)	Washington Hawthorn (503)	Douglas Fir (620)
Japanese Maple (1196)	Chinese Cherry (1531)	Autumn Cherry (621)
Red Oak (1429)		Kousa Dogwood (642)
Apple/Crabapple (1624)		Green Ash (660)
Littleleaf Linden (1626)		Mountain Ash (672)
White Birch (1796)		Pacific Maple (716)
Kwanzan Cherry (2398)		Sycamore Maple (742)
Thundercloud Plum (2430)		European Hornbeam (745)
Sweetgum (2435)		Common Hawthorn (809)
Blireiana Plum (2464)		Horse Chestnut (818)
Cherry Plum (2510)		Callery Pear (892)
Red Maple (2790)		London Plane (1477)
Norway Maple (2970)		
Midland Hawthorn (3154)		

avoid using unseen classes during training or model selection, which, we believe, is fundamentally important for properly evaluating the ZSL models.

Throughout our experiments, we use normalized accuracy as the performance metric, which we obtain by averaging per-class accuracy ratios. In this manner, we aim to avoid biases towards classes with a large number of examples.

4.5.2 Supervised Fine-grained Classification

Before presenting our ZSL results, we first evaluate our CNN model in a supervised-setting to compare it against other mainstream supervised classification techniques, and to give a sense of the difficulty of the fine-grained classification problem that we propose. For this purpose, we use logistic regression and random forest classifiers as our baselines. For a fair comparison, we train all methods on the supervised-train set, and tune their hyper-parameters on the supervised-validation set.

We train our CNN architecture using stochastic gradient descent with the Adam method [55] that we also use for ZSL model estimation as described in Section 4.1. Based on the supervised-validation set, we have set the initial learning rate of Adam to 10^{-3}, mini-batch size to 100, and ℓ_2-regularization weight to 10^{-5}. We also observed that it is beneficial to add perturbations of training

Table 4.3: Supervised classification results (in %)

	Random guess	Logistic regression	Random forest	CNN	CNN with perturbation
Normalized accuracy	5.6	16.4	15.7	27.9	34.6

examples by randomly shifting each region with an amount in the range from zero to 20% of the height/width.

We compare the resulting classifiers on the supervised-test set, as shown in Table 4.3. From these results we can see that all classification methods perform clearly better than the random guess baseline (5.6%). In addition, we can see that the proposed CNN model both without perturbation (27.9%) and with perturbation (34.6%) outperforms logistic regression (16.4%) and random forest (15.7%) by a large margin.

These results highlight the advantage of the deep image representation learned by the CNN approach. In addition, we can observe the difficulty of the fine-grained classification problem, which is quite different from the traditional classification scenarios that aim to discriminate buildings from trees or roads from grass. We believe that fine-grained classification is an important open problem in remote sensing, and can lead to advances in object recognition research.

4.5.3 Fine-grained Zero-shot Learning

In this part, we evaluate our ZSL approach on only RGB images and compare against three state-of-the-art ZSL methods: ALE [19], SJE [9], and, ESZSL [4]. We train all ZSL models over the supervised-train set, and tune all model hyper-parameters according to normalized accuracy on the ZSL-validation set.

For our approach, we initialize the W matrix randomly from a uniform distribution [63] and train the model using Adam optimizer [55]. We tune the

Table 4.4: Zero-shot learning results (in %)

	Random guess	ALE [19]	SJE [9]	ESZSL [4]	Ours
Normalized accuracy	6.3	12.5	12.6	13.2	14.3

hyper-parameters of initial learning rate of Adam and the number of training iterations (for early-stopping based regularization). For the ALE [19] and SJE [9] baselines, we use stochastic gradient descent (SGD) for training. Unlike the original papers that use a constant learning rate for SGD, we have found that decreasing the learning rate regularly over epochs leads to better performance for these baselines. We tune the the learning rate policy on the ZSL-validation set. For the ESZSL [4] baseline, we tune its regularization parameters λ and γ by choosing the best-performing combination of the parameters in the range $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\}$ according to the ZSL-validation set and fix the β hyper-parameter to $\lambda\gamma$, as suggested in [4]. In this case, the optimal compatibility matrix is given by a closed-form solution [4]. Finally, we note that all compared methods learn a single compatibility W matrix, which provides a fair comparison across them.

For all methods, we have observed that imbalance in terms of the number of examples across the training classes can negatively affect the resulting ZSL model. To alleviate this problem, we apply random over-sampling to the training set such that the size of the training set for each class is equivalent to the size of the largest class.

The ZSL results over the 16 ZSL-test classes are presented in Table 4.4. Our ZSL model achieves a 14.3% normalized accuracy, which is clearly better than the random guess baseline (6.3%), ALE (12.5%), SJE (12.6%), and ESZSL (13.2%). These results validate the effectiveness of our probabilistic ZSL formulation.

The image embedding can have a profound effect on the ZSL performance. To better understand the efficacy of our representation, we train our ZSL model

over the outputs of different CNN layers (Figure 4.2), and tune the number of training iterations on ZSL-validation for each one separately. When we use the 18-dimensional classification outputs, the ZSL performance drops from 14.3% to 8.5%. Similarly, if we use the outputs of the layers preceding the first fully-connected layer, the performance drops from 14.3 to 13.0% for last max-pooling output (equivalently, the *Flatten* output), to 12.8% for the convolutional layer with 3×3 kernels, and to 11.1% for the preceding max-pooling output. Simply using the original RGB image results in 8.3% ZSL performance. Overall, these results highlight the importance of the image representation on the ZSL performance, and suggest that the fully-connected layer preceding the classification layer results in relatively generic features that are suitable for ZSL, in our architecture.

Our class embedding is a combination of three different embedding techniques. To understand the contribution of each one, we present the ZSL performance for each possible combination of class embedding methods in Table 4.5. The first three rows of the table indicate that when the embedding techniques are used individually, they result in a comparable performance, with a higher performance for the Word2Vec (12.1%), compared to attributes (8.4%) and hierarchy (9.7%). The following three rows indicate that the hierarchy-Word2Vec (13.2%) embedding pair leads to better results compared to individual embeddings as well as the pairs of attribute-Word2Vec (12.6%) or hierarchy-attribute (11.2%). These results show that our hierarchy and Word2Vec embeddings are more effective than attribute embeddings. This observation suggests that the recognition accuracy can be improved possibly by defining more descriptive attributes. On the other hand, the final result based on the combination of all embeddings, which leads to the highest accuracy (14.3%), shows that our class embeddings are complementary to each other.

Another important aspect of the proposed method is extending the bilinear model by adding linear terms for the input and class embeddings. To understand the significance of this extension, we present an evaluation of the linear terms in Table 4.6. The table shows that without having any linear term, the normalized accuracy for ZSL is 11.8%. Adding $w_x^\top \phi(x)$, see (4.8), improves the performance

Table 4.5: Effect of different class embeddings on zero-shot learning performance (in %)

Attribute	Hierarchy	Word2Vec	Normalized accuracy
✓	✗	✗	8.4
✗	✓	✗	9.7
✗	✗	✓	12.1
✓	✓	✗	11.2
✓	✗	✓	12.6
✗	✓	✓	13.2
✓	✓	✓	**14.3**

to 12.2%, and adding $w_y^\top \psi(y)$ improves the performance to 13.4%. Finally, adding both terms together leads to our highest result of 14.3% normalized accuracy. These results validate the importance of adding linear terms into the bilinear ZSL model.

4.5.4 Discussion

The results presented so far show that the proposed ZSL approach performs significantly better than the random guess baseline, and also better than several other state-of-the-art ZSL methods. However, an important question is how well ZSL performs in a practical sense. To address this question, we compare our ZSL approach against supervised classification, and discuss the relative advantages and disadvantages of supervised versus zero-shot learning of novel class models.

For this purpose, we use five-fold cross validation over the whole ZSL-test set, where repeatedly one of the folds is utilized for training the supervised classifiers, and the remaining folds are utilized as the test subset. In our analysis, we consider two types of supervised classifiers: (i) CNN models that are trained from scratch over supervised ZSL-test examples only, (ii) pre-trained CNN models that

Table 4.6: Effect of linear terms on zero-shot performance (in %)

For image embedding	For class embedding	Normalized accuracy
✗	✗	11.8
✓	✗	12.2
✗	✓	13.4
✓	✓	**14.3**

are *fine-tuned* to the ZSL-test classes. For the latter approach, we re-initialize and re-train the last layer, i.e., the classification layer, of our pre-trained CNN model. Our motivation for fine-tuning is that all layers preceding the last layer are likely to extract a class-agnostic image representation, and the last layer can be interpreted as a linear classifier that transforms the learned image representation into the classification scores. In this way, we can effectively transfer knowledge from supervised-set to ZSL-test, using supervised training examples for the latter set.

It is well-known that the accuracy of a supervised classifier tends to improve as its training set gets larger. In this context, to understand the trade-off between using a ZSL approach, which uses zero training examples for the target classes, versus collecting supervised training examples, we train the supervised classifiers at varying number of training examples. More specifically, we train separate supervised classifiers by limiting the number of examples *per class* to each possible constant in $1, 2, 4, \ldots, 2^{10}$. We impose these limits by subsampling the training subset at each fold of five-fold cross-validation. To obtain reliable statistics, we repeat each experiment 10 times.

