Comparison and Extension of CNNs for Analysis of Salt Marsh Images

by

Jayant Parashar

(Under the Direction of Suchi M. Bhandarkar and Brian M. Hopkinson)

Abstract

Recent advances in computer vision, most notably deep convolutional neural networks (CNNs), are exploited to identify and localize various plant species in salt marsh images. Three distinct approaches are explored that provide estimations of abundance and spatial distribution at varying levels of granularity in terms of spatial resolution. Overall, a clear trade-off is observed between the CNN estimation quality and the spatial resolution of the underlying estimation thereby offering guidance for ecological applications of CNN-based approaches to automated plant identification and localization in salt marsh images. A novel way to train neural networks for semantic image segmentation, termed as Compositional Sparse Network (CSN), is also conceptualized and tested. By leveraging the properties of dynamic expansion, interconnection richness, and sparsity, a CSN is used as the backbone for the DeepLab-V3 architecture. Since CSN is analogous to Neural Architecture Search (NAS), it is also compared to a NAS-based semantic image segmentation approach.

Index words: Deep Learning, Semantic Segmentation, Image classification, Pruning, Dynamic Expansion, Ecological monitoring, Salt marsh monitoring.

Comparison and Extension of CNNs for Analysis of Salt Marsh Images

by

Jayant Parashar

B.Sc, India, University of Delhi, 2016

A Thesis Submitted to the Graduate Faculty of the

University of Georgia in Partial Fulfillment of the Requirements for the Degree

Master of Science

Athens, Georgia

2020

Comparison and Extension of CNNs for Analysis of Salt Marsh Images

by

Jayant Parashar

Major Professors: Suchendra M. Bhandarkar
Brian M. Hopkinson

Committee: Sheng Li
Frederick Maier

Electronic Version Approved:

Ron Walcott
Interim Dean of the Graduate School
The University of Georgia
August 2020

Acknowledgments

This research was supported in part by the US National Science Foundation GCE-LTER grant (OCE-1832178), to Dr. Bhandarkar and Dr. Hopkinson.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Literature Review

1.1 CNNs in Ecological Research

Ecological studies are often limited in spatial and temporal scale by the time needed to characterize the distribution and abundance of the constituent species. For example, most assessments of plant and invertebrate species distributions are limited to analysis of abundance or presence/absence within small areas of the study site (termed as *quadrats*). The quadrats typically comprise $< 1\%$ of the total study area, due to the intensive effort required to manually analyze these ecological systems [51]. The combination of computer vision and machine learning tools offer the possibility of automating and accelerating this work, making it possible to increase the spatial and temporal resolution of ecological surveys. In particular, deep learning approaches are rapidly gaining popularity for these tasks since they provide a unified computer vision and machine learning framework whose performance typically exceeds that of previous approaches that use predetermined hand-crafted features [9].

In Chaper 2, we examine deep learning approaches for identification, enumeration, and spatial localization of plant species in salt marsh ecosystems. Salt marshes are highly productive, inter-tidal marine habitats found along protected coastlines or behind barrier islands spanning temperate and subpolar re-

gions on the earth's surface [35]. Salt marsh sediments store high densities of organic carbon making them important *blue carbon* ecosystems, in which carbon is sequestered that would otherwise be released to the atmosphere increasing atmospheric CO_2 concentrations [34]. Few species inhabit the salt marsh ecosystem due to its harsh conditions. However, on account of the ecosystem's high productivity, the resident species capable of surviving salt marsh conditions often exhibit high abundance. Given their low biodiversity, salt marshes have long served as model ecosystems that are amenable to both experimental and observational work [7].

Salt marsh plant communities on the east coast of the United States are typically dominated by grasses in the genus *Spartina*, especially at lower marsh elevations as only this genus is capable of tolerating frequent flooding with salt water [38]. At higher marsh elevations, several additional species are also found, many of which are succulents or otherwise adapted to handle the harsh salt marsh conditions. The abundance and distribution of these resident species is commonly assessed using several semi-quantitative methods, employed either in real time in the field or on archived images. One approach is to simply indicate within a small quadrat (e.g. 0.25 m $\times$ 0.25 m) which plant species are present and which are absent. An alternative approach is to estimate the *percent cover* within the quadrat, i.e., the percentage of space occupied by each plant species or substrate, by randomly choosing points (25–100) within a small area (on the ground or in the image) and identifying the resident plant species (or substrate) at the chosen point. In this paper, we explore and assess various deep learning approaches to automate the aforementioned tasks on input images of salt marshes. Going beyond what is typically done in ecological studies due to time constraints, we also assess the ability of CNN architectures to perform semantic segmentation of salt marsh images. Semantic image segmentation potentially provides more accurate estimates of species abundance and the spatial distribution of plants at a much finer level of granularity defined by spatial resolution. In this work, the *DeepLab-V3* CNN architecture is employed for the semantic image segmentation task. As is typically encountered in studies of most ecological systems, the salt marsh images are characterized by fine-grained interleaving of classes, ambiguous class boundaries, and wide-variations in lighting and

viewing perspective that complicate automated image analysis procedures, particularly those pertaining to semantic image segmentation.

We specifically explore multi-layer convolutional neural network (CNN) architectures such as the *ResNet* [21], *PyramidNet* [18], *residual attention network (RAN)*, *DenseNet* [23], *ResNext* [57], and *Inception-V3* [46] in the context of multi-label classification. These CNN architectures are also employed for the percent cover computation task, which is formulated as the more simple image classification problem. In both presence absence and percent cover approaches, we found that that ResNext and Dual Path networks performed quite well. Both these approaches are multi-path approaches where interconnection richness is leveraged to produce better results. This idea of interconnection richness led us to question the reasons behind success of modern deep learning architectures. Therefore, we continued our research on the formulation of general-purpose CNN architectures based on these insights.

1.2 Searching for General-purpose CNNs

In deep learning, we observe that the search for general-purpose CNN architecture design can be differentiated based on the underlying problem formulation. We believe that such differentiation only helps us create general models if we can find a way to segregate high-level features based on their underlying neuronal structures. Since artificial neural networks are a black box, the problem-specific architecture search is tantamount to merely solving specific problems. A manual search of architectures for specific problems does not contribute to efficient out-of-distribution (O.O.D) generalization and general intelligence. We should be able to find deep learning architectures that, except for a few input and output layers, share exactly the same structure. One efficient approach to architecture search is termed as *Neural Architecture Search* (NAS) [33]. Although NAS approaches are efficient in their search procedure, their search space is limited to determining which stacking of convolution filters is optimal. NAS approaches typically usually focus their search on performing minor CNN enhancements to yield the optimal architecture. In typical NAS approaches, the CNN architectures are usually modeled as a rigid three-level hierarchy comprising

of (a) CNN channels, (b) blocks, and (c) network architecture. Being forced to act within the bounds of this hierarchy, NAS approaches have only produced incremental improvement in CNN performance.

We propose an alternative approach that tackles the above limitations. First, we recommend that future deep learning research focus on the study of inter-connectivity of individual neurons, and how features are learned and composed [10]. This knowledge, will allow us to confidently construct general-purpose architectures that are capable of generalization to a wide array of problems. Based on a hypothesis of how neural networks generalize using interconnection richness, we propose a new paradigm of training neural networks that performs, what we term as, a *pseudo* neural architecture search, essentially creating more complex architectures from simpler ones. Our primary vision underlying this approach is to leverage the rich interconnections within shallow sub-networks. Many intuitive ideas behind CNN architecture design have followed this simple heuristic of feature composition such as gradual down-sampling of the image, dilated convolution [11], and design of residual networks [21]. Recent work in demystifying the reasons for the success of residual networks has pointed towards their capacity to behave like ensembles of relatively shallow networks [49].

We believe that the proposed pseudo NAS process, termed as the *Compositional Sparse Network* (CSN), should be performed on a carefully chosen *base network* such as the *Graph Neural Network* (GNN) [59] or fully connected dense neural network (FCNN). In this work, however, we choose the 3×3-size kernel, the most basic CNN unit, as the minimalist base network to construct a modular architecture capable of composing features on its own, without the external help of down-sampling or dilated convolution. Moreover, the current movement towards general-purpose networks is not motivated by the need to tackle real-world data sets, but by novel problem-solving approaches such as meta-learning, few-shot learning, and continual learning [13]. These problem-solving approaches usually use toy data sets and toy networks of small size that have not been tested on larger real-world data sets and often do not generalize well in many cases. This begs the question: why can we not implement our theories of general-purpose architectures on common problems such as semantic image segmentation? Since, at an intuitive level, we can safely claim that a general-purpose network would, most likely, significantly outperform a

custom designed CNN, why do we not combine our hypotheses for O.O.D generalization [28], derived from problem solving approaches such as meta-learning, few-shot learning, and continual learning, and apply them to specific problems such as semantic image segmentation to test their efficacy [4], [31]. With that in mind, we have formulated a CSN-based approach to semantic image segmentation.

Deriving inspiration from Neuroscience [40] and continual learning-based approaches [31], the CSN performs a pseudo NAS procedure that combines our understanding of three basic principles required for incorporating general intelligence in neural networks: (a) *interconnection richness hypothesis*, (b) *dynamic expansion*, and (c) *sparsity*. The interconnection richness hypothesis is based on the idea that interconnections between local sub-networks and the respective interconnections between neurons in local sub-networks are key to the design of an effective architecture. The implementation of interconnection richness within a CSN-based framework is inspired by the structure of neocortex in the human brain [40]. To create a hierarchy of sub-networks we facilitate the dynamic expansion of repeating units. This idea is based on the intuition that a network must be able to increase its size based on the complexity of the problem. Optimization-based dynamic expansion is short-sighted and leads to a cutoff depth that is much lower than the optimal depth needed for a specific problem[24]. Therefore we formulate a manual external dynamic expansion coupled with network pruning to find general architectural solutions [3]. While expanding a network dynamically, we should also be able to simultaneously decrease its complexity to find the most essential features for the purpose of feature composition. To this end, network pruning is shown to increase the performance of a given network despite deletion of 60%-90% of its neurons [16].

Chapter 2

Estimation of Abundance and Distribution of Salt Marsh Plants from Images Using Deep Learning[1]

[1]Jayant Parashar*, Suchendra Bhandarkar, Jacob Simon, Brian Hopkinson, Steven Pennings. 2020. Submitted to the International Conference on Pattern Recognition, 06/15/2020.

2.1 Abstract

Recent advances in computer vision and machine learning, most notably deep convolutional neural networks (CNNs), are exploited to identify and localize various plant species in salt marsh images. Three different approaches are explored that provide estimations of abundance and spatial distribution at varying levels of granularity in terms of spatial resolution. In the coarsest-grained approach, CNNs are tasked with identifying which of six plant species are present/absent in large patches within the salt marsh images. CNNs with diverse topological properties and attention mechanisms are shown capable of providing accurate estimations with $> 90\%$ precision and recall in the case of the more abundant plant species whereas the performance declines for less common plant species. Estimation of *percent cover* of each plant species is performed at a finer spatial resolution, where smaller image patches are extracted and the CNNs tasked with identifying the plant species or substrate at the center of the image patch. For the percent cover estimation task, the CNNs are observed to exhibit a performance profile similar to that for the presence/absence estimation task, but with an ≈ 5–10% reduction in precision and recall. Finally, fine-grained estimation of the spatial distribution of the various plant species is performed via semantic segmentation. The *DeepLab-V3* semantic segmentation architecture is observed to provide very accurate estimations for abundant plant species; however, a significant degradation in performance is observed in the case of less abundant plant species and, in extreme cases, rare plant classes are seen to be ignored entirely. Overall, a clear trade-off is observed between the CNN estimation quality and the spatial resolution of the underlying estimation thereby offering guidance for ecological applications of CNN-based approaches to automated plant identification and localization in salt marsh images.

Keywords: Salt marsh monitoring, convolutional neural networks, network topology, attention mechanism, deep learning, ecological monitoring

2.2 Introduction

Ecological studies are often limited in spatial and temporal scale by the time needed to characterize the distribution and abundance of the constituent species. For example, most assessments of plant and invertebrate species distributions are limited to analysis of abundance or presence/absence within small areas of the study site (termed as *quadrats*). The quadrats typically comprise $< 1\%$ of the total study area, due to the intensive effort required to manually analyze these ecological systems [51]. The combination of computer vision and machine learning tools offer the possibility of automating and accelerating this work, making it possible to increase the spatial and temporal resolution of ecological surveys. In particular, deep learning approaches are rapidly gaining popularity for these tasks since they provide a unified computer vision and machine learning framework whose performance typically exceeds that of previous approaches that use predetermined hand-crafted features [9].