Figure 4.4 presents the results for the supervised-only CNN, fine-tuned CNN, and the ZSL model. The x-axis shows the number of training examples for the supervised classifiers, and the y-axis shows the corresponding normalized accuracy scores. The curves are obtained by averaging results over all folds and all runs, and setting the curve thickness to the standard deviation of the results. The ZSL

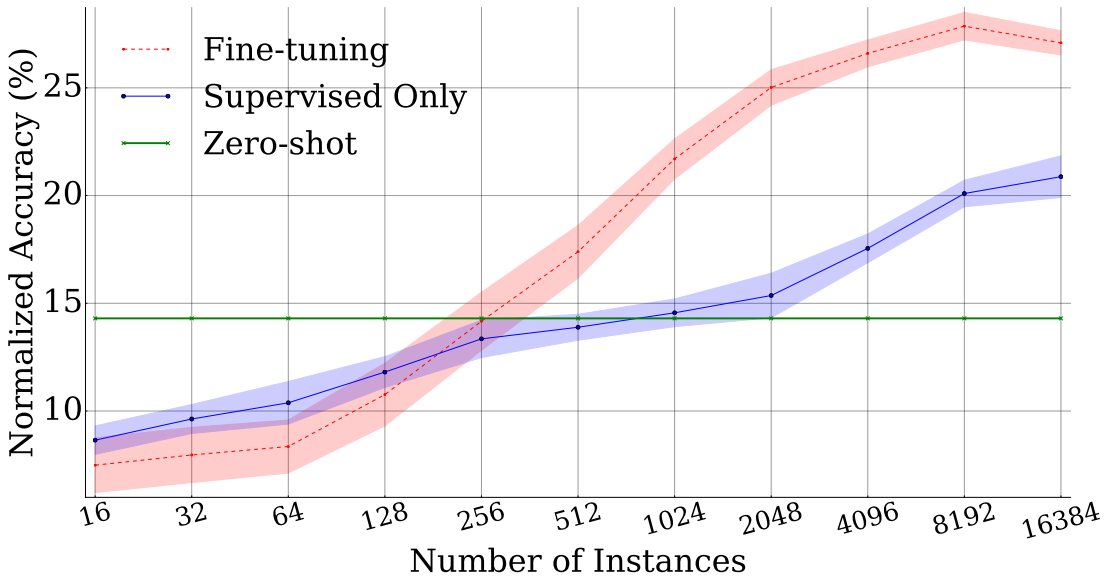

Figure 4.4: Performance comparison of the proposed framework with fine-tuning and supervised-only methods on zero-shot test classes. Fine-tuning and supervised-only results are derived at different points when the number of instances is increasing. The x-axis is shown in log-scale.

approach is shown as a single horizontal line, as it does not use any supervised training examples.

From the results we can see that supervised-only CNN starts to match the ZSL performance only when the number of training examples is more than 512, and the fine-tuned CNN reaches the ZSL performance at 256 samples. This is a significant achievement considering that (i) ZSL approach uses zero-training examples from the target classes, and (ii) we are working with fine-grained categories that are hard to distinguish even by visual inspection of the image data. We expect ZSL performance to further improve following the advances in image representation, image resolution, class embeddings, and ZSL formulations.

Importantly, we should also note that the collection and annotation of even 256 training examples can be a very costly task: sample collection may require spatially surveying a very large area, and annotating them with class labels typically requires inspection of the instances or their close-by pictures by domain-experts, as the most fine-grained categories are very difficult to distinguish. For example, Figure 4.5 illustrates the Seattle region and the spatial distribution of our 16-class ZSL-test instances in this region; Figure 4.6 shows the spatial distribution of correct predictions for all instances of ZSL-test classes and Figure 4.7 illustrates the spatial distribution of correct predictions for each ZSL-test classes. In these figures, we observe that the instances and classes are scattered all around an area of 217 km^2 with mixture of true and false predictions, which casts the data collection and annotation a very time-consuming and challenging task. In this context, we believe that zero-shot learning of fine-grained categories can potentially become a central topic towards building semantically rich image understanding systems for remote sensing.

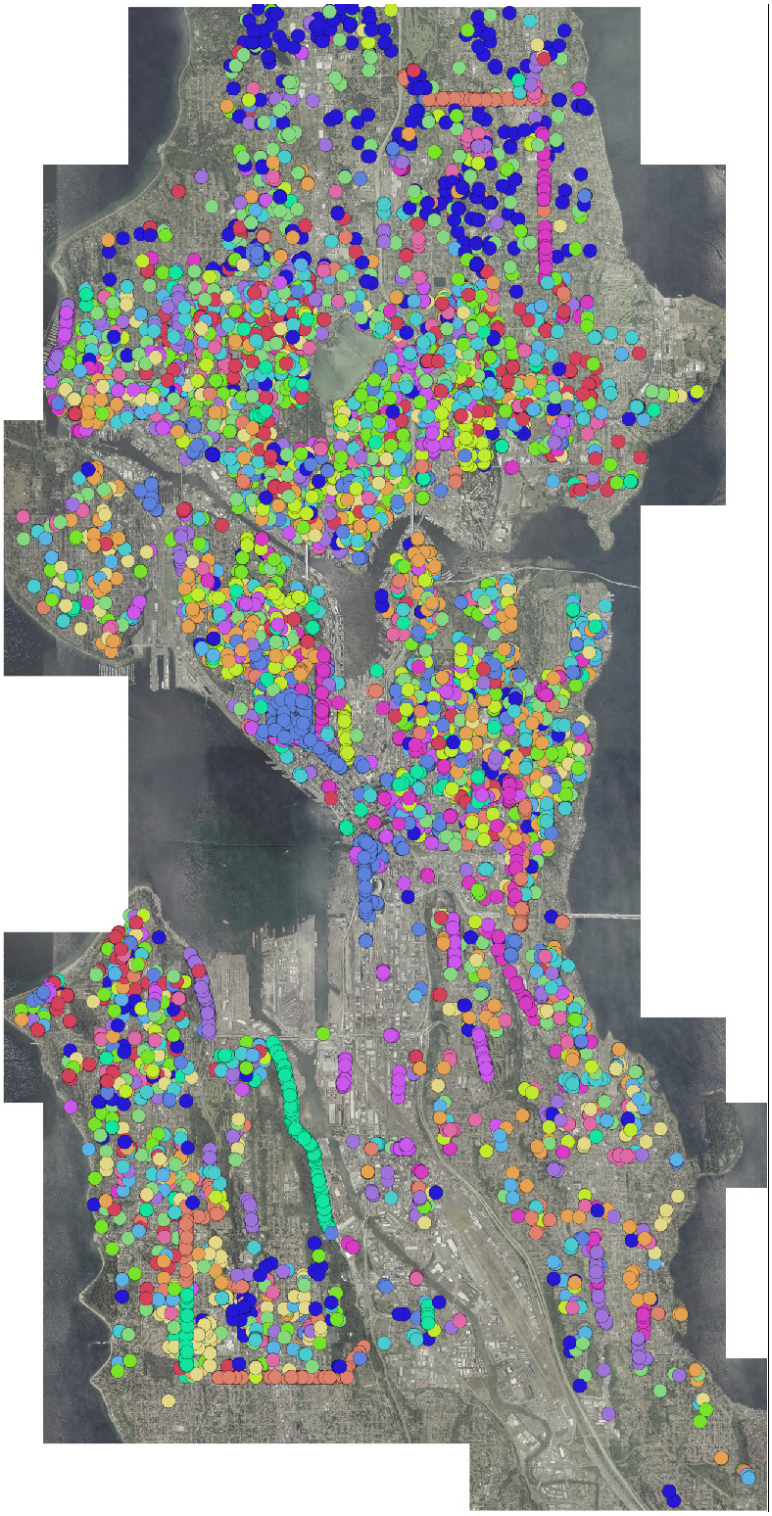

Figure 4.5: Spatial distribution of instances belonging to the zero-shot test (unseen) classes. Each point shows one instance, and the point colors represent the classes.

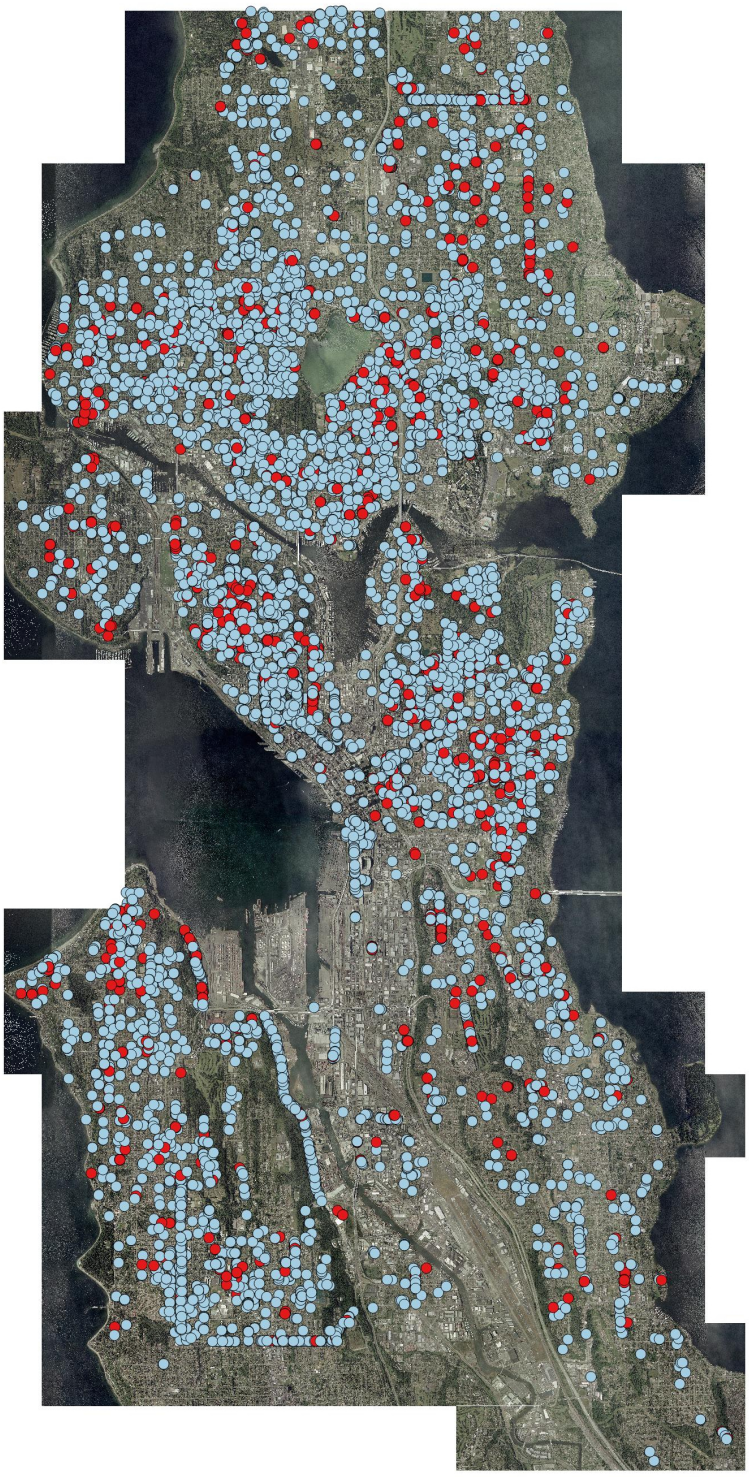

Figure 4.6: Spatial distribution of true predictions for instances belonging to the zero-shot test (unseen) classes. Each point shows one instance, and the point colors represent whether the instance was truely predicted or not. Red color indicates true prediction whereas blue color indicates wrong prediction.

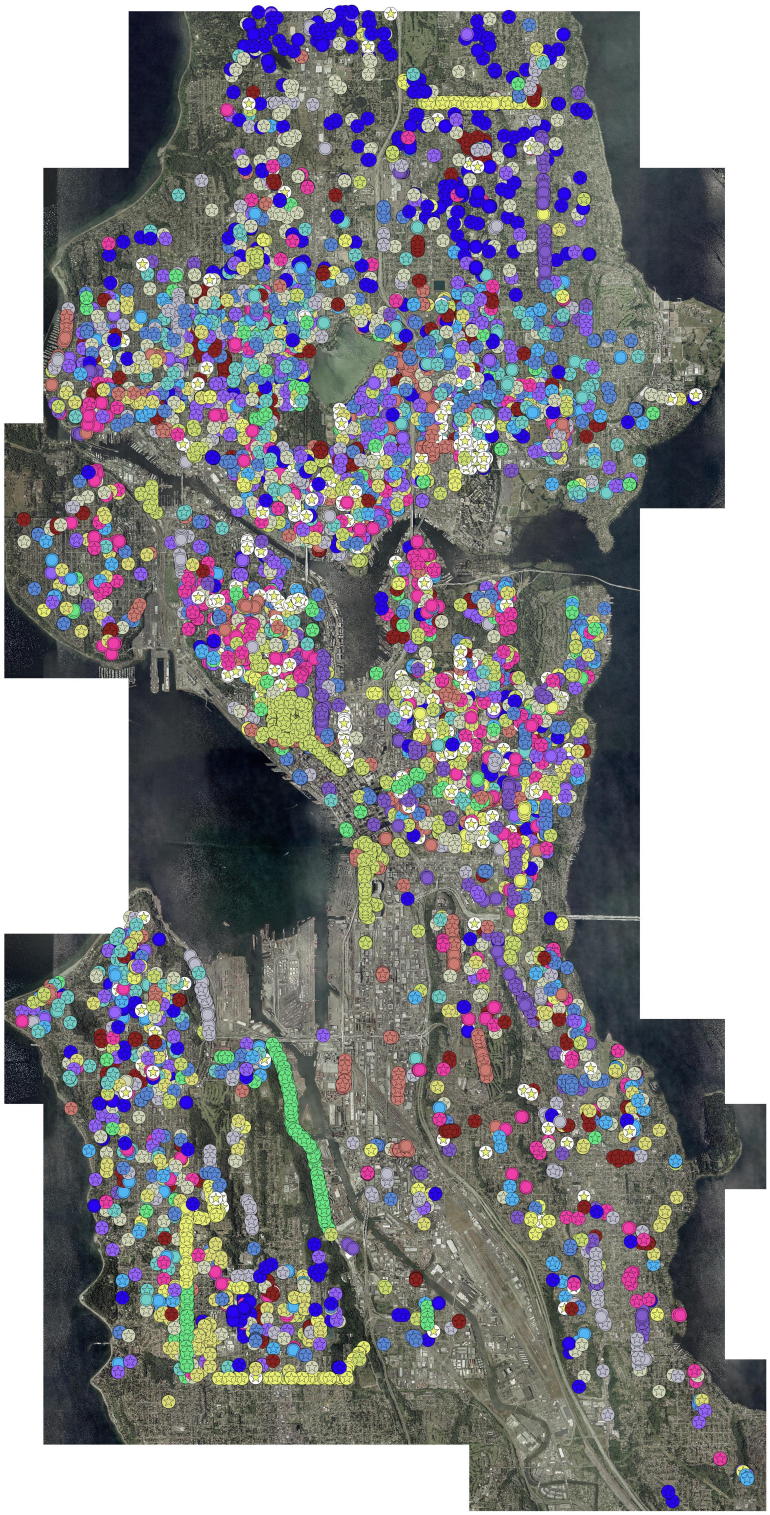

Figure 4.7: Spatial distribution of true predictions for each zero-shot test (unseen) classes. Each point shows one instance, the point colors represent the classes and the point shapes indicates the prediction correctness. Star shaped instances are wrongly predicted and circle shaped instances are correctly predicted.

Chapter 5

Multisource Fine-Grained Object Recognition

In this chapter, we first introduce the multisource object recognition problem and present a simple feature concatenation scheme for multisource recognition. Then, we explain our multisource weight estimation approach, and the realization of this approach via a deep neural network architecture. At the end of the chapter, detailed experimental analysis of our approach will be presented.

5.1 Multisource object recognition problem

In the multisource object recognition problem, we assume that there exists M different source domains, where the space of samples from the m-th domain is represented by $\mathcal{X}^m$. Our goal is to learn a classification function that maps a given tuple of input instances from the source domains $(x^1 \in \mathcal{X}^1, ..., x^M \in \mathcal{X}^M)$ to one of the target classes $y \in \mathcal{Y}$ where $\mathcal{Y}$ is the set of all target classes.

In this work, we focus on the problem of object recognition from multiple source images, where each source corresponds to a particular image sensor, such as RGB,

multispectral, hyperspectral, LIDAR, or similar. We are particularly interested in the utilization of remote sensing imagery, where the samples are typically collected from cameras with different viewpoints and elevations, resolutions, dates and time of day. Such differences in imaging conditions across the data sources makes the precise spatial alignment of the images very difficult, or even impossible in certain cases. In addition, the image contents may partially be different, due to changes in the region over time, and, occlusions in the scene.

In the next section, we first present a simple baseline approach towards utilizing such multiple sources, and, then we explain our approach towards addressing these challenges in a much more rigorous way.

5.1.1 Multisource object recognition by feature concatenation

A simple and commonly used scheme for utilizing multiple imagery in classification is to extract features independently across the images and then concatenating them later, which is often called *early fusion*. More precisely, for each source m, we assume that there exists a feature extractor, $\phi_m(x^m)$, such as a convolutional neural network, which maps the input x^m to a d_m-dimensional feature vector. Here it is presumed that all images $x^1, ..., x^m$ within a tuple x, which consists of the images from all sources, contains approximately the same region around a particular object instance. Then, the representation $\phi(x)$ for the tuple is obtained by concatenating per-source feature vectors:

$$\phi(x) = [\phi_1(x^1)^\top, ..., \phi_M(x^M)^\top]^\top. \tag{5.1}$$

Once the representation is obtained, the mapping from an image tuple to object classes can be carried out by concatenating image features from multiple sources. The overall approach is illustrated using plate notation in Figure 5.1.

The main assumption of the simple feature concatenation approach is that the representation obtained indepedently from each source successfully captures the characteristics of the object within the target region. However, registration across

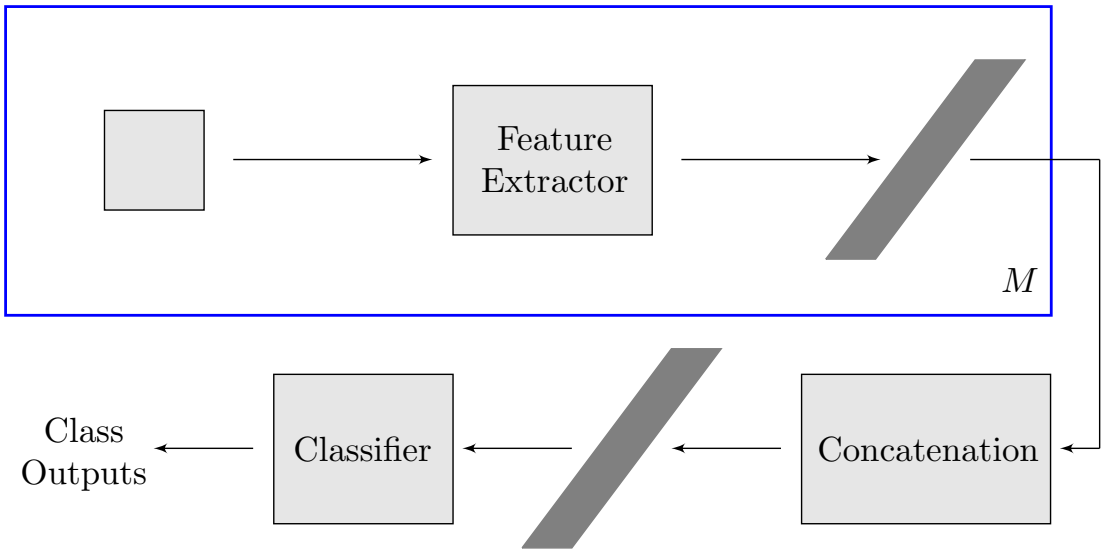

Figure 5.1: Basic multisource model. Feature-level fusion vector of multiple images acquired from M different sources is used for class identification. Note that plate notation is used for the illustration of different image source domains.

the sources is usually imprecise, which requires choosing relatively large regions to ensure that all images within a tuple contains the same object instance. In this case, however, the features extracted from these relatively large regions are likely to be dominated by background information, which can greatly degrade the accuracy of the final classification model.

This problem is tackled by our multisource weight estimation approach, which we explain in the following section.