In this paper, we examine deep learning approaches for identification, enumeration, and spatial localization of plant species in salt marsh ecosystems. Salt marshes are highly productive, inter-tidal marine habitats found along protected coastlines or behind barrier islands spanning temperate and subpolar regions on the earth's surface [35]. Salt marsh sediments store high densities of organic carbon making them important *blue carbon* ecosystems, in which carbon is sequestered that would otherwise be released to the atmosphere increasing atmospheric CO_2 concentrations [34]. Few species inhabit the salt marsh ecosystem due to its harsh conditions. However, on account of the ecosystem's high productivity, the resident species capable of surviving salt marsh conditions often exhibit high abundance. Given their low biodiversity, salt marshes have long served as model ecosystems that are amenable to both experimental and observational work [7].

Salt marsh plant communities on the east coast of the United States are typically dominated by grasses in the genus *Spartina*, especially at lower marsh elevations as only this genus is capable of tolerating frequent flooding with salt water [38]. At higher marsh elevations, several additional species are also found, many of which are succulents or otherwise adapted to handle the harsh salt marsh conditions. The

abundance and distribution of these resident species is commonly assessed using several semi-quantitative methods, employed either in real time in the field or on archived images. One approach is to simply indicate within a small quadrat (e.g. 0.25 m $\times$ 0.25 m) which plant species are present and which are absent. An alternative approach is to estimate the *percent cover* within the quadrat, i.e., the percentage of space occupied by each plant species or substrate, by randomly choosing points (25–100) within a small area (on the ground or in the image) and identifying the resident plant species (or substrate) at the chosen point. In this paper, we explore and assess various deep learning approaches to automate the aforementioned tasks on input images of salt marshes. We specifically explore multi-layer convolutional neural network (CNN) architectures such as the *ResNet* [21], *PyramidNet* [18], *residual attention network (RAN)*, *DenseNet* [23], *ResNext* [57], and *Inception-V3* [46] in the context of multi-label classification. These CNN architectures are also employed for the percent cover computation task, which is formulated as the more simple image classification problem.

Going beyond what is typically done in ecological studies due to time constraints, we also assess the ability of CNN architectures to perform semantic segmentation of salt marsh images. Semantic image segmentation potentially provides more accurate estimates of species abundance and the spatial distribution of plants at a much finer level of granularity defined by spatial resolution. In this paper, the *DeepLab-V3* CNN architecture is employed for the semantic image segmentation task. As is typically encountered in studies of most ecological systems, the salt marsh images are characterized by fine-grained interleaving of classes, ambiguous class boundaries, and wide-variations in lighting and viewing perspective that complicate automated image analysis procedures, particularly those pertaining to semantic image segmentation.

2.3 Background

2.3.1 Deep Learning Applications in Ecology

Previous applications of computer vision and machine learning techniques to the analysis of ecosystem imagery has progressed from traditional pipelines that extract predetermined, hand-crafted features, followed

Figure 2.1: Example images of marsh plant species (from left to right and top to bottom): *Sarcocornia, Spartina, Limonium, Borrichia, Batis* and *Juncus*.

by application of traditional classifiers such as support vector machines (SVMs) to employing end-to-end deep neural networks (DNNs) or deep learning (DL) methods. The pioneering work of Beijbom et al. [5], [6], focusing on coral reef surveys, is an excellent example of the traditional approach to automated classification of ecosystem images. Employing a maximum response filter bank in conjunction with a multiscale patch/texton dictionary to characterize the features in underwater coral reef images, Beijbom et al. [5] use a traditional SVM-based classifier to categorize the image patches as belonging to various coral organism classes. Beijbom et al. [6] also outline the many challenges unique to the task of automated analysis of ecological images such as extreme variations in the size, color, shape, and texture of each of the taxa, the organic and ambiguous nature of the class boundaries and significant alterations in ambient lighting and image colors. In the ecological remote sensing literature, classification approaches have focused almost exclusively on pixel-level spectral information ignoring spatial context, although there have been some

notable exceptions [22]. However, classifiers based entirely or primarily on pixel-level spectral data are less relevant to *local* ecosystem images which are typically acquired with consumer-grade RGB cameras.

In recent times, CNNs (or ConvNets) and related deep neural networks (DNNs) have revolutionized computer vision, especially with regard to image segmentation, feature extraction and classification, and object detection and recognition [27], [29]. The superior performance of CNN- and related DNN-based approaches has led to their rapid adoption in ecological research [9], [52]. Brodrick et al. [9] argue that CNNs may become essential tools for ecologists due to their power, generality, and relative ease of use. CNNs have shown notable success in detection of animals, such as hummingbirds [53] and shorebirds [8]. In several remote sensing applications, CNNs have led to dramatically improved methods for specific tasks such as to automatically identify and inventory termite mounds [9] and various species of trees [2], [54].

In addition to very specific tasks, progress has also been made on using CNNs to provide a broad-scale overview of community composition. Williams et al. [56] have employed CNNs to assess the abundance of major taxa and substrates on coral reefs, achieving classification accuracies similar to those attained by human annotators. In contrast to traditional approaches to ecosystem image analysis [5], [6] that employ complex feature extraction and classification algorithms requiring extensive knowledge of computer vision and machine learning, a typical CNN-based computational pipeline provides an integrated image analysis framework by leveraging existing, pre-trained CNNs wherein the feature extractors and classifiers have been automatically learned from training data. Their user-friendliness, superior performance, and the fact that they require minimal background in computer vision and machine learning suggest that CNNs will be widely adopted to accelerate ecological research in the near future.

2.3.2 Convolutional Neural Networks (CNNs)

CNNs have become a standard tool for several machine vision tasks such as image classification, semantic image segmentation and automated image captioning. The various CNN architectures described in the research literature differ from each other based on their topological properties across multiple dimen-

sions such as, the type of convolution operation performed, network depth, spatial dimensions of the network layers, and the width and design of multiple network pathways [25]. The three-layer *ConvNet* with backpropagation-based weight learning proposed by LeCun et al. [30] for automated recognition of hand-written zip codes represents one of the early successful applications of the CNN architecture to an important real-world problem.

Inception CNN The *Inception* CNN employs the principles of variability and modularity to deal with increasing model size and computational costs associated with scaling up of CNNs to address real-world computer vision problems [46]. A key feature of the *Inception* CNN is the introduction of inception layers comprising of multiple-size convolutional filter kernels. Earlier versions of the *Inception* CNN used small-size convolutional filters arguing their computational efficiency. Subsequent versions of the *Inception* CNN create a much deeper network by combining blocks of varying filter sizes using split, transform and merge strategies [46]. These strategies ensure a multi-path flow of information with varying filter sizes allowing for effective design of deeper CNNs. The varying filter sizes capture spatial information at multiple scale which is subsequently combined within a single block using 1×1 convolutional filters [32]. Additionally, the *Inception* CNN uses an auxiliary classifier to deal with the problem of degradation of input between successive network layers [45].

Residual Learning CNN Construction of deeper CNNs cause degradation of the input between successive network layers leading to the *vanishing gradient problem* that severely limits backpropagation learning [45]. Residual learning CNNs, termed *ResNets*, mitigate this problem by introducing *skip* connections between CNN layers [21]; an idea that represents a paradigm shift in CNN architecture design. Various versions of the *ResNet* have been shown to yield highly competitive performance across a variety of datasets [47]. *ResNets* do not alter the topology of the network connections, rather they simply combine the outputs of alternating layers making it possible to stack as many as 200 layers without overfitting or having to deal with the vanishing gradient problem.

Densely Connected CNN (DenseNet) Densely Connected CNNs, termed *DenseNets* [23], unlike *ResNets*, concatenate the output of every layer with the outputs of all previous layers in a given block. *DenseNets* have compelling advantages in that they alleviate the vanishing gradient problem and strengthen feature propagation. Although the number of direct connections increases quadratically in the number of layers, *DenseNets* reduce the number of parameters substantially by encouraging feature reuse while simultaneously addressing the problem of input degradation between successive layers.

Dual Path Network (DPN) The *dual path network* (DPN) unifies the *ResNet* and *DenseNet* architectures to generate a *higher-order recurrent neural network* (HORNN) [12]. The HORNN architecture introduces recurrent connections between non-neighboring units based on an order hyper-parameter [43]. Since *ResNets* [21] allow for efficient feature reuse whereas *DenseNets* [23] are particularly effective at feature discovery, a multi-path network comprising of both *DenseNets* and *ResNets* allows one to achieve the best of both worlds.

ResNext and Pyramid Networks The *ResNext* [57] is a simple, highly modularized network architecture that is constructed by repeating a building block that aggregates a set of transformations with the same topology. In a sense, the *ResNext* combines the best of the Inception [46] and *ResNext* [21] architectures. The *ResNext* creates blocks with multiple pathways using group convolutions [27] which are later combined using 1×1 convolutional filters [32]. The pyramidal network *PyramidNet* [18] is an enchancement of the *ResNet* wherein the dimensions of the feature map are increased gradually following an arithmetic progression (*additive PyramidNet*) or geometric progression (*multiplicative PyramidNet*). The *PyramidNet* is shown to perform better than the *ResNet* on image classification tasks since it circumvents loss of useful information [18].

Semantic Segmentation The aforementioned CNNs are used primarily for image classification tasks where an entire image is classified as belonging to a certain category. However, for more fine-grained analysis, one needs to perform semantic segmentation of the image where each image pixel is classified as

belonging to a certain category. In this paper, we employ the *DeepLab-V3* architecture which leverages *atrous convolution* and *atrous pyramid spatial pooling* to generate pixel-wise labels for semantic image segmentation [11]. Atrous convolution is a special type of convolution which acts over a dilated sub-grid of the input [11].

2.3.3 Attention Mechanisms

When training a CNN model, it is imperative that the model be able to focus on important portions of the image while discarding the irrelevant portions in order to speed up the training process. One way of accomplishing this is via incorporation of *attention mechanisms* within the training procedure. In this work, we employed the *residual attention network* (RAN) architecture, which creates stacks of trunk and mask branches where the mask branches perform downsampling (i.e., convolution) and upsampling (i.e., deconvolution) to create attention masks [50]. In contrast, the trunk branch performs only convolution. Upsampling effectively recreates the image dimensions after convolution has reduced them. The *RAN* stacks are trained in an end-to-end fashion.

2.4 Datasets

Overhead images of a roughly rectangular section of a salt marsh on Sapelo Island, Georgia, USA were collected in June 2014 using a consumer grade DSLR camera (Nikon D7100) with a wide-angle lens (24 mm). The camera was attached ≈1.5 m above the ground to a wheeled platform that was pulled across the marsh while high-resolution images (4000 × 6000 pixels) were continuously acquired at a rate of 1 Hz (1 frame/sec). After the camera had traversed ≈ 20–40 m the mobile platform was stopped, and the images acquired over that distance were deemed to comprise a *row*. The imaging platform was then moved ≈ 1 m perpendicular to the previous row and pulled again across the marsh to image an adjacent row. A section of marsh covering 80 rows was imaged in this manner moving from higher marsh elevation (with a low row number) to lower marsh elevations (with high row numbers).

Six plant species are commonly found in salt marshes on Sapelo Island and all six were present in the images: *Spartina alterifora, Juncus roemerianus, Batis maritima, Sarcocornia spp., Borrichia frutescens,* and *Limonium carolineanum*, all of which are subsequently identified by their genus (Fig. 2.1). The only plant species found at low marsh elevations is *Spartina*, but all species are found at high marsh elevations with *Spartina* gradually disappearing from the assemblage at the highest elevations giving way to a diverse mix of the remaining species. *Spartina* is a medium to tall grass (height: 0.3–2 m) with wide blades that emerge from a thick stem. The blades appear dark green to light green depending on the time of the day and dead blades attached to the stems are not uncommon. *Juncus* is a tall rush (height $\approx$ 1 m) with smooth cylindrical leaves that are gray-green in color. *Sarcocornia* is a low growing succulent plant with branching stems that lack leaves. Most of the stems are green but shade into red. *Batis* is a dense, succulent shrub with alternate, green to yellow leaves growing to $\approx$ 0.3 m tall in the imaged section of the marsh. *Borrichia* is a shrub with characteristic, grey-green oval leaves and bright yellow flowers at the end of each branch throughout most of the summer. We created three separate data sets for training, validation, and testing for each of the tasks, i.e., presence/absence determination, percent cover computation and semantic image segmentation as shown in Fig. 2.2.

2.4.1 Presence/Absence Determination

In the presence/absence determination task, the input images were divided into 15 sections (3 rows $\times$ 5 columns) as shown in Fig. 2.2. An expert human annotator then performed multi-label classification, i.e., delineation of all the plants in each image section for randomly selected images from the Sapelo Island marsh image data set. The multi-label classification dealt with seven classes, the six aforementioned plant classes and one *background* class. The most dominant class encountered in the presence/absence task was *Spartina* which was present in more than half of the image sections. In contrast, *Batis* and *Sarcocornia* were present in $\approx$ 17% of the image sections whereas *Limonium, Batis* and *Juncus* were observed to be rare classes, each accounting for $\approx$ 4% of the image sections.