5.2 Multisource Weight Estimation Framework

To handle this scenario, we propose a weight estimation framework assuming that (i) image patches from reference source are correctly aligned with ground truth labels and (ii) correct spatial region of objects in image patches from other sources can be located in region proposals extracted from these patches depending on their weights. Consider $\mathcal{S} \in \{1, ..., M\}$ as the set of reference domains and $\overline{\mathcal{S}} = \{t : t \in \{1, ..., M\}, t \notin \mathcal{S}\}$ as the target domains, weight estimation function, F_W, calculates the region proposal weights of target images by leveraging the reference image representations. Each estimated weight represents the confidence in assuming that true object region is in corresponding proposal. As a simplification of derivation, it will be assumed after this point that reference image pacthes are taken from only one domain and other domains are regarded as target domains in which alignment may be incorrect. Thus, the weight estimation function gives region proposal weights:

$$F_W(x_1^t, ..., x_R^t, \phi_s(x^s)) = [w_1^t, ..., w_R^t]^\top \tag{5.2}$$

where x_i^t, $i \in 1, ..., R$ is the i^{th} region proposal of an image pacth from the t^{th} target domain considering a series of R different region proposals, w_i^t is the weight of i^{th} region proposal and x_s is the image patch from reference domain s. After finding weights, estimated representation of this image is found in the following

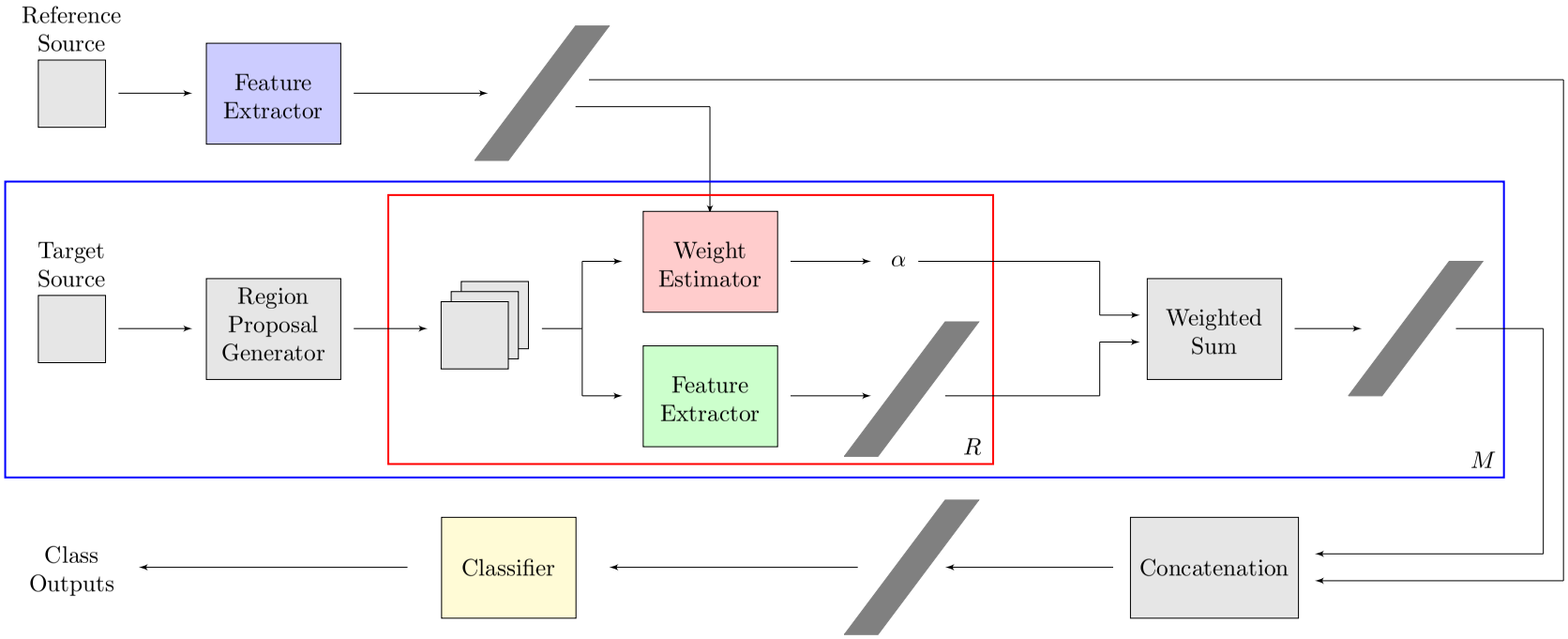

Figure 5.2: Our weight estimation framework. Unlike the simple concatenation between all images in basic multisource model, we first differentiate different image domains as reference ($\mathcal{S}$) and target ($\overline{\mathcal{S}}$) with respect to their alignment exactness. Correspondence between an object in an image (x^s) from the source domain and the ground truth label of this image is assumed as correct. However, this can not be a case for an image (x^t) from target domains. For this type of image, region proposals ($\{x_1^t, ..., x_R^t\}$) and representations of source images are used to estimate the weight ($\{w_1, ..., w_R\}$) of region proposals' representation. Thus, feature-level fusion ($\phi^{'}(x)$) of weighted sum of the target proposal features ($\phi^{'}(x^t)$) and source feature ($\phi_s(x^s)$) are used for final class prediction. Note that, there is only one source image and multiple target image in the figure. Plate notation is used for indicating R region proposals for each target and for indicating M different sources.

way:

$$\phi^{'}(x^t) = \sum_{i=1}^{R} \alpha_i^t \phi_t(x_i^t) \tag{5.3}$$

where α^t is the normalized weight vector which is defined as follows:

$$\alpha^t = \left[\frac{w_1^t}{\sum_{i=1}^{R} w_i^t}, ..., \frac{w_R^t}{\sum_{i=1}^{R} w_i^t} \right]^\top. \tag{5.4}$$

Thus, we can now define the new fused representation of images from all different sources as again a dM-dimensional vector:

$$\phi'(x) = [\phi_s(x^s)^\top, \phi^{'}(x^2)^\top, ..., \phi^{'}(x^M)^\top]^\top. \tag{5.5}$$

Once the new representation is learned, fine-grained recognition is achieved by assigning the set of input images to the class y^* with one of the classifier, C, as follows:

$$y^* = \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\, C(\phi^{'}(x)|y). \tag{5.6}$$

An illustration of our weight estimation framework can be found in Figure 5.2.

For the extraction of region proposals, we trace all possible regions whose size are determined as the expected area covering an object. For this, we search all proposals with sliding window in an image whose size is much larger than the object as parameter. Additionally, we do not use any padding method.

5.3 Neural Network Model

We utilize a deep neural network in order to realize our overall framework. The motivation behind this preference is having the capability (*i*) to exploit both the spectral information and the spatial texture content from all different domains, (*ii*) to estimate the region proposal weights, (*iii*) to obtain class confidence scores at the same time.

To achieve this aim, the proposed architecture was formed by the combination of three deep convolutional neural network (CNN) and a fully-connected block

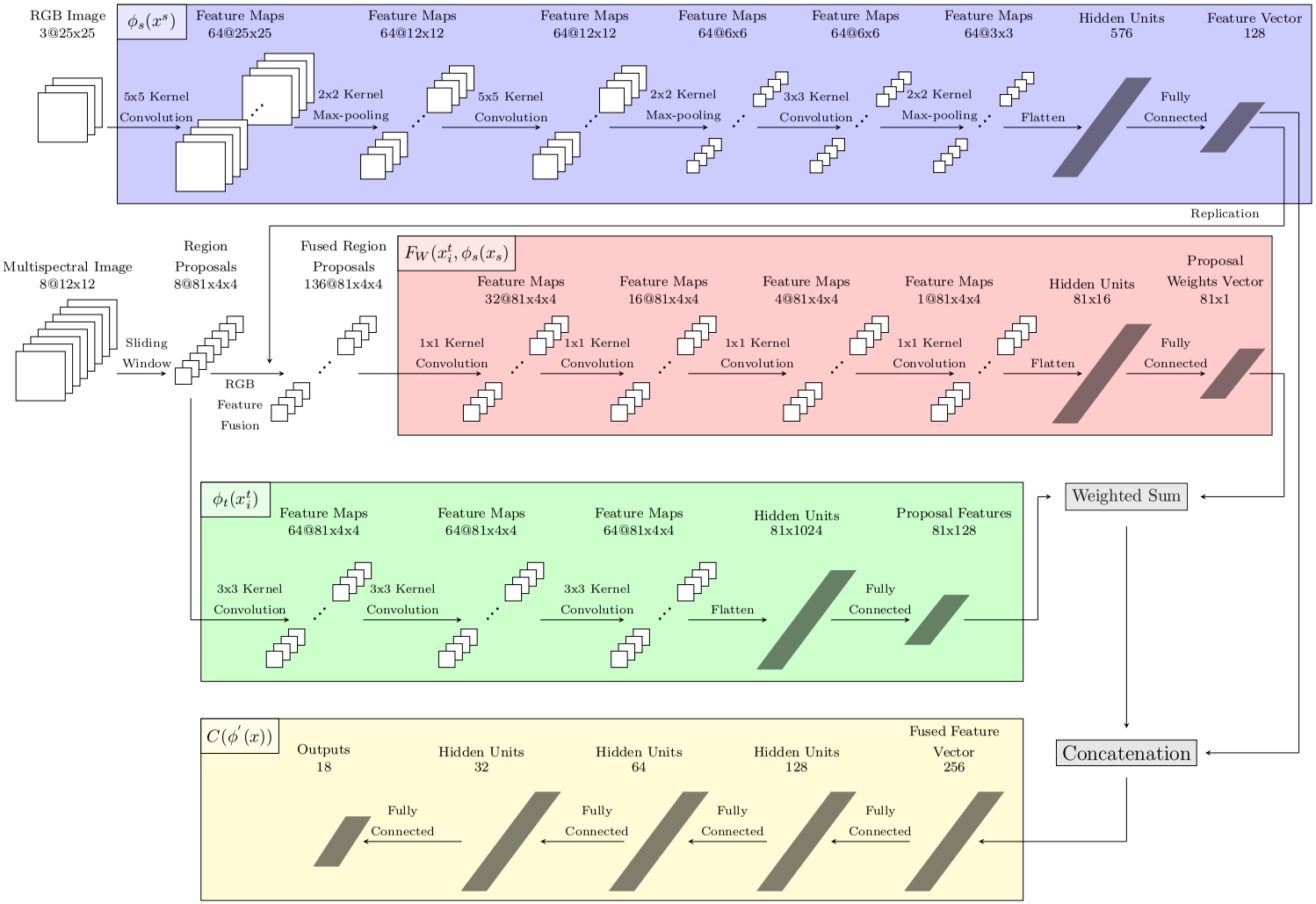

Figure 5.3: Proposed deep neural network architecture with four branch. First one (ϕ_s) acts as feature extractor for reference domain, aerial RGB image. It contains there convolutional layers containing 64 filters each with sizes 5×5, 5×5, and 3×3, respectively, followed by a fully-connected layer containing 128 neurons. We apply max-pooling after each convolutional layer. Second branch (F_W) estimates the weights of region proposals of target domain, multispectral image patches, with the help of the feature vector of corresponding RGB image patches. This one contains four convolutional layers containing 32, 16, 4, 1 filters each with size 1×1, followed by a fully-connected layer containing 16 and 1 neurons, respectively. Third one (ϕ_t)gives the representations of multispectral image pacthes' region proposals and includes there convolutional layers containing 64 filters each with size 3×3, 5×5 followed by a fully-connected layer containing 128 neurons. Last branch (C) calculates the class scores from the concatenation of weighted sum of region proposals' representations and RGB feature vector. It consists of four fully-connected layers containing 128, 64, 32 and 18 neurons, which give class scores, respectively. Note that, the feature map sizes and descriptive names are stated at the top of each layer.

of layers. First part extracts the image representations of reference domain. For this, we adapt the architecture from our previous work [53]. Second block takes the region proposals of the image pacthes from target domains and appends feature vectors of reference domain image patches to the end of each pixels' band channel via replicating them. Four convolutional layers with 1×1 dimensional filters and a fully-connected layer estimates the weights of region proposals. Third branch in which image representations for each region proposals are extracted differs from the first branch with using only 3×3 filters and not using max-pooling. Then, concatenation of weighted sum of region proposal features and the representatitons of reference image pacthes goes to the last branch in which four fully-connected layers with 128, 64, 32 and 18 neurons give the final class scores. Stride for all convolutional layers is set at 1 to escape from information loss. We also use zero-padding in order to avoid reduction in the spatial dimensions over convolutional layers.

While the number of filters is selected as 64 for each convolutional layer in the first and third branch, correctly finding the point between model capacity and preventing overfitting is aimed. However, for the weight estimation branch, we prefer to use decreasing number of filters from 32 to the 1 in order to have correct number of weights at the end. Finally, altough we have tried to use more deeper or wider models, we reached the best performance with the presented network. Additional details of the architecture can be found in Figure 5.3. As we have one source domain with RGB images and one target domain with multispectral images (see Chapter 3 for data set details), the figure illustrates our framework in that way. However, any number of target and source domains without any restriction for the number of spectral bands can be applied to the neural network.

Training the model is carried out over the classes by employing cross-entropy loss meaning that maximization of label log-likelihood is aimed in the training set. For enhancement on training, we benefitted from Dropout regularization [57] and Batch Normalization [58]. Additional training details and a comparison of our model are provided in Section 5.4.

5.4 Experiments

In this section, experimental analysis of our multisource framework is presented. Firstly, experimental setup is specified. Then, we compare the evaluation results for using different domains alone. Thirdly, evaluation of our deep neural network model is stated with its results with different parameters in addition to basic multisource model evaluation as a baseline. Finally, we discuss the results.

5.4.1 Experimental setup

In our experiments, we follow the same class split with the Section 4.5 and we evaluate our results on both 18 and 40 classes. For both cases, we split images from all sources into *train* (60%), *validation* (20%) and *test* (20%) subsets. Based on our previous observations, we add perturbations to training images of all domains. To do so, we shift each one randomly with an amount ranging from zero to 20% height/width.

For all experiments, training is carried out on the train set with the Adam method [55] of stochastic gradient descent and we tune hyper-parameters on the validation set by maximizing the performance in order to have a fair comparison and the initial learning rate of Adam, mini-batch size and ℓ_2-regularization weight are set to 10^{-3}, 100, and 10^{-5}, respectively.

Throughout our experiments, we prefer to use normalized accuracy, for which per-class accuracy ratios are averaged, as the performance metric. With the help of this choice, avoding biases because of classes having a large number of examples is aimed.

Table 5.1: Single-source results for 18 classes (in %)

	Random guess	RGB	Multispectral 4×4 patches	Multispectral 12×12 patches
Normalized accuracy	5.6	34.6	39.0	**47.7**

5.4.2 Effect of Different Sources on Supervised Classification

Before the evaluation of our multisource model, we appraise different CNN models on images from different single domains. For this purpose, we use the first branch of our deep neural network for aerial RGB images and third branch for multispectral satellite images as CNN models by adding one more fully-connected layer that map the output of the last fully-connected layer to the 18 different class scores. Additionally, the third branch takes images directly instead of region proposals as input.

Comparision for the performance of different sources on the test set is shown in Table 5.1. These results show that all results are clearly better than random guess baseline (5.6%). Additionally, we can see that the CNN model for multispectral images outperforms the CNN model for RGB images (34.6%). Although the patch size that coincides with the spatial region extent of RGB images is 4×4, the result for bigger patch size (12×12) in multispectral images (47.7%) outperforms the corresponding size (39.0%). Beacuse object region does not fit some misaligned images for multispectral domain, it negatively affects the performance.

These results show how difficult fine-grained classification problem is since it is different from the traditional classification problems that has more distinguishable classes. Additionally, for images which is not correctly aligned with ground trurth labels, bigger patch size can increase the performance. However, using bigger bathces does not resolve the alignment problem directly. With this, possible regions for objects are simply taken into account by the CNN model.

Table 5.2: Multisource fine-grained classification results (in %)

18 classes	Normalized Accuracy
Random guess	5.6
Basic multisource model	55.5
Weight estimation framework	**63.7**
40 classes	
Random guess	2.5
Basic multisource model	39.1
Weight estimation framework	**46.6**

5.4.3 Multisource Fine-grained Classification

For the experimental analysis of multisource scenario, we evaluate our deep neural network model on multisource image patches while comparing against basic multisource model.

For our proposed approach, we train whole model without learning different branches seperately that is called end-to-end learning. In this manner, the network accepts image pacthes from multiple sources and produces class scores and back propagation is carried out with respect to the calculated loss from class labels. Thus, our neural network learns both (i) the spatial distribution of object regions in target domain by finding the weights of which region is likely to contain a true object and (ii) mapping from multiple images to class probabilities at the same time. For this, our model search true regions in larger windows. Figure 5.4 presents the effect of region proposal's spatial dimension size for multispectral image pacthes (target domain) on our model's performance. It shows that finding weights for region proposals having 4×4 dimension gives the best result 63.7% on the test set. This is also matching spatial dimension with RGB images.

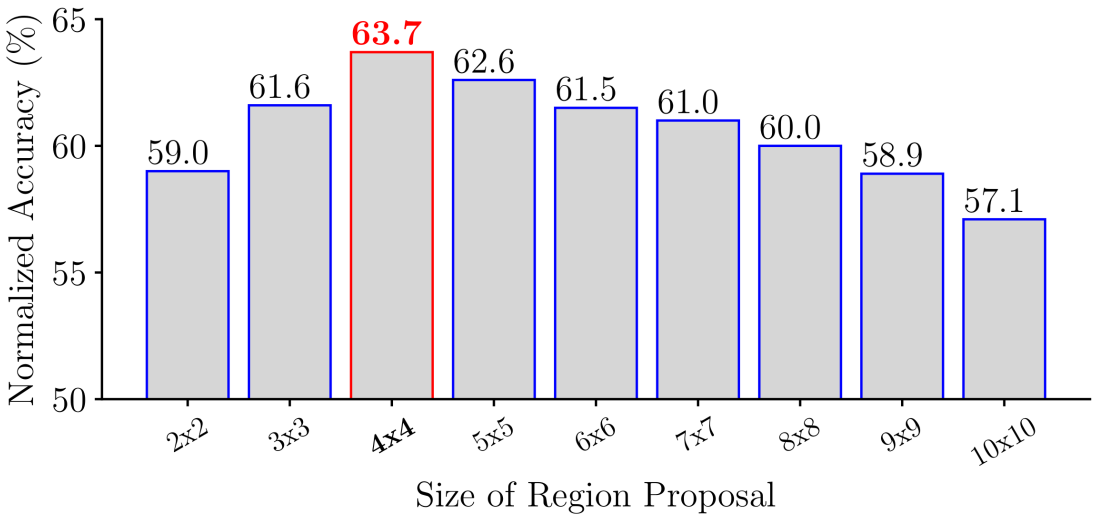

Figure 5.4: Effect of region proposal size on classification performance. The y-axis is shown in the range of $[50-65]$ when the size of region proposal is increasing from 2×2 to 10×10.

Table 5.3: Confusion matrix for the classification of 40 classes when multiple sources are used with the weight estimation framework. Since rows indicate true classes, columns indicate the predicted classes. Each entry in table is the classification rates.