2.4.2 Percent Cover Computation

Percent cover, the fraction of space occupied by a particular plant species when viewed from overhead, provides a more refined abundance metric than presence/absence. In the percent cover computation task, 25–50 points were randomly selected in an image. An image patch (512 × 512 pixels) surrounding each point was presented to an expert human annotator who performed single-label classification of the patch, i.e., labeled the patch based on the class present at the selected point (Fig. 2.2). There were 9 classes under consideration for this task: the aforementioned six plant classes, *Soil*, *Other*, and *Unknown*. The class *Other* was used to indicate an identifiable entity that did not belong to one of the six plant classes or *Soil*, such as *invertebrates* (crabs, snails, etc.) or portions of the imaging platform captured by the camera. The *Unknown* class denotes a situation where the class underlying the selected point was not identifiable, which typically occurred when the image section was out of focus or heavily shadowed. The dominant classes encountered in this task were *Spartina, Batis* and *Sarcocornia* which collectively accounted for ≈ 85% cover in approximately equal proportion. *Borrichia* and *Juncus* were deemed rare classes, each with ≈ 6% cover and *Limonium* the rarest class with ≈ 3% cover.

2.4.3 Semantic Image Segmentation

The goal of semantic image segmentation is to classify each pixel in an image into one of predetermined categories. To generate training data for the semantic image segmentation task, we used a superpixel labeling tool described in [26]. Each pixel was classified into one of nine classes, which were slightly different than those used for percent cover cpmputation: the aforementioned six plant classes, *dead Spartina* (due to its common occurrence and substantially different appearance compared to *live Spartina*), background (which accounts for the classes *Soil* and *Unknown* in percent cover computation), and *Other* (which accounts for *invertebrates*, and portions of imaging platform captured by the camera).

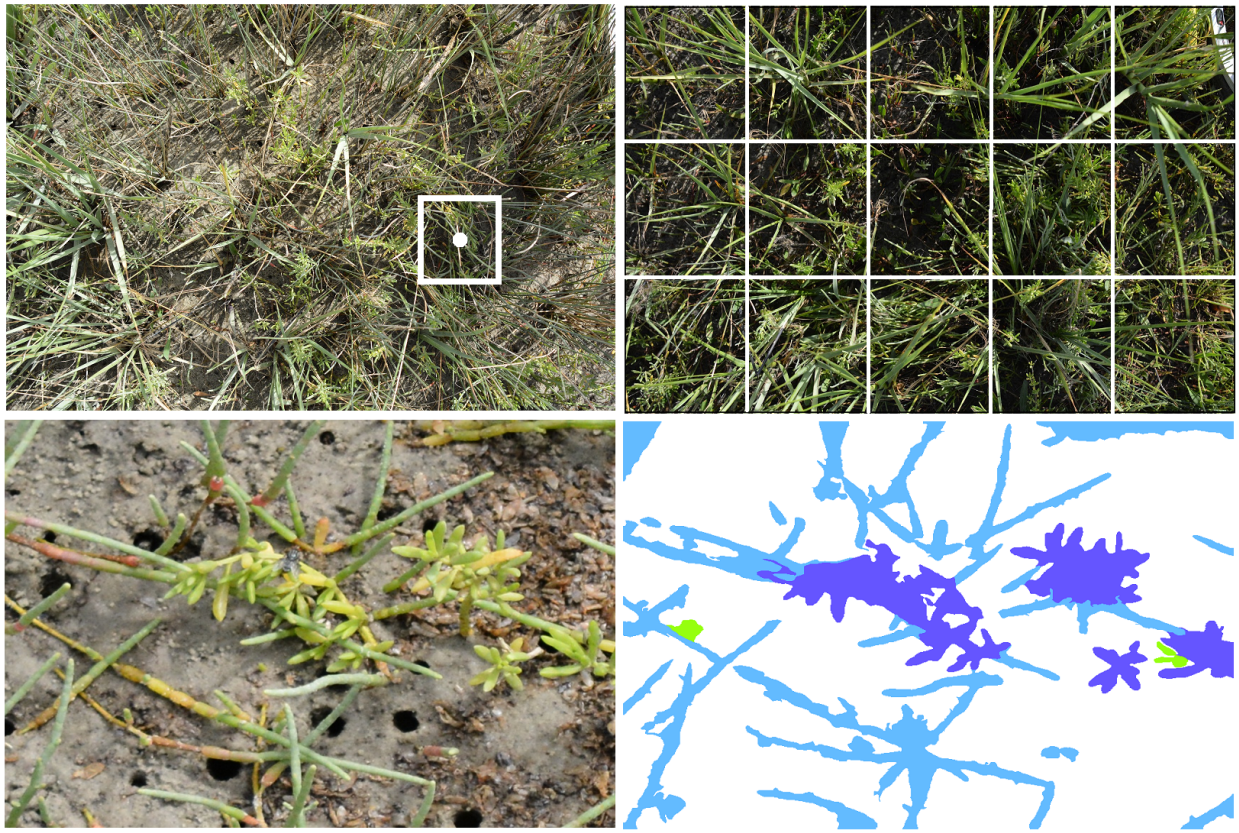

Figure 2.2: Image analysis tasks from left to right and top to bottom: percent cover computation, presence/absence determination, semantic image segmentation and the corresponding segmentation masks.

2.5 Experimental Results

2.5.1 Performance Evaluation Metrics

Precision, *recall* and *f-1* scores were used as evaluation metrics to compare the various deep learning models both in terms of overall performance and performance on specific classes. For overall performance assessment, both micro- and macro-averaged metrics were computed. Macro-averaged metrics were computed by first computing the precision, recall and *f-1* score metrics for each individual class and then averaging these metrics across all classes. In contrast, micro-averaged metrics were computed by summing the true positives, false positives, false negatives, and true negatives across the entire data set regardless of class and then computing precision, recall, and *f-1* score metrics. In the case of single-label classification (as is done in percent cover computation and semantic image segmentation) micro-averaged precision, recall, and *f-1* score metrics are equal to overall accuracy.

2.5.2 Evaluation of Presence/Absence Computation

Approximately 24,000 salt marsh image sections (from 1600 images) were manually labeled for presence/absence of the six plant species. Images in the manually annotated data set were split into training (60%), testing (20%) and validation (20%) datasets. CNNs were trained for multi-label classification on the training data set using the Adam optimizer and binary cross entropy loss function. We initialized the model weights with values pre-trained on *ImageNet* [14] for all models and trained the weights until the loss value stopped declining on the validation data set.

After an initial examination of the performance of a trial model (*ResNet*) on the task, we found the classifier performed reasonably well on most classes, but had trouble identifying *Juncus* and falsely predicted the presence of *Sarcocornia* (and occasionally *Batis*) in low-elevation marsh regions where only *Spartina* was present. *Juncus* was rare and an examination of the classifier showed that *Juncus* was being misclassified as the more common *Spartina* grass. In an attempt to overcome this issue, additional images

containing *Juncus* were identified and manually labeled to increase the representation of *Juncus* in the training data set. To address the false positives associated with *Sarcocornia* in the low-elevation marsh regions, additional low-elevation marsh images containing only *Spartina* were manually classified and added to the data sets. The addition of targeted training data helped improve the performance of the trial model (*ResNet*) and we proceeded to evaluate the performance of the remainder of the CNN architectures under consideration.

The seven CNN architectures that we assessed were generally observed to yield similar performance with micro-averaged precision, recall, and *f-1* scores, all exceeding 0.9 (Table 2.1). The macro-averaged metrics were somewhat lower due to poorer performance on the rarer plant species (Table 2.2). The relative performance of individual models was generally consistent across micro- and macro-averaged metrics with *ResNext* yielding the best precision and *f-1* score values and *DPN* the highest recall values. Unlike the other CNN architectures, *DPN* and *ResNext* both use a multi-path strategy which potentially helps to generate complex feature combinations.

The performance of the CNN architectures on individual classes was observed to vary significantly based on the relative abundance of the plant species in the data set (Figs. 2.3 and 2.4). *Spartina*, the most abundant species, was classified extremely accurately with precision and recall values exceeding 0.95. The recall value for *Sarcocornia* was also exceedingly high (> 0.95) but the precision was notably lower at ≈ 0.9 due to the presence of false positives in the low-elevation marsh regions despite attempts at improvement. Although *Sarcocornia* is commonly present in most image sections, it is often not very abundant with only a few stems present in a given image section, which potentially contributes to the difficulty in achieving high precision values for this class. The quality of predictions for the remaining classes (*Batis, Borrichia, Limonium*, and *Juncus*) was generally observed to follow their occurrence frequency in the manually annotated data sets. While the performance of different CNN architectures at the class level was generally similar, both the *DPN* and *ResNext* exhibited uniquely high individual precision and recall values for multiple plant categories (Figs. 2.3 and 2.4). The recall values for DPN for *Limonium* and *Juncus* were at least 5% higher than those of all other CNN models. The precision value for *Limonium* was the highest

in the case of the *ResNext*, showing an 8% difference from the next best CNN architecture. The *RAN* performed the worst overall in terms of the *f-1* score. However, the *RAN* was observed to yield a precision value that almost matched that of the *ResNext*. The attention mechanism used in the *RAN* did not seem to hold any advantages for this data set and task.

Table 2.1: Presence/absence micro-averaged results

CNN type	***Micro precision***	***Micro recall***	***Micro f-1 score***
ResNet101	0.918	0.932	0.925
DenseNet121	0.912	0.930	0.921
DPN92	0.906	**0.938**	0.922
ResNext101	**0.928**	0.931	**0.930**
Inception	0.910	0.923	0.916
RAN	0.924	0.889	0.906
PyramidNet101	0.911	0.923	0.917

Table 2.2: Presence/absence macro-averaged results

CNN type	***Macro precision***	***Macro recall***	***Macro f-1 score***
ResNet101	0.844	0.872	0.858
DenseNet121	0.838	0.868	0.853
DPN92	0.827	**0.884**	0.855
ResNext101	**0.861**	0.867	**0.864**
Inception	0.834	0.845	0.839
RAN	0.848	0.761	0.802
PyramidNet101	0.820	0.840	0.830

2.5.3 Evaluation of Percent Cover Computation

To assess the ability of CNNs to automate percent cover computation, approximately 15,000 points were manually labeled in $\approx$ 500 images randomly sampled from the salt marsh image data set. The manually annotated data set was split, at the level of an individual image, into training (60%), validation (20%), and test (20%) data sets. The CNN models were trained in a manner similar to that described for the presence/absence computation task except that a multi-class cross entropy loss function was used. Table 2.3

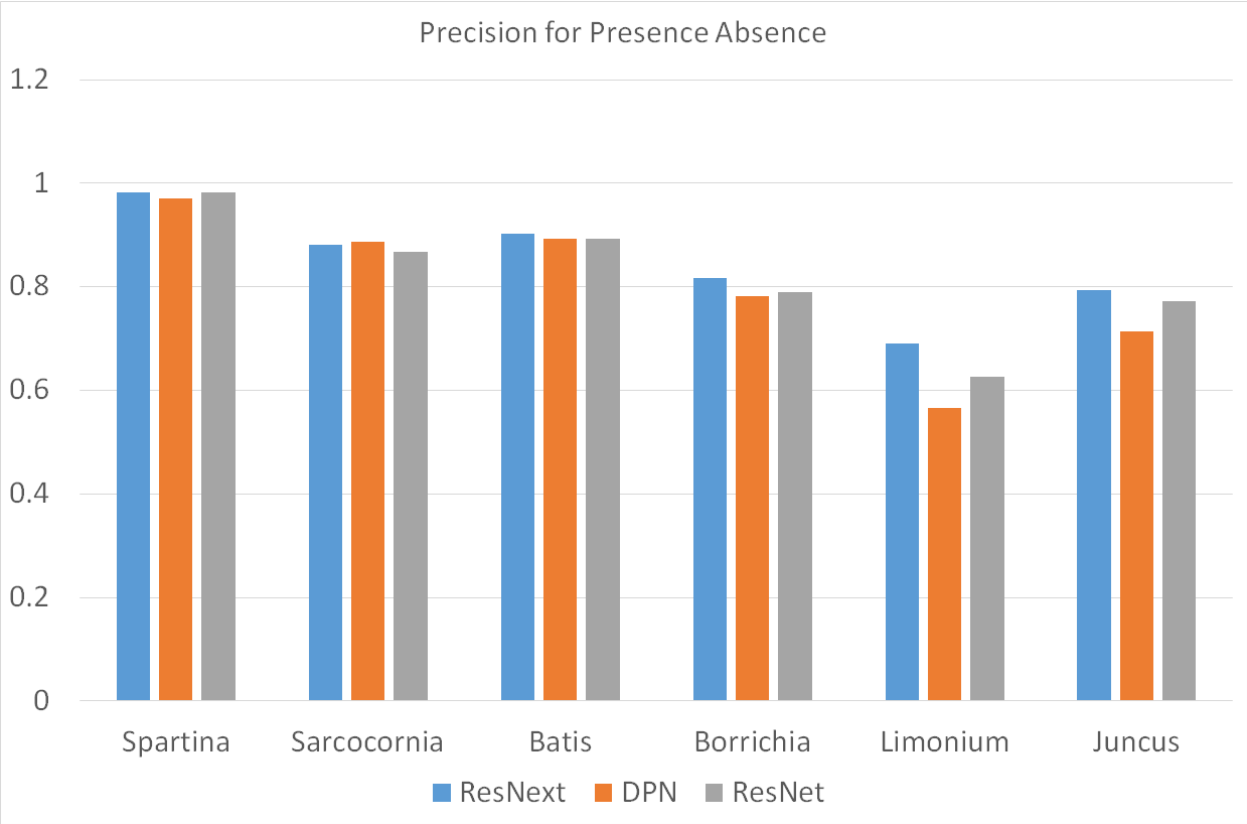

Figure 2.3: Precision values for individual classes in presence/absence computation for (a) *ResNext*, (b) *Dual Path Network*, and (c) *ResNet*