	(European) White Birch	Thundercloud Purpleleaf Plum	Norway Maple	English Midland Hawthorn	Blireiana Purpleleaf Plum	Apple/Crabapple	Cherry (Flowering) Plum, Myrob	Sweetgum	Kwanzan Flowering Cherry	Red Maple	Littleleaf Linden	Double Chinese Cherry	London Plane (Tree)	Red Oak	(Smooth) Japanese Maple	Red Sunset Red Maple	Callery Pear	Bigleaf Maple	Honey Locust	Horse Chestnut	Common Hawthorn	European Hornbeam	Sycamore Maple	Western Red Cedar	Flame Narrowleaf Ash	Common (Euro) Mountain Ash	Green (Red) Ash	Kousa Dogwood	Autumn Flowering Cherry	Douglas Fir	Orchard (Common) Apple	Autumn Brilliance Serviceberry	Washington Hawthorn	Scarlet Oak	Paperbark Maple	Japanese Snowbell Tree	Katsura Tree	Pacific Sunset Maple	Norwegian Sunset Maple	Flame Amur Maple
(European) White Birch	0.48	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.03	0.01	0.03	0.04	0.01	0.01	0.01	0.02	0.01	0.02	0.01	0.02	0.02	0.01	0.02	0.03	0.01	0.01	0.00	0.01	0.02	0.03	0.05	0.01	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.02
Thundercloud Purpleleaf Plum	0.00	0.50	0.01	0.00	0.05	0.00	0.34	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.03	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Norway Maple	0.01	0.01	0.35	0.00	0.00	0.00	0.01	0.02	0.01	0.02	0.01	0.00	0.02	0.04	0.01	0.00	0.02	0.13	0.00	0.08	0.00	0.02	0.09	0.00	0.01	0.02	0.02	0.01	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.02	0.02	0.00
English Midland Hawthorn	0.06	0.00	0.00	0.16	0.02	0.04	0.01	0.01	0.03	0.00	0.01	0.01	0.00	0.00	0.02	0.01	0.02	0.01	0.01	0.01	0.24	0.01	0.00	0.02	0.00	0.03	0.00	0.02	0.06	0.02	0.08	0.01	0.02	0.00	0.00	0.01	0.01	0.00	0.00	0.01
Blireiana Purpleleaf Plum	0.02	0.08	0.00	0.00	0.34	0.02	0.21	0.00	0.01	0.00	0.00	0.02	0.00	0.00	0.03	0.00	0.01	0.01	0.01	0.02	0.02	0.00	0.00	0.01	0.01	0.02	0.00	0.04	0.03	0.01	0.02	0.01	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.02
Apple/Crabapple	0.03	0.02	0.01	0.03	0.05	0.04	0.03	0.01	0.09	0.00	0.00	0.05	0.00	0.00	0.02	0.00	0.04	0.01	0.01	0.01	0.07	0.01	0.00	0.01	0.00	0.03	0.00	0.03	0.11	0.04	0.11	0.02	0.01	0.01	0.01	0.02	0.01	0.01	0.00	0.04
Cherry (Flowering) Plum, Myrob	0.01	0.29	0.01	0.00	0.08	0.00	0.49	0.00	0.01	0.00	0.01	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.02	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
Sweetgum	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.66	0.01	0.02	0.01	0.00	0.00	0.04	0.01	0.01	0.02	0.02	0.03	0.01	0.00	0.00	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.00	0.00	0.04	0.00	0.01	0.00	0.00	0.00	0.00
Kwanzan Flowering Cherry	0.04	0.00	0.01	0.00	0.01	0.03	0.02	0.01	0.22	0.02	0.01	0.09	0.00	0.01	0.02	0.01	0.03	0.01	0.01	0.01	0.01	0.00	0.01	0.01	0.00	0.03	0.00	0.01	0.09	0.02	0.14	0.01	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.02
Red Maple	0.02	0.00	0.03	0.00	0.00	0.00	0.01	0.07	0.02	0.06	0.01	0.03	0.00	0.07	0.00	0.29	0.05	0.02	0.01	0.01	0.00	0.02	0.04	0.01	0.05	0.01	0.01	0.00	0.01	0.02	0.02	0.00	0.01	0.02	0.00	0.00	0.02	0.02	0.01	0.01
Littleleaf Linden	0.03	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.00	0.00	0.01	0.00	0.02	0.01	0.00	0.00	0.01	0.00	0.03	0.02	0.01	0.04	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.01	0.01	0.00	0.00	0.01	0.00	0.00	0.00
Double Chinese Cherry	0.06	0.01	0.01	0.00	0.02	0.02	0.03	0.01	0.18	0.03	0.02	0.10	0.00	0.01	0.01	0.02	0.04	0.02	0.02	0.01	0.01	0.00	0.01	0.01	0.00	0.01	0.00	0.03	0.12	0.01	0.12	0.01	0.02	0.00	0.01	0.03	0.00	0.00	0.00	0.02
London Plane (Tree)	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.82	0.03	0.01	0.01	0.00	0.03	0.01	0.01	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.00
Red Oak	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.03	0.00	0.01	0.00	0.01	0.01	0.45	0.00	0.02	0.01	0.06	0.02	0.01	0.00	0.01	0.02	0.01	0.00	0.00	0.02	0.00	0.00	0.03	0.00	0.00	0.00	0.17	0.00	0.00	0.01	0.04	0.01	0.00
(Smooth) Japanese Maple	0.01	0.02	0.00	0.00	0.03	0.02	0.03	0.01	0.01	0.00	0.00	0.01	0.01	0.00	0.26	0.00	0.03	0.02	0.01	0.01	0.01	0.00	0.00	0.02	0.00	0.01	0.00	0.15	0.01	0.02	0.05	0.07	0.00	0.00	0.00	0.06	0.00	0.01	0.00	0.05
Red Sunset Red Maple	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.03	0.00	0.02	0.00	0.06	0.00	0.61	0.01	0.02	0.02	0.01	0.00	0.00	0.02	0.00	0.05	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.02	0.00	0.00
Callery Pear	0.01	0.00	0.01	0.00	0.01	0.01	0.01	0.06	0.02	0.01	0.01	0.02	0.00	0.01	0.02	0.01	0.40	0.02	0.02	0.03	0.02	0.05	0.02	0.01	0.02	0.00	0.00	0.05	0.01	0.00	0.06	0.01	0.01	0.00	0.00	0.06	0.01	0.02	0.01	0.01
Bigleaf Maple	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.04	0.03	0.02	0.00	0.00	0.57	0.01	0.10	0.00	0.01	0.04	0.00	0.01	0.02	0.01	0.01	0.00	0.06	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.01	0.00
Honey Locust	0.01	0.00	0.02	0.00	0.01	0.00	0.01	0.04	0.01	0.01	0.00	0.01	0.03	0.01	0.00	0.01	0.00	0.02	0.71	0.01	0.00	0.00	0.01	0.02	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.02	0.01	0.00	0.01	0.01	0.00	0.00
Horse Chestnut	0.03	0.00	0.02	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.02	0.00	0.01	0.01	0.01	0.01	0.01	0.07	0.00	0.72	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.02	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.00
Common Hawthorn	0.04	0.01	0.00	0.07	0.04	0.02	0.01	0.01	0.01	0.00	0.01	0.03	0.01	0.00	0.01	0.00	0.05	0.01	0.00	0.06	0.28	0.00	0.03	0.03	0.00	0.03	0.00	0.01	0.03	0.03	0.05	0.01	0.05	0.01	0.01	0.01	0.03	0.01	0.00	0.03
European Hornbeam	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.02	0.00	0.00	0.04	0.00	0.86	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.01	0.01	0.00	0.00
Sycamore Maple	0.00	0.00	0.06	0.00	0.00	0.00	0.01	0.01	0.00	0.01	0.01	0.00	0.02	0.04	0.00	0.03	0.02	0.12	0.02	0.03	0.00	0.00	0.55	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01
Western Red Cedar	0.03	0.00	0.00	0.00	0.01	0.00	0.01	0.01	0.01	0.00	0.01	0.00	0.01	0.00	0.00	0.01	0.02	0.05	0.00	0.01	0.01	0.03	0.00	0.56	0.00	0.01	0.00	0.04	0.00	0.13	0.01	0.01	0.00	0.01	0.00	0.01	0.01	0.00	0.01	0.00
Flame Narrowleaf Ash	0.01	0.00	0.01	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.01	0.01	0.00	0.01	0.00	0.01	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.84	0.00	0.01	0.01	0.01	0.01	0.01	0.00	0.00	0.03	0.01	0.00	0.00	0.00	0.00	0.00
Common (Euro) Mountain Ash	0.03	0.00	0.02	0.00	0.01	0.04	0.02	0.00	0.02	0.01	0.00	0.02	0.00	0.02	0.00	0.02	0.02	0.06	0.01	0.05	0.04	0.01	0.02	0.03	0.01	0.24	0.00	0.03	0.04	0.05	0.12	0.00	0.00	0.02	0.00	0.01	0.03	0.00	0.00	0.02
Green (Red) Ash	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.02	0.00	0.01	0.00	0.00	0.02	0.05	0.00	0.02	0.00	0.06	0.00	0.00	0.00	0.00	0.03	0.02	0.06	0.00	0.66	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.02	0.01	0.00	0.00
Kousa Dogwood	0.02	0.02	0.00	0.00	0.02	0.01	0.02	0.02	0.01	0.01	0.00	0.05	0.00	0.01	0.03	0.00	0.02	0.02	0.00	0.00	0.01	0.00	0.00	0.02	0.00	0.00	0.00	0.45	0.05	0.01	0.10	0.03	0.00	0.00	0.00	0.05	0.00	0.02	0.00	0.04
Autumn Flowering Cherry	0.02	0.00	0.01	0.00	0.04	0.04	0.01	0.00	0.08	0.00	0.01	0.06	0.00	0.00	0.02	0.01	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.02	0.01	0.01	0.00	0.02	0.44	0.02	0.08	0.00	0.02	0.00	0.01	0.00	0.01	0.00	0.06
Douglas Fir	0.02	0.00	0.00	0.00	0.01	0.00	0.03	0.01	0.00	0.00	0.01	0.00	0.02	0.02	0.02	0.01	0.01	0.01	0.04	0.01	0.02	0.01	0.00	0.00	0.05	0.00	0.03	0.00	0.01	0.00	0.64	0.01	0.02	0.00	0.01	0.00	0.00	0.01	0.00	0.00
Orchard (Common) Apple	0.07	0.00	0.00	0.00	0.01	0.02	0.01	0.02	0.05	0.00	0.01	0.05	0.01	0.00	0.01	0.00	0.02	0.01	0.01	0.02	0.02	0.00	0.01	0.00	0.00	0.02	0.00	0.07	0.04	0.02	0.39	0.02	0.01	0.00	0.02	0.02	0.01	0.00	0.00	0.05
Autumn Brilliance Serviceberry	0.01	0.00	0.01	0.00	0.02	0.02	0.02	0.00	0.03	0.02	0.01	0.04	0.01	0.00	0.04	0.00	0.00	0.02	0.00	0.02	0.01	0.00	0.00	0.04	0.00	0.02	0.01	0.12	0.07	0.02	0.03	0.34	0.01	0.00	0.01	0.04	0.00	0.01	0.02	0.02
Washington Hawthorn	0.01	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.01	0.01	0.02	0.00	0.00	0.03	0.01	0.02	0.00	0.00	0.01	0.00	0.01	0.02	0.00	0.03	0.00	0.66	0.00	0.00	0.02	0.02	0.01	0.01	0.03
Scarlet Oak	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.20	0.01	0.01	0.00	0.04	0.01	0.01	0.00	0.00	0.01	0.00	0.05	0.01	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.55	0.01	0.00	0.00	0.01	0.01	0.00
Paperbark Maple	0.01	0.00	0.00	0.00	0.01	0.00	0.01	0.05	0.01	0.00	0.00	0.03	0.00	0.02	0.03	0.00	0.01	0.05	0.02	0.00	0.01	0.01	0.01	0.01	0.01	0.00	0.01	0.11	0.05	0.01	0.03	0.09	0.00	0.01	0.28	0.03	0.00	0.03	0.01	0.00
Japanese Snowbell Tree	0.01	0.02	0.02	0.00	0.02	0.01	0.01	0.01	0.03	0.00	0.00	0.03	0.00	0.00	0.02	0.01	0.04	0.03	0.00	0.01	0.01	0.03	0.00	0.00	0.01	0.02	0.00	0.09	0.03	0.03	0.14	0.00	0.00	0.00	0.02	0.35	0.00	0.00	0.00	0.00
Katsura Tree	0.01	0.00	0.01	0.00	0.00	0.00	0.01	0.03	0.00	0.01	0.01	0.01	0.00	0.00	0.03	0.00	0.05	0.00	0.02	0.03	0.00	0.02	0.00	0.05	0.01	0.01	0.00	0.00	0.00	0.00	0.05	0.00	0.01	0.01	0.00	0.01	0.50	0.02	0.00	0.06
Pacific Sunset Maple	0.00	0.01	0.02	0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.04	0.01	0.02	0.07	0.01	0.01	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.03	0.00	0.04	0.00	0.00	0.03	0.03	0.02	0.49	0.05	0.01
Norwegian Sunset Maple	0.00	0.00	0.00	0.00	0.02	0.03	0.00	0.00	0.02	0.02	0.02	0.00	0.00	0.03	0.05	0.02	0.05	0.03	0.00	0.00	0.00	0.00	0.03	0.02	0.00	0.02	0.00	0.02	0.00	0.00	0.00	0.02	0.00	0.02	0.03	0.05	0.02	0.15	0.37	0.00
Flame Amur Maple	0.02	0.00	0.02	0.00	0.02	0.02	0.00	0.00	0.04	0.00	0.02	0.00	0.00	0.00	0.02	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.02	0.00	0.10	0.02	0.10	0.06	0.00	0.00	0.00	0.02	0.00	0.02	0.00	0.48