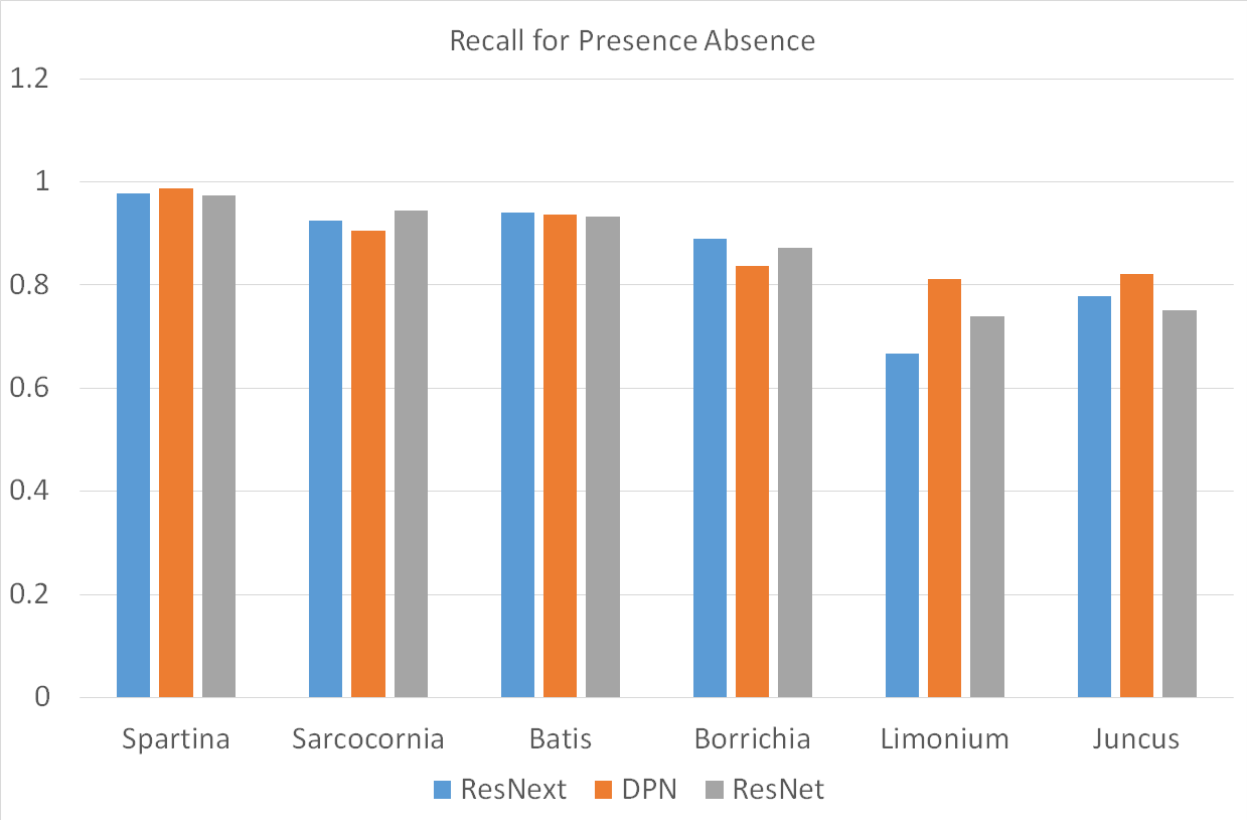

Figure 2.4: Recall values for individual classes in presence/absence computation for (a) *ResNext*, (b) *Dual Path Network*, and (c) *ResNet*

reports the results of the same seven CNN architectures used in presence/absence computation, for the percent cover computation task. Since percent cover is a multi-class, single-label classification problem, the micro-averaged precision and recall values are equal and denoted by the term *accuracy*. The values of the macro-averaged precision, recall, and *f-1* scores were observed to be ≈ 0.10 lower for percent cover computation compared to presence/absence computation for each of the seven CNN architectures.

A significant challenge for the percent cover computation is the fine-grained interleaving of classes in salt marsh images. Image patches extracted for percent cover computation often contain multiple plant species and the classifier must learn to classify the plant observed in the center of the image patch. Recognizing this issue, we tested an attention-based CNN, i.e., the *RAN*, hypothesizing that *RAN* would learn to focus attention on the center of the image patch for the purpose of classification while using the outer regions of the patch for context. However, the performance of the *RAN* was generally observed to be inferior to that of the other CNN architectures. The initial patch size used (512×512 pixels) was relatively large so we attempted to reduce the size of the patch to 256×256 pixels to limit the number of classes present in image patches. However, the performance of all the CNN architectures, in terms of recall, dropped precipitously by ≈ 0.20 on the smaller patches across all classes. This was likely because the smaller patches did not provide sufficient context that was critical for accurate classification, such as overall leaf shape, instead, forcing the classifier to rely on small-scale texture and color features. As shown in Table 2.3, the *ResNext* was observed to achieve the highest overall accuracy and *f-1* score closely followed by the *PyramidNet* and *DPN*. The confusion matrix in Fig. 2.5 shows that even the best performing *ResNext* CNN had particular difficulty recognizing lower abundance classes such as *Juncus, Limnonium*, and, in some cases, *Borrichia*.

2.5.4 Evaluation of Semantic Image Segmentation

Semantic image segmentation is not commonly employed in ecological research due to the labour intensive nature of labelling entire images at the pixel level. However, automated semantic segmentation offers potentially unprecedented spatial resolution in the field of ecology allowing novel insights into spatial relationships amongst organisms as well as computation of a more accurate abundance metric. We

Table 2.3: Results of percent cover computation

CNN type	*precision*	*recall*	*f-1 score*	*Accuracy*
ResNet101	0.742	0.700	0.720	0.833
DenseNet121	0.717	0.668	0.692	0.846
DPN92	0.743	0.732	0.737	0.843
ResNext101	**0.767**	0.736	**0.751**	**0.857**
Inception	0.738	0.672	0.703	0.833
RAN	0.713	0.618	0.662	0.851
PyramidNet101	0.751	**0.743**	0.748	0.844

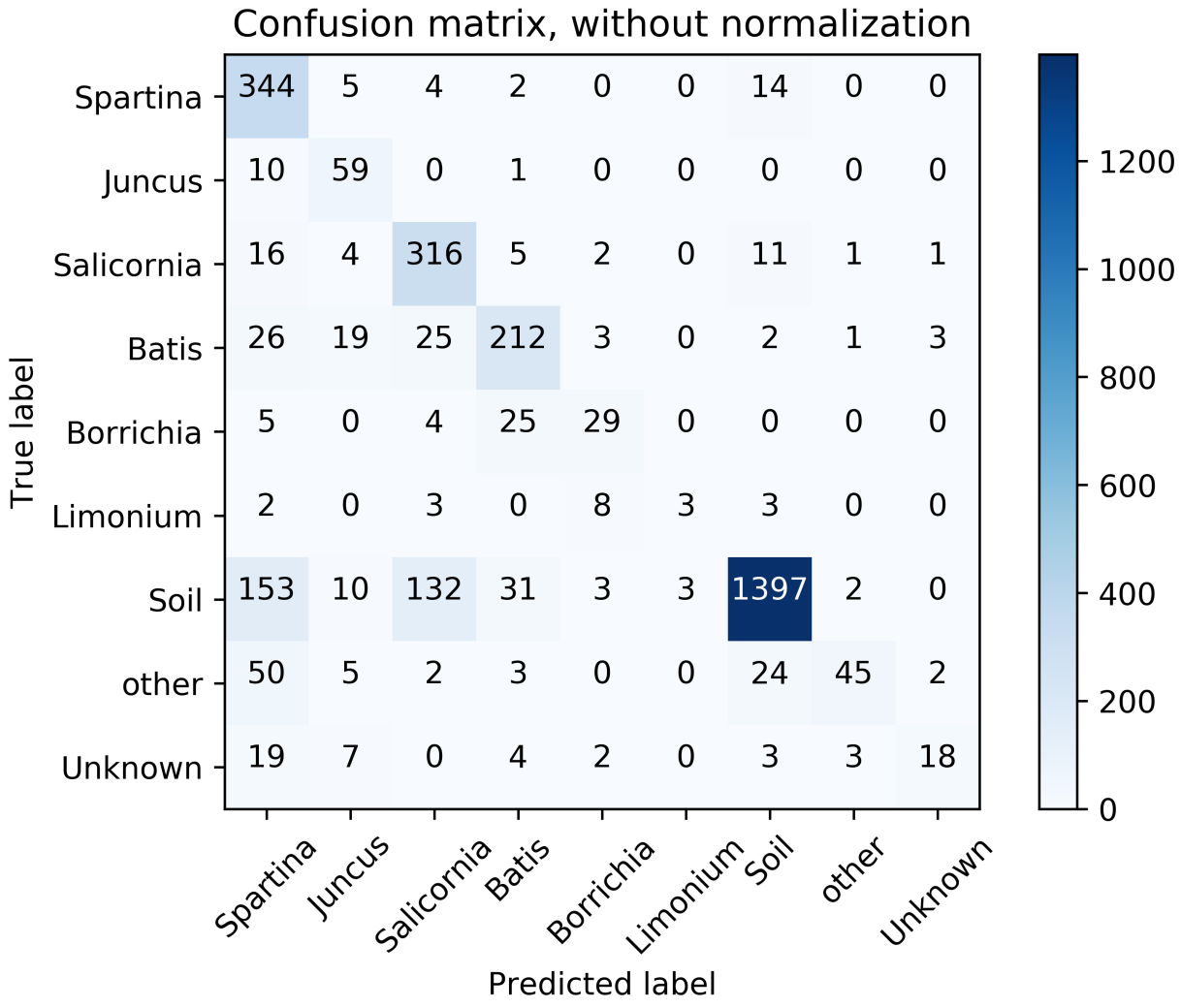

Figure 2.5: Confusion matrix for percent cover computation for *ResNext*

manually labeled ≈ 200 salt marsh image sections for semantic image segmentation using a super-pixel segmentation and labeling tool developed in [26]. The manually labeled data set was split into training (60%), validation (20%), and testing (20%) sets. The *DeepLab-V3* CNN [11] was trained on the training dataset for 100 epochs, while retaining the model that performed best on the validation set.

The training of the *DeepLab-V3* CNN employed a stochastic gradient descent optimizer with a learning schedule for weight decay. The best *mean Intersection-over-Union* (*mIoU*) measure was achieved with the *ResNet* backbone of *Deeplab-V3* which was more than twice of that achieved using the *Inception* backbone. The *mIoU* measure achieved using the *ResNet* backbone on the test data set was 0.54 with an overall pixel accuracy of 85%, and macro-averaged precision and macro-averaged recall scores of 0.570 and 0.607 respectively. For abundant classes such as *Spartina, Sarcocornia* and *Batis*, the semantic segmentation was highly accurate as illustrated in Fig. 2.6 and reflected in the high overall pixel-level accuracy (Table 2.4). However, the less abundant classes were either poorly predicted (*Borrichia*) or neglected entirely (*Limonium* and *Juncus*), which negatively impacted the macro precision and macro recall metrics. The current performance of the semantic segmentation approach is sufficiently accurate that it could be applied in ecological research to estimate the abundance and spatial distribution of abundant classes, but it clearly needs to be further improved before it can be used to study the rarer taxa. Additional training data on these less abundant classes is likely the best way to improve the performance of semantic image segmentation.

2.5.5 Comparison of Approaches

Three distinct tasks, i.e., presence/absence computation, percent cover computation, and semantic image segmentation, were evaluated as alternative approaches to assess the abundance and distribution of plants in salt marsh images. There are multiple potential scientific applications for these distinct approaches and our intent was not necessarily to determine the *best* approach but rather to explore the performance of these approaches and identify the underlying trade-offs. When the best performing CNN architecture was employed, all of the approaches performed reasonably well at classifying the abundant classes, i.e., *Spartina* and *Sarcocornia*, with $> 85\%$ precision and recall. The performance was observed to decline

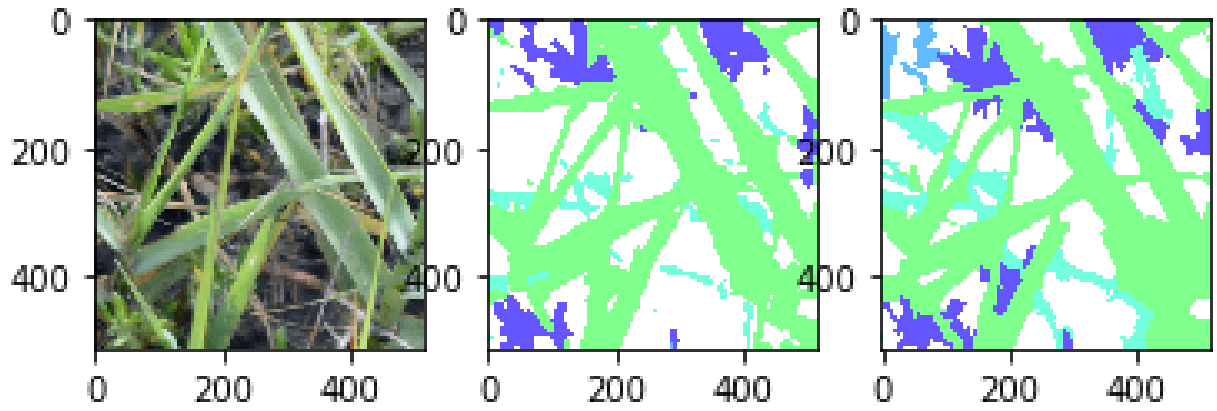

Figure 2.6: Semantic segmentation results for *DeepLab-V3*, showing from right to left : original image, output mask and target mask

for less abundant classes. The decline was more dramatic in the case of percent cover computation and semantic image segmentation. For example, semantic image segmentation was observed to completely ignore the *Limonium* and *Juncus* classes whereas presence/absence computation attained relatively high values of precision and recall for both classes compared to the other approaches. On many ecological tasks inter-agreement between human annotators is 70-90% indicating that in many cases our automated classifiers are likely to be as accurate as humans [6].

In essence, there was a clear trade-off observed between the performance (in terms of precision and recall) and spatial resolution of estimation with higher-resolution approaches exhibiting lower performance. Table 2.4 illustrates this trade-off showing that *accuracy* (i.e., micro-averaged *f-1* score) and the *f-1 score* (macro-averaged *f-1* score) decrease from the presence/absence computation task to the semantic image segmentation task whereas the *estimation resolution* (i.e., estimations per pixel) increases. One caveat of these comparisons is that different data sets were used for each approach. We put roughly the same effort in terms of human annotation hours into producing each manually annotated data set. Consequently, the comparison in Table 2.4 represents trade-offs for a similar amount of training data. It is expected that the performance of any of the approaches could be increased, to some extent, with additional training data. However, the difficulty of accurately detecting salt marsh plants at high resolution might require machine

learning approaches capable of representing concepts such as ambient lighting, object boundaries and object shapes.

Table 2.4: Comparison of results from different approaches

Approach	***Model***	***Accuracy***	***f-1 score***	***Resolution***
Presence/Absence	*ResNext*	0.929	0.853	e-7
Percent Cover	*ResNext*	0.857	0.770	e-3
Segmentation	*DeepLab*	0.849	0.587	1

2.5.6 Application example: Plant distribution across an elevation gradient

As a demonstration of the potential use of these methods, we employed the *ResNet101* classifier designed for presence/absence computation for all the images from the Sapelo Island marsh data set. The images were split into 15 sections (3 rows $\times$ 5 columns), as is done in the training procedure. The presence/absence classifier was used to determine which plant species were present in each image section. The total number of image sections (ranging from 0 to 15) in which a plant was present in each image was used as a semi-quantitative metric of plant abundance. This semi-quantitative index was averaged over all images for a given row to produce an estimate of plant abundance as a function of the row number, and plotted for all 80 rows as shown in Fig. 2.7. The results show a diverse plant community in the high-elevation marsh regions (row numbers < 25) transitioning to *Spartina* dominance in the low-elevation marsh regions (row numbers > 35) which is consistent with the expected distribution. Some spurious *Sarcocornia* predictions were observed in the low-elevation marsh regions (row numbers > 35) and efforts are currently underway to provide additional training data in this section to improve the quality of estimations. Nonetheless, this example illustrates the presence/absence computation method's potential to rapidly assess plant community structure across environmental (i.e., elevation) gradients in a salt marsh. Future work aims to apply these tools to assess changes in plant community structure over time at this Sapelo Island site and to extend the spatial scale of the sampling to additional sites.

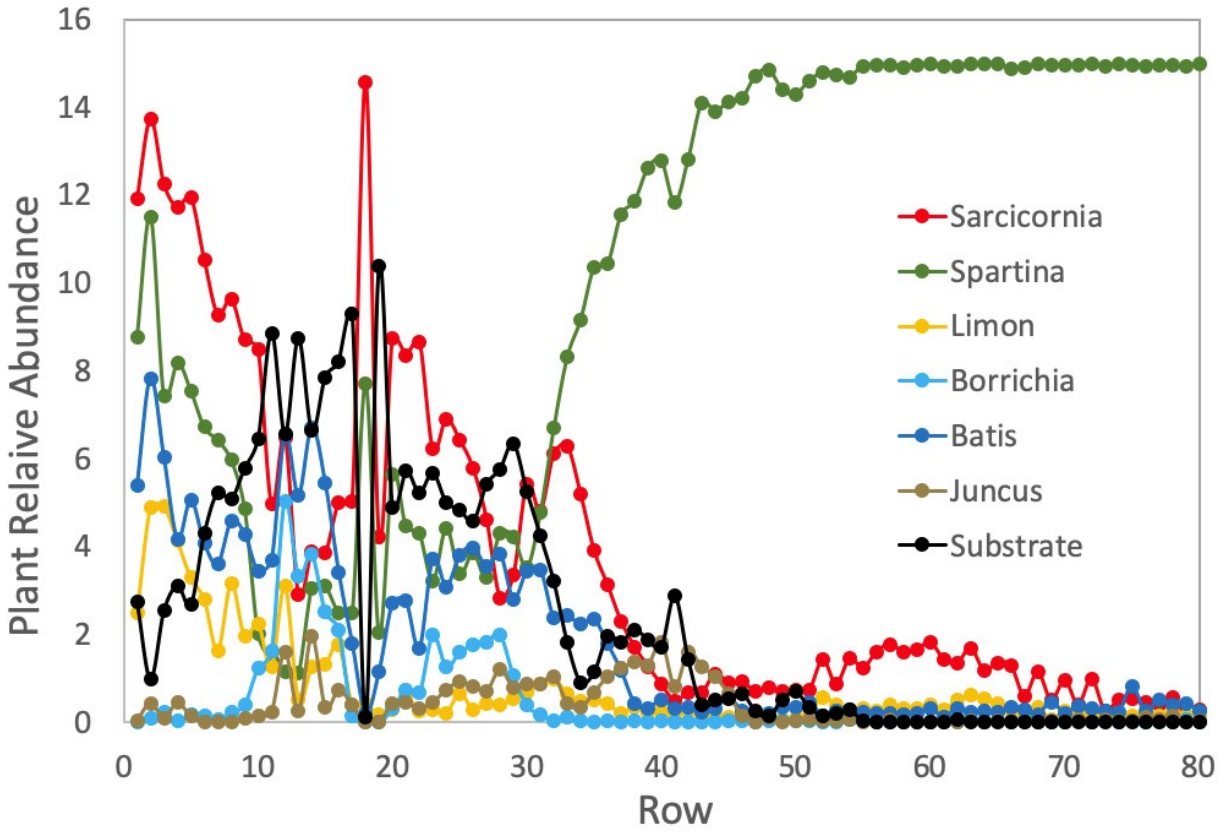

Figure 2.7: Row-wise distribution of plants across an elevation gradient

2.6 Future Work

In future work, we plan to introduce cross-talk between percent cover computation, presence/absence computation and semantic image segmentation. Rather than comparing these approaches based on results on their individual test data sets, we plan to devise a common testing data set to ascertain the relative performance of these approaches on producing a spatial plant class distribution. Since our plant classes are interweaved and require deep understanding of many high-order concepts, we argue that the best CNN models for our approach must possess generality. We also hypothesize that the best way to create neural networks capable of generality is through dynamic modularity [4]. Our proposed data set will provide a test for novel image analysis approaches that incorporate dynamic modularity. We propose to address these issues in our future work.

Chapter 3

Compositional Sparse Networks for Semantic Segmentation on Salt Marsh Images[1]

[1] Jayant Parashar*, Suchendra Bhandarkar,Brian Hopkinson, Steven Pennings. 2020. To be submitted to the International Conference on Pattern Recognition.

3.1 Abstract

In this chapter, we seek inspiration from the fields of neuroscience and continual learning to create networks that can compose meaningful features. We identify three critical principles lacking in modern neural networks that should be part of general-purpose networks: (a) interconnection richness, (b) sparsity, and (c) dynamic expansion. These principles form the basis of a general-purpose network architecture which we term as a *Compositional Sparse Network* (CSN). Since we dynamically expand the network and use learned weights from lower levels to train higher levels in the CSN, the CSN design process can be viewed as a pseudo Neural Architecture Search (NAS) procedure. The CSN is first tested on the CIFAR-10 image data set, and subsequently on the salt marsh image data set where the CSN is used as a backbone for the DeepLab-V3 architecture. We compare the performance of the CSN with a NAS-based approach called Auto-DeepLab which serves as a backbone for the DeepLab-V3 architecture. The proposed CSN approach is observed to perform worse than the NAS-based approach because the higher-level CSNs are seen to not fit the data distribution.

Keywords: convolutional neural networks, network topology, deep learning, semantic segmentation, compositional sparse networks, neural architecture search.

3.2 Introduction

In deep learning, we observe that the search strategies for the optimal network architecture are typically differentiated based on the underlying problem formulation. We believe that such differentiation helps us create general-purpose models only if we can find a way to segregate features based on their underlying neuronal structures. Since neural networks are a black box, the problem-specific architecture search process is merely tantamount to solving a set of specific problems. A manual search of neural network architectures for different problems does not contribute to efficient out-of-distribution (O.O.D) generalization and general intelligence. We should aim to find deep learning architectures that, except for a few input and output layers, share the same structure. Neural architecture search (NAS) [33] is an efficient architecture

search procedure used to determine an optimal neural architecture. Although NAS-based approaches are efficient in terms of the search procedure, they are usually limited to determining which stacking order of convolution filters is optimal. NAS-based approaches usually focus their search on the discovery of minor CNN enhancements that would potentially lead to an optimal architecture.

The architecture of a typical CNN follows a rigid three-level hierarchy comprising of: (a) CNN channels, (b) blocks, and (c) network architecture. Acting within the bounds of this hierarchy, the NAS procedure typically results in only incremental improvement in classification accuracy. We propose an alternative approach that tackles the above limitations. First, we recommend that deep learning research focus on the inter-connectivity of individual neurons, and how features are learned and composed [10]. Only based on this knowledge, can we then confidently construct general-purpose architectures that are capable of generalization on a wide array of problems. Based on a hypothesis of how neural networks generalize using interconnection richness, we propose a new paradigm for training neural networks. This paradigm is one of pseudo neural architecture search (NAS) that essentially constructs more complex neural architectures from simpler ones. The guiding vision behind this paradigm is to leverage rich interconnections within shallow sub-networks. Many intuitive ideas underlying CNN architecture design have followed this simple paradigm of feature composition such as gradual down-sampling of the image, dilated convolution [11], and residual network design [21]. Recent work in demystifying the reasons for the success of residual network has pointed towards their capacity to behave like ensembles of relatively shallow networks [49].

We believe that the proposed pseudo NAS process, termed as *Compositional Sparse Network* (CSN) design should be performed on carefully chosen *base networks* such as *graph neural networks* (GNNs) [59] or *fully connected dense neural networks* (FCNNs). However, in our work, we choose a 3×3 kernel, the most basic CNN unit, as our base network. The primary motivation behind choosing such a minimalist base networks is to construct a modular architecture capable of composing features on its own, without the external help of image down-sampling or dilated convolution. Moreover, the current trend towards general-purpose network design is not motivated by the need to tackle real-world data sets, but by novel

problem formulations such as meta-learning, few-shot learning, and continual learning[13]. These novel problem formulations usually exploit toy data sets and/or toy networks of small size. They have not been tested on larger real-world data sets and have been seen to not generalize well in many cases. This begs the question; why do we not implement our theories of general-purpose architecture design on common real-world problems such as semantic image segmentation? Since intuitively, we can expect a general-purpose network to significantly outperform a CNN that is custom designed for a specific problem, why do we not combine our hypotheses for out-of-distribution (O.O.D) generalization [28] arising from problem solving strategies such as meta-learning, few-shot learning, and continual learning and apply them to specific real-world problems such as semantic image segmentation to test their efficacy [4], [31]? With this in mind, we have formulated the concept of the CSN.