Additionally, we also compare our approach against basic multisource model in which simply relation between images from multilpe domain and class probabilities is learnt. For this, there is no second branch which finds proposal weights as in our model. Thus, it simply concatenates feature vectors coming from the feature extractor branches and finds class probabilities. Comparision for the both model evaluated on the test set of multisource images is shown in Table 5.2. Our multisource model achieves a 63.7% normalized accuracy at 18 classes and 46.6% normalized accuracy at 40 classes. These are clearly better than the random guess baselines (5.6% and 2.5%) and basic multisource model (55.5% for 18 classes and 39.1% for 40 classes). Additionally, these are also better than single-source RGB and multispectral evaluation results. Thus, all results confirm the validity of our weight estimation framework at both 18 and 40 classes while considering the complexity of fine-grained classification problem. Table 5.3 shows the confusion matrix for the classification 40 classes. To exemplify, 34% of thundercloud plum tree samples are wrongly predicted as cherry plum and 29% of cherry plum tree samples are wrongly predicted as thundercloud plum since these are two different plum species and discriminating various types of plum is a very challenging task on 25×25 RGB and 12×12 multispectral images, even for experts. As a more extreme scenario, only 6% percent of red maple tree samples are correctly classified and 29% of them are predicted as sunset red maple. Since these trees are only distinguished with respect to their subspecies level in scientific taxonomy and they have almost the same visual appearance, even seperating them from ground-view images with a high accuracy is too hard.

5.4.4 Multisource Fine-grained Zero-shot Learning

We also evaluate our multisource approach on zero-shot learning (ZSL) scenario in which an object is tried to recognize among new unseen categories. There is no training examples of these categories. For this, we follow the same methodology, class split and experimental setup with the single-source scenario except that the way of how region representations are extracted and the use of multiple sources. In order to be able to use multiple domain in this ZSL approach, all sources must

Table 5.4: Zero-shot learning results (in %)

Type of Image Representation	Normalized Accuracy
Random guess	6.3
RGB	14.3
Multispectral 4×4 patches	15.2
Multispectral 12×12 patches	16.7
Simple fusion of features	15.8
Proposed approach	**17.7**

be represented in a vector as the embedding of images that can play a significant role on the performance. For this, we train this ZSL approach on the outputs of first fully-connected layer in the fourth branch of our model. As a comparision, we also use the simple fusion of image features from RGB and multispectral images as the image representations after basic multisource is trained once. We also evaluated single source case in which RGB image features are used from Section 4.2 and multispectral image features are extracted from the outputs of the first fully connected layer in the fourth branch of our model which is trained solely before.

Comparision of different multisource and single source representations of images evaluated on 16 ZSL-test classes is shown in Table 5.4. Considering the single source case, using corresponding patch size (4×4) with RGB images for multispectral images (15.2%) is better than RGB performance (14.3%) and using bigger patch size (12×12) for multispectral images (16.7%) gives the best result. In terms of multisource performance, altough the performance is decreased to 15.8% when fusion of feature vectors with the basic multisource model is used, ZSL performance reaches the highest value (17.7%) when outputs of the first fully-connected layer in the fourth branch of our model are preferred as image representations. Thus, in addition to supervised classification results, ZSL results also highlight the efficacy of our multisource model.

5.4.5 Discussion

Demonstrated results up to now highlight that the proposed multisource approach performs remarkably better comparing to both the random guess baseline and the feature-level fusion model which is one of the main approach in remote sensing multisource image analysis. Nevertheless, how well our model tackles the alignment problem for the target domain is still an significant question to ask and it is almost the same with the inquiry about whether our model finds the correct weights for region proposals or not.

To investigate this, Figure 5.5 shows the weight distribution of randomly selected test images. Weights are shown under the RGB bands of multispectral image pacthes and to the right of corresponding RGB image pacthes covering the same area. This visualization shows that our model gives higher weights to region proposals having more related spatial content. Therefore, our model is capable of correctly aligning target image pacthes while classifying them into different categories.

Figure 5.5: Weights of region proposals estimated from randomly selected 12 multispectral test images. For each sample, 25×25 pixel RGB patch is shown on the left, the normalized RGB bands of the 12×12 multispectral patch covering the same region with RGB patch is shown on top-right, and the weight distribution of region proposals is shown on bottom-right. Each weight value is the weight of a 4×4 region proposal whose center pixel corresponds to the pixel of this weight value.

Chapter 6

Conclusion

We studied the different scenarios of fine-grained object recognition task in remotely sensed imagery that are zero-shot learning in single source domain and alignment problem in multisource domain.

For single source fine-grained object recognition, to cope with the difficulty of learning a very large number of very similar object categories as well as the need for being able to recognize classes even when there are no training examples, our framework exploited alternative sources of auxiliary information to build an association between the seen and unseen classes. The proposed approach learned a bilinear function from the seen classes so that the compatibility between the visual characteristics observed in the input image data and the auxiliary information that described the semantics of the classes of interest is modeled. Then, we showed how this compatibility function could be used for performing knowledge transfer during the inference of the unseen classes. Extensive experiments using different partitionings of a challenging aerial data set with 40 types of street trees defined as fine-grained target classes showed that our method obtained 14.3% classification accuracy, which was significantly better than random guessing (6.3%) for 16 test classes and three other zero-shot learning algorithms from the literature. Future work includes new representations for auxiliary information that models different aspects of spectral and spatial data characteristics as well as the

domain-specific class semantics.

For multisource fine-grained object recognition, in order to deal with the complexity of learning many sub-categories having subtle differences from multiple image sources with diverse spatial and spectral resolutions and with misregistration errors, we proposed a framework that simultaneously selects the most compatible image patches from potentially misaligned image regions using a weight estimation framework and learns the classification function from the selected multisource image representations. Our approach learns the likelihood weights of each region proposals in a bigger image window from target domains having not precise alignment by leveraging the image representations of source domains which are assumed as reliable. Then, weighted sum of the feature vectors of region proposals provide a new image representation for target domains. Thus, feature-level fusion of new estimated image representations from target domains and traditionally extracted embeddings from source domains are used to predict true classes of objects. Broadly conducted experiments using the street trees image data set from aerial and satellite sources having different spatial and spectral resolutions showed that our approach achieved 63.7% and 46.6% classification accuracies for 18 and 40 classes, respectively, and performed significantly better than the most commonly used feature-level fusion approach that achieved 55.5% and 39.1% for the same settings. Future work can contain the evaluation of our model on more diversified domains and model enhancements to solve more different multisource problems such as missing information.

Bibliography

[1] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sensing*, vol. 7, no. 11, pp. 14 680–14 707, November 2015.