Deriving inspiration from the fields of Neuroscience [40] and Continual Learning [31], a CSN-based approach performs a pseudo NAS procedure that integrates our understanding of three basic principles required for incorporating general intelligence in neural networks: (a) interconnection richness hypothesis, (b) dynamic expansion and (c) sparsity. The interconnection richness hypothesis is based on the idea that the interconnections between local sub-networks and the corresponding interconnections between neurons in local sub-networks are critical to the performance of a neural architecture. The implementation of interconnection richness within a CSN framework is inspired by the structure of neocortex in the human brain [40]. In the CSN framework, to create a hierarchy of sub-networks, we facilitate dynamic expansion of repeating units. This idea is based on the intuition that a network must be able to increase its size based on the complexity of the problem. Traditional optimization-based dynamic expansion is short-sighted and leads to a cutoff depth that is much lower than the optimal depth required for a specific problem. Therefore, we propose and formulate a manual external dynamic expansion procedure coupled with network pruning to find general solutions [3]. While expanding a network dynamically, we should also be able to decrease its complexity to determine the most essential features to be composed, which is achieved via network pruning. Network pruning has been found to enhance the performance of a network despite deleting 60%-90% of its constituent neurons [16].

3.3 Background

Our main goal in this work is to create a general purpose training algorithm and network that is capable of O.O.D generalization [28] when applied to the salt marsh image data set [37]. The main idea behind O.O.D generalization is to learn stable features instead of spurious ones where stable features are deemed to be meaningful features with less variance in the training distribution compared to the spurious ones. The proposed CSN framework is intended as an alternative to the more common NAS framework in the context of semantic image segmentation. First, we expand on the three basic principles that CSNs are based on, and then describe recent developments in NAS techniques in the context of semantic segmentation, especially Auto-DeepLab [33].

3.3.1 Interconnection Richness Hypothesis

The overall structure of the neocortex in the human brain has been well known for several years [40]. In the neocortex, 80-110 neurons are arranged to form a cortical minicolumn with 50-100 cortical minicolumns forming a hypercolumn. Similarly, these hypercolumns collectively form uniform structures of cortical layers that culminate in specific cortices, resulting in a hierarchical organization of the neocortex. The interconnections between the cortical layers also exhibit a similar hierarchical structure. It is well known that the neocortex exhibits rich inter-connectivity across all levels of its structural hierarchy. We derive inspiration from the topology of the neocortex and argue that we can enhance the performance of artificial neural networks by ensuring interconnection richness. Recent research in combining ideas from Machine Intelligence and Neuroscience has shown that multiple models of the same concept or object are formed at a particular position in the cortical hierarchy [20]. This shows that biological neural networks probably have functional hierarchy, where at a particular point in hierarchy, the neocortex behaves as an ensemble of its smaller constituent units. We hypothesize that this structure could be highly conducive to feature composition and generality. Seeking inspiration from Neuroscience is not a new trend in Machine Learning and it is critical to the effort to design general-purpose neural networks [19].

But why is interconnection richness important? Input degradation and intermediate feature degradation, limit the capacity for learning general compositional features within a neural architecture. The degradation problem is a primary reason for limiting the number of layers within a deep network. Residual networks have partly solved the degradation problem and are considered a paradigm shift in CNN design [21]. But how exactly did they solve the degradation problem? Recent research indicates that residual networks unfold deep networks as an ensemble of relatively shallow networks [49]. The skip connections could also potentially help in creating multiple mini-modules within a ResNet by attempting to learn the same high level features as is done in the neocortical grid cells [20]. It is also probable that the features composed by a network ensemble that try to perform the same task is what produces a stable feature instead of a spurious one.

However, how does one progress beyond residual learning? How can we preserve the input and the intermediate- and high-level features so that we can facilitate their composition into general features through rich interconnections? One modern architecture that has circumvented the degradation problem and maintained rich interconnections is the transformer architecture [48]. Intuitively speaking, transformer architectures have managed to avoid the degradation problem by defining relatively shallow networks that are highly parallel in nature. By ensuring rich interconnections between input and the intermediate- and high-level features via self-attention, they enable multiple theories of an object or concept using multiple heads of attention. The multiple heads of attention function as an ensemble within the transformer network. However we do not envisage transformer networks as a long-term solution. A long-term solution would entail the hard path to find a way to merge memory systems with reasoning systems [55]. Replaying of memories can solve the degradation problem by preserving the input and the intermediate- and high-level features for an infinitely deep network. Therefore, the interconnection richness hypothesis will likely be able to achieve composition of general features if the inputs and the intermediate- and high-level features are preserved in memory. In the present work, we do not address the memory component required for composition of features, rather we merely exploit the residuals in our base network and the repeating network structure.

In our previous work on comparison of deep learning techniques for analysis of salt marsh image data, we found that multi-path networks such as the DPN [12] and ResNext [57] performed better than the ResNet, albeit slightly. Thus, the richness of interconnections across varying levels of the network hierarchy may yield high-level features that are stable and thus generalize well even beyond the given training data distribution. Next, we discuss how we may be able to create such features incrementally through dynamic expansion.

3.3.2 Dynamic Expansion

A great deal of work towards building general purpose networks has been done within the continual learning and meta-learning problem formulations. The main goal of continual learning is to prevent catastrophic forgetting. This is achieved by producing general features across separate domains that are re-usable. A seminal work in continual learning via dynamic expansion is the formulation of the *Dynamic Expandable Network* (DEN) [31]. The intuition underlying the DEN is to prevent semantic drift by duplicating and splitting units using relevant measures. Lee et al. [31] show the efficacy of small DENs that can expand units and duplicate network structures for small-scale continual learning problems. They show DENs to be a part of an incremental learning formulation where the classes are added incrementally. The expansion procedure is external to the network, and is not part of the optimization of the loss function, as is the case with our CSN formulation. Lee et al. [31] also use group-sparsity regularization, which is an optimization-based method, to introduce sparsity. Although the DEN formulation of Lee et al. [31] bears some resemblance to our CSN formulation, it should be noted that unlike Lee et al.[31] we choose to apply the proposed CSN framework to a larger-scale real-world problem.

There are other ways to expand a neural network, such as by using optimization-based methods. Continuously Constructive Deep Neural Networks (CCDNNs) [24] employ optimization-based expansion which either expands the network to introduce a new unit or a new layer based on a control parameter which is part of the network structure. While training, the size of the network increases rapidly and then

decreases gradually during inference. Moreover, Irsoy and Alpaydin [24] have not yet generalized their CCDNN formulation to include CNNs.

The problem with current optimization-based expansion methods is that they do not naturally tend to result in deeper networks. This is partly because larger networks are harder to train. Moreover, there is always a local minimum that a smaller or shallower network can find which obviates the need for a larger or deeper network. Non-parametric neural networks [39] is also an emerging approach to dynamic network expansion using optimization-based methods. However, instead of adding layers gradually, they can generate a network of any arbitrary size in a single step. The number of network layers is determined by the network parameters which are regularized to eliminate redundancies. We consider dynamic expansion to be an essential property for incorporating general intelligence in artificial neural network design and regard cortical reorganization in Neuroscience [15] to be the analogy of dynamic expansion in artificial neural networks.

3.3.3 Sparsity

There are many ways to incorporate sparsity in deep learning models. Sparsity can be introduced using group-sparsity regularization [42], neuromodular meta-networks [4], attention and pruning. It is a well known fact that human brain works on sparse representations and some recent research has shown that sparse representations are more robust to noise and interference and hence should be implemented in artificial neural networks despite the recent success of overparameterized neural networks [1]. In our proposed CSN framework, we focus on pruning to induce sparsity. In their seminal work in network pruning, termed as the Lottery Ticket (LT) hypothesis, Frankle and Carbin [16] showed that reinitializing the network weights to their initial (i.e., prior to pruning) values after the pruning procedure can produce better performance than the original (i.e., unpruned) dense network. Their LT algorithm [16] essentially finds the winning ticket, i.e., the best sub-network within a dense network. The results of the LT algorithm came as a surprise to the research community and caused a great deal of speculation as to the cause(s) for its

success. The success of the LT algorithm also spawned research in the area of better initialization methods based on determining winning tickets across different data sets [36].

Significant recent work by Zhou et al. [58] at UberAI has attempted to answer the important question of why pruned networks outperform their non-pruned counterparts when reinitialized with their initial weights before training. They formulate *supermasks* that give more than chance accuracy when applied to models and show that signs of the initial weights initialization play a significant role in the performance of the LT network. Their explanation of why effective pruning is so tightly coupled with the choice of the initial weights is that pruning eliminates weights that were going towards zero anyway in the initial random initialization. Their results show that overparameterized networks can heavily benefit from sparse representations [58].

Our primary interest in the pruning procedure is to extract general features from the network. The work of Bartoldson et al. [3] presents an interesting generalization versus stability tradeoff during the network pruning process. The more pruning causes instability in testing accuracy in the immediate term, the greater the generalization achieved upon convergence. The effect of the generalization determined by the degree of destabilization is similar to one achieved by neuronal dropout. Thus, pruning can either hasten or destabilize the training procedure to improve generalization accuracy in faster networks [3].

3.3.4 Neural Architecture Search (NAS) on Semantic Segmentation

NAS architectures typically have a two-level hierarchy comprising of the cellular level and the network level; however, most NAS architectures perform search solely on the cellular level. The Auto-DeepLab [33] architecture improves the NAS procedure by searching through both, the cellular and network levels. Auto-DeepLab is also the first attempt to extend the NAS approach to the semantic image segmentation problem. Auto-DeepLab traverses the entire search space of neural architectures using a continuous relaxation technique and is optimized via an appropriate choice of meta-parameter values. The best searched neural architecture is then deciphered after measuring the dense cellular-level meta-parameters

and network-level meta-parameters. The best searched architecture is then used as a backbone for the DeepLab-V3 architecture [11].

3.4 Data Sets

3.4.1 Semantic Image Segmentation on the Salt Marsh Data Set

We have ≈ 500 labeled images in the salt marsh image data set. We use 350 images for training, 50 images for validation and 100 images for testing. The goal of semantic image segmentation is to classify each pixel in an image into one of predetermined categories or classes. To generate training data for the semantic image segmentation task, we used a superpixel labeling tool described in [26]. Each image pixel is classified into one of nine classes, which are slightly different from those used for percent cover computation. The nine classes include the previously mentioned six plant classes, *dead Spartina* (due to its common occurrence and substantially different appearance compared to *live Spartina*), background (which accounts for the classes *Soil* and *Unknown* in percent cover computation), and *Other* (which accounts for *invertebrates*, and portions of imaging platform captured by the camera).

3.4.2 The CIFAR-10 Data Set

CIFAR-10 is a general image classification data set comprising of ≈ 60,000 images with ≈ 40,000 images used for training and ≈ 10,000 images each for validation and testing. The ten different classes are *airplane, car, bird, cat, deer, dog, frog, horse, ship*, and *truck* where each class has ≈ 6,000 images. The size of each image is 32×32 pixels which makes testing new architectural models easier. We used the CIFAR-10 data set to search the best hyper-parameters for the proposed CSN such as efficacy of pruning, network depth, network width, number of exponential expansion levels etc.

3.5 CSN Design Methodology

The design of a Compositional Sparse Network (CSN) is based on the unification of three principles: *interconnection richness, dynamic expansion* and *sparsity*. The CSN design starts with a base network, copies of which are combined using a cerebral cortex-like design in a *cerebral module* and inter-connected using an *interconnection module*. The cerebral module is a generic design that can work on a base network and itself. It represents a recursive structure which interconnects predefined modules with a predefined interconnection class. It is inspired by the structure of the neocortex in the human brain [40].

The proposed CSN design methodology can compose increasingly complex networks by stacking its lower-level network structure across a given depth and width. These modules are then pruned, resulting in a minimal set of performing units which are then repeated to generate higher levels in the CSN architecture. The CSN design methodology thus comprises of multiple cycles of training, pruning and expansion procedures, resulting in a pseudo NAS process that finds then best neural network structure through dynamic expansion, pruning and interconnection richness. We have implemented two expansion procedures, i.e., exponential expansion and linear expansion which differ in the manner in which the parameters of the base network scale with increasing CSN levels. Note that the concept of linear expansion derives inspiration from the cortical reorganization phenomenon in the human brain [15].

3.5.1 Base Network for CSN Design

For the base network in the CSN, we use three convolution layers of size 3×3 with 64 channels. We have one residual connection in the base network and one residual connection in each cerebral module. We experimented with multiple designs inspired by the early-age CNN architectures.

3.5.2 Interconnection Richness Hypothesis

The *interconnection richness* hypothesis is based on a premise that the general stable features extracted in a network are dependent on the interconnection distribution across the network architecture in the context

of the provided input signals. Therefore, we hypothesize that the best way to build neural networks capable of O.O.D generalization [28] is to leverage the hierarchical interconnections between the various levels of sub-networks within the neural architecture. This may lead to compositional features by creating locally deeply connected sub-networks at multiple hierarchical levels of the neural architecture. Moreover, these features learned by small networks of similar shape and size would function as several brains in tandem thereby producing generality [20]. Note that even the success of residual learning can be explained by the unfolding of their depth into an ensemble of smaller-size networks [49].

We take an idealistic approach to designing architectures for neural networks by letting the architecture compose features on its own, guided by its design. We do not down-sample the input or use dilated convolution, rather we seek to compose features using only 1×1 convolutions and the cerebral cortex-like design. It is important to note that width of the proposed CSN architectures is designed to be orders of magnitude higher than that of typical CNNs since the cerebral module is designed to symmetrically expand across depth and width of the neural architecture.

The biggest obstacle to feature composition within a deep neural network, in our opinion, is memory [55]. Free flow of information in a neural network leads to a rapid fitting phase and potentially responsible for high-level feature composition [41]. The manner in which low-level features are combined to produce high-level features depends largely on how easy it is for information to flow within the network. The biggest impediment to such feature composition is the degradation of the input and the features over the depth of the network. The multiple levels of hierarchy of sub-networks are characterized by varying levels of interconnections that could potentially contribute towards the composition of features. Since the incorporation of pooling, strides, convolutional filters and dilated convolution increase the neural receptive fields and thus promote richer interconnections, the success of architectures that leverage them is no surprise. We maintain the same input image size across a majority of the CSN design to promote a neural network that can combine features on its own without having to resort to stride- and pooling-based feature combinations. The proposed CSN design is based on an intuition about the flow of information in neural networks. It aims to train many small network sub-modules to get them to find independent

features in a manner similar to manual feature construction based solely on network topology. In this regard, the proposed CSN approach bears some similarity to the NEAT system proposed by Stanley and Miikkulainen [44].

3.5.3 Cerebral Module and Interconnection Module

The cerebral module stacks multiple children networks, for example the base network, across the given width and depth values of the neural architecture. The children networks are interconnected using an interconnection module that also has a depth value of 2 or 3. The output of each cerebral module also contains residuals from its input causing every module to output a residual value from its input to its output. The channel size of the interconnection module is based on number of modules it is connecting. The goal of the interconnection module is to sew together multiple sub-networks and ensure interconnection richness.

3.5.4 Network Expansion

The cerebral module recomposes itself when expanded. There are two ways in which the cerebral module can expand; the first is exponential expansion where both the network depth and width increase exponentially which increases the number of levels of hierarchy in the network (Fig. 3.1). The second is linear expansion, which adds multiple low-level CSNs across the network depth or network width or both (Fig. 3.2). Linear expansion does not increase the size of network exponentially but enables a slower linear network growth. Most of our experiments are done using linear expansion as exponential expansion turned out to be difficult to implement on a GPU. The manner in which the weights from lower-level CSNs are used for subsequent CSNs is illustrated in Fig. 3.1 and Fig. 3.2. In the case of exponential expansion, all the CSN weights are reused multiple times whereas in the case of linear expansion, the weights from the lower-level CSNs are inserted in the middle of the network.

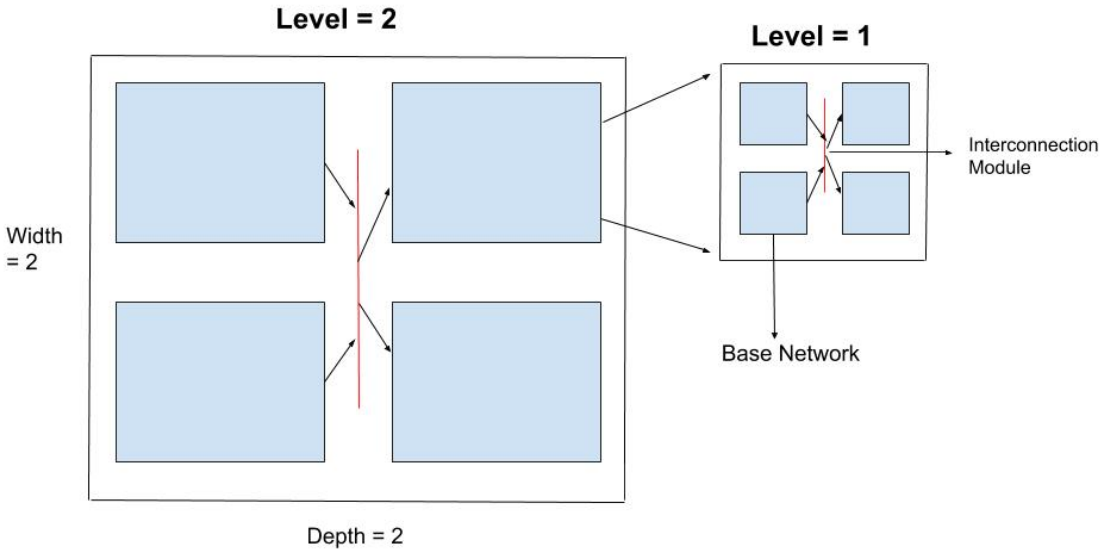

Figure 3.1: An example of exponential expansion is shown. Weights from lower levels are used at higher levels.

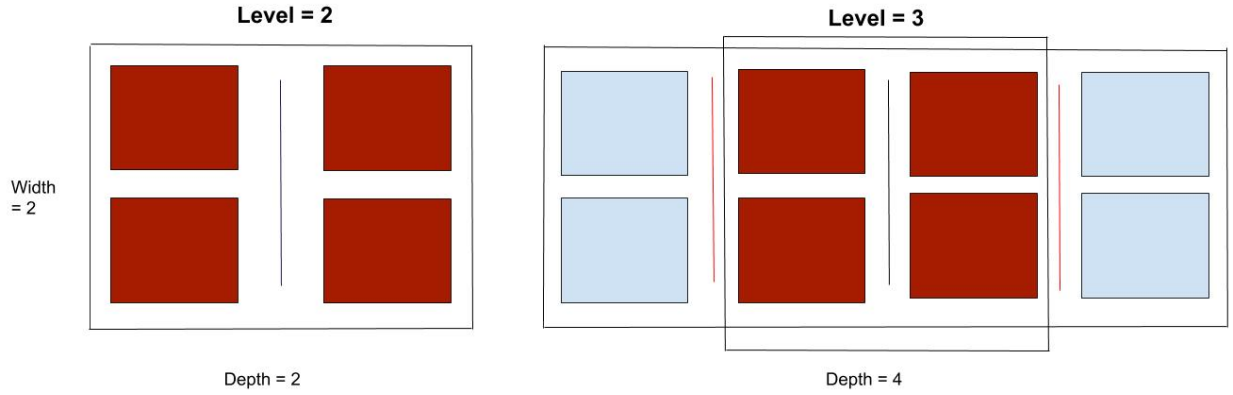

Figure 3.2: An example of linear expansion is shown. Weights from lower levels are reused at higher levels as indicated by blocks and lines of the same color.

3.5.5 Network Pruning

We prune our Cerebral Modules using either structured or global pruning. Structured pruning is when percentage of pruning is evenly distributed on each layer. Whereas global pruning prunes a network by comparing weights to all other weights in the network. Structural integrity becomes a problem in global pruning, but not in structured pruning. We also randomize the pruned weights by taking a weighted mean of the pruned weights with random initialization of the same. The motivation behind randomizing weights is to induce instability that leads to better generalization[3].

3.5.6 CSN Algorithm

The CSN algorithm expands the base network using either exponential expansion or linear expansion as illustrated in Fig. 3.1 and Fig. 3.2 respectively. The base network model is the first trained model and is chosen before training the entire network. The base model is used to construct the CSN model at level i denoted by CSN_i by using CSN training algorithm with the expansion procedure but without the training and pruning procedures. The CSN algorithm is illustrated in Algorithm 1.

3.6 Experimental Results

3.6.1 Experimental Setup

We experimented with base networks with a varying number of CNN filters. The first level of the proposed CSN architecture, termed as CSN_1, is shown in Fig. 3.3. We settled on three $3 \times 3 \times 64$ convolution filters and used a Tesla P100 GPU. For exponential expansions, we considered network width and depth values of 2 since higher depth and width values resulted in fewer expansions due to limited computational resources. In the case of linear expansion, we expanded the network depth by 4 units at each step whereas the network width was kept constant because of limited computational resources. The first trained model

Algorithm 1: CSN Training Algorithm

$CSN_0 = BaseNetwork$
$BaseModel\_Level = i$
$Exponential\_Expansion\_Level = j$
$Linear\_Expansion\_Level = k$
$CSN_i = Expand(CSN_0, i)$
for num in range$(0, j)$ **do**
  $CSN_{i+num} = Expand(CSN_0, i + num, "Exponential")$
  Load Weights of $CSN_{i+num-1}$ into CSN_{i+num}
  Train(CSN_{i+num})
  Prune(CSN_{i+num})
  Save Weights(CSN_{i+num})
end for
for num in range$(0, k)$ **do**
  $CSN_{i+j+num} = Expand(CSN_0, i + num, "Linear")$
  Load Weights of $CSN_{i+j+num-1}$ into $CSN_{i+j+num}$
  Train($CSN_{i+j+num}$)
  Prune($CSN_{i+j+num}$)
  Save Weights($CSN_{i+j+num}$)
end for

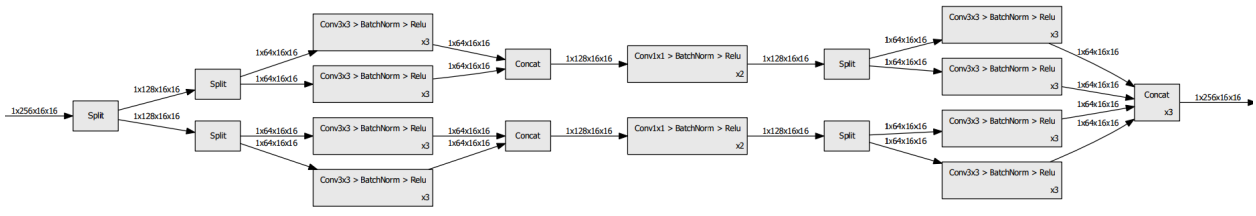

Figure 3.3: CSN Level 1 architecture

whose weights are used for subsequent CSNs is termed the base model. We typically start with a 1-2 level exponential expansion followed by 2-4 level linear expansion.

3.6.2 Proof of Concept Implementation on the CIFAR-10 Data Set

The CSN training algorithm pipeline was fully implemented on the CIFAR-10 data set as a proof of concept. We used the Adam optimizer and found a learning rate of 5e-4 to perform the best. In case of exponential expansion, the width and depth values are kept perfectly symmetrical. We ran exponential expansion on CSN_1 and CSN_2 and, as a result, CSN_3 was our largest exponentially expanded model with 12 million parameters. We found that most of our models converged to around 70%-80% test accuracy with global and structured pruning with structured pruning performing slightly better than global pruning. As expected, higher-level CSNs took longer to train and were found to perform slightly better than lower-level CSNs.

We conducted experiments without pruning to see the effect of dynamic expansion. Without pruning, the larger-size networks were observed to always overfit. The generalization error increased rapidly as we increased the number of levels from 1 to 4 as shown in Table 3.1. The results in Table 3.1 suggest that the higher-level CSNs without pruning are highly susceptible to fitting to the noise. It is possible that the reusing of weights led to convergence to the same local minima. When we pruned the weights and randomly varied them, we observed lower generalization error in the case of higher-level CSNs as shown in Table 3.2. However, as seen in Table 3.2, CSN_3 and CSN_4 are not able to completely learn the features produced by CSN_1. We surmise that this is because higher-level CSNs are merely trying to relearn what CSN_1 has already learned. The entire training process just reversing the instability we induced [3]. We had expected that, like ensembles, the different modules in the CSN would start to specialize in the formulation of distinct features. However, the experimental results ran quite contrary to that expectation. Also, despite the presence of residuals at multiple levels of the CSN hierarchy, we witnessed vanishing gradients above CSN_5.