[2] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu, "AID: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, July 2017.

[3] A. Loi, G. Jun, and J. Ghosh, "Active learning of hyperspectral data with spatially dependent label acquisition costs," in *IEEE Intl. Geosci. Remote Sens. Symp.*, 2009.

[4] B. Romera-Paredes and P. H. S. Torr, "An embarrassingly simple approach to zero-shot learning," in *Intl. Conf. Mach. Learn.*, vol. 37, 2015, pp. 2152–2161.

[5] D. Tuia, C. Persello, and L. Bruzzone, "Domain adaptation for the classification of remote sensing data: An overview of recent advances," *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 2, pp. 41–57, June 2016.

[6] V. Ferrari and A. Zisserman, "Learning visual attributes," in *Adv. Neural Inf. Process. Syst.*, 2007, pp. 433–440.

[7] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 1778–1785.

[8] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 453–465, March 2014.

[9] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 2927–2936.

[10] Y. Xian, B. Schiele, and Z. Akata, "Zero-shot learning — the good, the bad and the ugly," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[11] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Intl. Conf. Adv. Geogr. Inf. Syst.*, 2010, pp. 270–279.

[12] J. D. Wegner, S. Branson, D. Hall, K. Schindler, and P. Perona, "Cataloging public objects using aerial and street-level images - urban trees," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 6014–6023.

[13] S. Branson, J. D. Wegner, D. Hall, N. Lang, K. Schindler, and P. Perona, "From google maps to a fine-grained catalog of street trees," *ISPRS J. Photogram. Remote Sens.*, vol. 135, pp. 13–30, January 2018.

[14] B.-C. Kuo and D. A. Landgrebe, "A covariance estimator for small sample size classification problems and its application to feature extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 4, pp. 814–819, April 2002.

[15] B.-C. Kuo and K.-Y. Chang, "Feature extractions for small sample size classification problem," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 3, pp. 756–764, March 2007.

[16] F. Li, L. Xu, P. Siva, A. Wong, and D. A. Clausi, "Hyperspectral image classification with limited labeled training samples using enhanced ensemble learning and conditional random fields," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 8, no. 6, pp. 2427–2438, 2015.

[17] D. Tuia, F. Ratle, F. Pacifici, M. F. Kanevski, and W. J. Emery, "Active learning methods for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 7, pp. 2218–2232, July 2009.

[18] B. Demir, C. Persello, and L. Bruzzone, "Batch-mode active-learning methods for the interactive classification of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 3, pp. 1014–1031, March 2011.

[19] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1425–1438, July 2016.

[20] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg, "Baby talk: Understanding and generating simple image descriptions," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1601–1608.

[21] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *IEEE Intl. Conf. Comput. Vis.*, 2009, pp. 365–372.

[22] B. Siddiquie, R. S. Feris, and L. S. Davis, "Image ranking and retrieval based on multi-attribute queries," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 801–808.

[23] J. Liu, B. Kuipers, and S. Savarese, "Recognizing human actions by attributes," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 3337–3344.

[24] Y. Wang and G. Mori, "A discriminative latent model of object classes and attributes," in *European Conf. Comput. Vis.*, 2010, pp. 155–168.

[25] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 951–958.

[26] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[27] A. Li, Z. Lu, L. Wang, T. Xiang, and J. R. Wen, "Zero-shot scene classification for high spatial resolution remote sensing images," vol. 55, no. 7, pp. 4157–4167, July 2017.

[28] L. Bruzzone, D. F. Prieto, and S. B. Serpico, "A neural-statistical approach to multitemporal and multisource remote-sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 3, pp. 1350–1359, May 1999.

[29] M. Datcu, F. Melgani, A. Piardi, and S. B. Serpico, "Multisource data classification with dependence trees," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 3, pp. 609–617, March 2002.

[30] G. Camps-Valls, L. Gomez-Chova, J. Munoz-Mari, J. L. Rojo-Alvarez, and M. Martinez-Ramon, "Kernel-based framework for multitemporal and multisource remote sensing data classification and change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 6, pp. 1822–1835, June 2008.

[31] A. Voisin, V. A. Krylov, G. Moser, S. B. Serpico, and J. Zerubia, "Supervised classification of multisensor and multiresolution remote sensing images with a hierarchical copula-based approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 6, pp. 3346–3358, June 2014.

[32] M. Dalponte, L. Bruzzone, and D. Gianelle, "Tree species classification in the southern alps based on the fusion of very high geometrical resolution multispectral/hyperspectral images and lidar data," vol. 123, pp. 258–270, 2012.

[33] Y. Zhang, H. L. Yang, S. Prasad, E. Pasolli, J. Jung, and M. Crawford, "Ensemble multiple kernel active learning for classification of multisource remote sensing data," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 8, no. 2, pp. 845–858, February 2015.

[34] R. Vohra and K. C. Tiwari, "Object based classification using multisensor data fusion and support vector algorithm," vol. 9, no. 1, pp. 63–81, 2018.

[35] X. Xu, W. Li, Q. Ran, Q. Du, L. Gao, and B. Zhang, "Multisource remote sensing data classification based on convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 937–949, February 2018.

[36] P. Ghamisi, B. Höfle, and X. X. Zhu, "Hyperspectral and lidar data fusion using extinction profiles and deep convolutional neural network," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 10, no. 6, pp. 3011–3024, 2017.

[37] Y. Chen, C. Li, P. Ghamisi, X. Jia, and Y. Gu, "Deep fusion of remote sensing data for accurate classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 8, pp. 1253–1257, 2017.

[38] Y. Liu, S. Piramanayagam, S. T. Monteiro, and E. Saber, "Dense semantic labeling of very-high-resolution aerial imagery and lidar with fully-convolutional neural networks and higher-order crfs," in *IEEE Proc. Comput. Vis. Pattern Recog. Workshop*, 2017.

[39] S. Sukhanov, I. Tankoyeu, J. Louradour, R. Heremans, D. Trofimova, and C. Debes, "Multilevel ensembling for local climate zones classification," in *IEEE Intl. Geosci. Remote Sens. Symp.*, 2017.

[40] J. Hu, L. Mou, A. Schmitt, and X. X. Zhu, "Fusionet: A two-stream convolutional neural network for urban scene classification using polsar and hyperspectral data," in *Proc. Joint Urban Remote Sens. Event (JURSE)*, 2017.

[41] L. Pibre, M. Chaumont, G. Subsol, D. Ienco, and M. Derras, "How to deal with multi-source data for tree detection based on deep learning," in *IEEE Global Conf. Signal Inf. Process.*, 2017.

[42] F. T. Mahmoudi, F. Samadzadegan, and P. Reinartz, "Object recognition based on the context aware decision-level fusion in multiviews imagery," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 8, no. 1, pp. 12–22, January 2015.

[43] L. Gomez-Chova, D. Tuia, G. Moser, and G. Camps-Valls, "Multimodal classification of remote sensing images: A review and future directions," *Proc. IEEE*, vol. 103, no. 9, pp. 1560–1584, September 2015.

[44] D. Tuia, M. Volpi, M. Trolliet, and G. Camps-Valls, "Semisupervised manifold alignment of multimodal remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 12, pp. 7708–7720, December 2014.

[45] D. M. Gonzalez, G. Camps-Valls, and D. Tuia, "Weakly supervised alignment of multisensor images," in *IEEE Intl. Geosci. Remote Sens. Symp.*, 2015.

[46] D. Tuia, D. Marcos, and G. Camps-Valls, "Multi-temporal and multi-source remote sensing image classification by nonlinear relative normalization," *ISPRS J. Photogram. Remote Sens.*, vol. 120, pp. 1–12, October 2016.

[47] I. R. Farah, W. Boulila, K. S. Ettabaa, B. Solaiman, and M. B. Ahmed, "Interpretation of multisensor remote sensing images: Multiapproach fusion of uncertain information," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 12, pp. 4142–4152, December 2008.

[48] Y. Han, F. Bovolo, and L. Bruzzone, "Edge-based registration-noise estimation in vhr multitemporal and multisensor images," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 9, pp. 1231–1235, September 2016.

[49] D. Marcos, R. Hamid, and D. Tuia, "Geospatial correspondences for multimodal registration," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[50] M. Vakalopoulou, K. Karantzalos, N. Komodakis, and N. Paragios, "Graph-based registration, change detection, and classification in very high resolution multitemporal remote sensing data," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 9, no. 7, pp. 2940–2951, July 2016.

[51] City of Seattle, Department of Transportation. (2016, October) Seattle Street Trees. [Online]. Available: http://web6.seattle.gov/SDOT/StreetTrees/

[52] Washington State Geospatial Data Archive. (2016, October) Puget Sound Orthophotography. [Online]. Available: https://wagda.lib.washington.edu/data/type/photography/puget_sound/

[53] G. Sumbul, R. G. Cinbis, and S. Aksoy, "Fine-grained object recognition and zero-shot learning in remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 770–779, February 2018.

[54] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Intl. Conf. Learn. Represent.*, December 2014, pp. 1–41.

[56] D. Mishkin, N. Sergievskiy, and J. Matas, "Systematic evaluation of CNN advances on the ImageNet," *Comput. Vis. Image Understand.*, 2017, to appear.

[57] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, January 2014.

[58] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Intl. Conf. Mach. Learn.*, 2015, pp. 448–456.

[59] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014.

[60] University of Florida. (2016, October) 680 tree fact sheets: Trees by common name. [Online]. Available: http://hort.ifas.ufl.edu/database/trees/trees_common.shtml

[61] Natural Resources Conservation Service of the United States Department of Agriculture. (2016, October) USDA Plants. [Online]. Available: https://plants.usda.gov/java/

[62] R. Mittelman, M. Sun, B. Kuipers, and S. Savarese, "A Bayesian generative model for learning semantic hierarchies," *Frontiers in Psychology*, vol. 5, p. 417, May 2014.

[63] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feed-forward neural networks," in *Intl. Conf. Artificial Intelligence and Statistics*, 2010.