Table 3.1: Performance of CSN on CIFAR-10 without pruning

Model	***Train Accuracy***	***Test Accuracy***	***Optimizer***
CSN_1(Base)	84.03	74.46	Adam
CSN_2	96.53	75.32	Adam
CSN_3	92.26	69.33	Adam

Table 3.2: Performance of CSN on CIFAR-10 with pruning

Model	***Train Accuracy***	***Test Accuracy***	***Optimizer***
CSN_1(Base)	83.31	77.87	Adam
CSN_2	85.61	78.01	Adam
CSN_3	88.64	78.07	Adam
CSN_4	71.205	73.27	Adam

3.6.3 Semantic Segmentation on Salt Marsh Image Data Set

We tested the CSN training algorithm as a backbone for DeepLab-V3 [11] and compared it with a ResNet101 backbone for DeepLab-V3 and with Auto-DeepLab [33]. The performance of DeepLab-V3 was observed to be the best. The Auto-DeepLab search was performed on a smaller subset of the entire data set where the input images were resized to a low resolution for faster search. Auto-DeepLab's search produced a model that performed slightly worse than DeepLab-V3 as shown in Table 3.3. The CSN was also tried and tested as a backbone for DeepLab-V3. The CSN stem for the backbone was created such that input was down-sampled to 1/4 before input to the core CSN and subsequently down-sampled to 1/4 after the core CSN resulting in an overall down-sampling rate of 1/16 for the backbone. This was done to test the efficacy of the structure of the CSN.

In our experiments, the CSN performed worse than both, Auto-DeepLab and DeepLab-V3. This is not a complete surprise given that testing on the CIFAR-10 data set also showed that the CSN design is not learning at par with modern CNNs. The degradation in mIoU values can be primarily attributed to the underrepresented classes. We observe that CSN_1 CSN_2 and CSN_3 exhibit similar performance

despite the exponential variation in the number of parameters. We also see that the CSN_3 base model does not perform any worse than CSN_3. This indicates that the reuse of weights from CSN_1 and CSN_2 via dynamic expansion is not the cause for the lacklustre performance of CSN_3. However, it points towards a problem in the construction of either the base network or the cerebral module.

Table 3.3: Performance comparison between CSN and CNNs on Salt Marsh image data set with network pruning

Model	***mIOU***	***Pixel Accuracy***	***Optimizer***
DeeplabV3	*0.451*	86.54	SGD
AutodeepLab	*0.3706*	85.049	SGD
CSN_1(Base)	*0.2483*	81.6	Adam
CSN_2	*0.2411*	81.6	Adam
CSN_3	*0.2361*	81.76	Adam
CSN_3(Base)	*0.2379*	81.04	Adam

3.7 Conclusion

The consequence of the increased computational cost of exponential expansion and linear expansion is that the interconnection hypothesis is not implemented properly. Since we are only able to reach hierarchies of up to 3 levels in the CSN, this is merely 1 level above that of modern CNN's. Another cause for concern is that the CSN base models at higher levels are prone to not learning anything new. The challenge of training extremely deep and wide networks is evident in our work. We believe that the solution to this problem will be pivotal in moving the field forward. A lot more work will be needed to find general-purpose neural network architectures that work and scale well on real-world problems. We believe that the reward for this effort will be in the form of solutions that are extremely unique and useful. The reason that our interconnection design has under-performed is likely due to the fact that the cerebral module or base network design needs to be revamped. We believe our interconnection design may not have met our criterion of interconnection richness due to the lack of a memory system [55]. Most modern CNNs are organized to compose features hierarchically before degradation of the input is observed. Unfortunately,

in our efforts to find a novel CSN-based approach, we negated many architectural advancements that have contributed to the success of modern CNNs.

Chapter 4

Conclusion and Future Work

By comparing various approaches for percent cover computation, presence/absence determination and semantic image segmentation on salt marsh images we found a trade-off between resolution and precision of predictions. The presence/absence determination approach was the most accurate but lacked in resolution. In contrast, the semantic image segmentation approach was the highest in terms of resolution but lacked in precision as it missed low frequency classes. Moreover, while comparing the performance of various convolutional neural networks (CNNs) on salt marsh image classification, we found that multi-path networks outperformed others. This observation inspired us to design cerebral cortex-like neural network architectures and thus define a principle called *interconnection richness*. We tested this hypothesis in conjunction with dynamic expansion and pruning. However, current results show that a lot more work needs to be done.

There is a much room for improvement in designing the cerebral module and base network. We see this as an opportunity and an unexplored area for future research. We believe that continual learning, meta-learning and few-shot learning problems should be specifically targeted by compostional sparse networks (CSNs). A clear and precise mathematical definition of the interconnection hypotheses in terms of information flow is needed [17]. We have to essentially decrease cost of information flow through the network interconnections. Thus, a more rigorous analysis of information flow in neural network architectures is called for.

Network pruning turned out to be a niche concept in our experimental setting, having little impact on final performance. Our initial expectation of network pruning was that it would, like dropout, lead to the formulation of more general features. However, pruning, at least in our case, remains merely a curious phenomenon of preservation or fine-tuning of current features, rather than learning of new ones. Network pruning might not be the best way forward for creating or exploiting sparsity. Optimization-based methods for creating sparsity such as group-sparsity regularization may be a better fit for this problem.

We think that dynamic expansion of CSNs also holds a lot of promise and we did observe early convergence resulting from dynamic expansion compared to static architectures. However, our aim was to find convergence in situations where low-level hierarchies would fail to do so. Our experiments did not point to any viable results in that direction. We did not anticipate that high-level CSNs might potentially reuse weights of lower-level CSNs to converge to the same local minima.

Future work should focus on formulating interconnection schemes for cerebral modules by taking dense connections into account. Base networks could also be formulated solely in terms of graph neural networks [59] as they have been shown to be uniquely successful in few-shot learning problems with their inherent interconnection richness design. Moreover, dynamic expansion should be attempted with much larger computational resources. There is also hope for unifying the concept of the CSN with meta-learning approaches under an optimization-based framework that also incorporates long-term memory in its design [55].

Bibliography

[1] S. Ahmad and L. Scheinkman, "How can we be so dense? the benefits of using highly sparse representations", *ArXiv*, vol. abs/1903.11257, 2019.

[2] E. Ayrey and D. Hayes, "The use of three-dimensional convolutional neural networks to interpret lidar for forest inventory", *Remote Sensing*, 2018.

[3] B. Bartoldson, A. S. Morcos, A. Barbu, and G. Erlebacher, "The generalization-stability tradeoff in neural network pruning", *ArXiv*, vol. abs/1906.03728, 2019.

[4] S. Beaulieu, L. Frati, T. Miconi, J. Lehman, K. Stanley, J. Clune, and N. Cheney, "Learning to continually learn", *arXiv*, 2020.

[5] O. Beijbom, P. Edmunds, D. Kline, B. Mitchell, and D. Kriegman, "Automated annotation of coral reef survey images", *Proc. IEEE CVPR*, 2012.

[6] O. Beijbom, P. Edmunds, C. Roelfsema, J. Smith, D. Kline, and B. N. et al., "Towards automated annotation of benthic survey images: Variability of human experts and operational modes of automation", *PLoS One*, 2015.

[7] M. Bertness and A. Ellison, "Determinants of pattern in a new england salt marsh plant community", *Ecological Monographs*, 1987.

[8] C. Bowley, A. Andes, S. Ellis-Felege, and T. Desell, "Detecting wildlife in uncontrolled outdoor video using convolutional neural networks", *Proc. IEEE Intl. Conf. E-Science*, 2016.

[9] P. Brodrick, A. Davies, and G. Asner, "Uncovering ecological patterns with convolutional neural networks", *Trends Ecol. Evol.*, 2019.

[10] N. Cammarata, S. Carter, G. Goh, C. Olah, M. Petrov, and L. Schubert, "Thread: Circuits", *Distill*, 2020, https://distill.pub/2020/circuits. DOI: 10.23915/distill.00024.

[11] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation", *arXiv*, 2017.

[12] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng, "Dual path networks", *Proc. NIPS*, 2017.

[13] J. Clune, "AI-GAs: AI-generating algorithms, an alternate paradigm for producing general artificial intelligence", *arXiv e-prints*, arXiv:1905.10985, arXiv:1905.10985, May 2019. arXiv: 1905.10985 [cs.AI].

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database", *Proc. IEEE CVPR*, 2009.

[15] T. Elbert, A. Sterr, B. Rockstroh, D. Charbonnier, H. Flor, C. Pantev, C. Wienbruch, S. Knecht, and E. Taub, "Cortical reorganization in arm amputees: Alterations of the somatosensory representation of the intact arm", in *Biomag 96*, C. J. Aine, G. Stroink, C. C. Wood, Y. Okada, and S. J. Swithenby, Eds., New York, NY: Springer New York, 2000, pp. 999–1002, ISBN: 978-1-4612-1260-7.

[16] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks", *arXiv: Learning*, 2019.

[17] Z. Goldfeld, E. van den Berg, K. H. Greenewald, I. Melnyk, N. Nguyen, B. Kingsbury, and Y. Polyanskiy, "Estimating information flow in neural networks", *ArXiv*, vol. abs/1810.05728, 2018.

[18] D. Han, J. Kim, and J. Kim, "Deep pyramidal residual networks", *Proc. IEEE CVPR*, 2017.

[19] D. Hassabis, D. Kumaran, C. Summerfield, and M. M. Botvinick, "Neuroscience-inspired artificial intelligence", *Neuron*, vol. 95, pp. 245–258, 2017.

[20] J. Hawkins, M. Lewis, M. Klukas, S. Purdy, and S. Ahmad, "A framework for intelligence and cortical function based on grid cells in the neocortex", *Frontiers in Neural Circuits*, vol. 12, p. 121, 2019, ISSN: 1662-5110. DOI: 10.3389/fncir.2018.00121. [Online]. Available: https://www.frontiersin.org/article/10.3389/fncir.2018.00121.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", *Multimedia Tools and Applications*, 2015.

[22] J. Heinzel and B. Koch, "Investigating multiple data sources for tree species classification in temperate forest and use for single tree delineation", *Intl. Jour. Applied Earth Observation and Geoinformation*, 2012.

[23] G. Huang, Z. Liu, L. V. D. Maaten, and K. Weinberger, "Densely connected convolutional networks", *Proc. IEEE CVPR*, 2017.

[24] O. İrsoy and E. Alpaydın, "Continuously constructive deep neural networks", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 4, pp. 1124–1133, 2020.

[25] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks", *arXiv*, 2019.

[26] A. King, S. M. Bhandarkar, and B. M. Hopkinson, "A comparison of deep learning methods for semantic segmentation of coral reef survey images", *Proc. IEEE CVPR*, 2018.

[27] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks", *Proc. NIPS*, 2012.

[28] D. A. Krueger, E. Caballero, J. Jacobsen, A. Zhang, J. Binas, R. L. Priol, and A. C. Courville, "Out-of-distribution generalization via risk extrapolation (rex)", *ArXiv*, vol. abs/2003.00688, 2020.

[29] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning", *Nature*, 2015.

[30] Y. Lecun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Backpropagation applied to handwritten zip code recognition", *Neural Computation*, 1989.

[31] J. Lee, J. Yoon, E. Yang, and S. J. Hwang, "Lifelong learning with dynamically expandable networks", *ArXiv*, vol. abs/1708.01547, 2018.

[32] M. Lin, Q. Chen, and S. Yan, "Network in network", *arXiv*, 2013.

[33] C. Liu, L.-C. Chen, F. Schroff, H. Adam, W. Hua, A. Yuille, and L. Fei-Fei, "Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation", *CVPR*, 2019.

[34] E. McLeod, G. Chmura, S. Bouillon, R. Salm, M. Björk, C. Duarte, C. Lovelock, W. Schlesinger, and B. Silliman, "A blueprint for blue carbon: Toward an improved understanding of the role of vegetated coastal habitats in sequestering co_2"", *Frontiers in Ecology and Environment*, 2011.

[35] W. Mitsch and J. Gosselink, "Wetlands", *Wiley*, 2015.

[36] A. S. Morcos, H. Yu, M. Paganini, and Y. Tian, "One ticket to win them all: Generalizing lottery ticket initializations across datasets and optimizers", *ArXiv*, vol. abs/1906.02773, 2019.

[37] J. Parashar, S. Bhandarkar, J. Simon, B. Hopkinson, and S. Pennings, "Estimation of abundance and distribution of salt marsh plants from images using deep learning", *ICPR*, vol. preprint, 2020.

[38] S. Pennings, M. Grant, and M. Bertness, "Plant zonation in low-latitude salt marshes: Disentangling the roles of flooding, salinity and competition", *Jour. Ecology*, 2004.

[39] G. Philipp and J. G. Carbonell, "Nonparametric neural networks", *ArXiv*, vol. abs/1712.05440, 2017.

[40] A. Rockel, R. Hiorns, and T. Powell, "The basic uniformity in structure of the neocortex", *Brain : a journal of neurology*, vol. 103, no. 2, pp. 221–244, Jun. 1980, ISSN: 0006-8950. DOI: 10.1093/brain/103.2.221. [Online]. Available: https://doi.org/10.1093/brain/103.2.221.

[41] A. M. Saxe, Y. Bansal, J. Dapello, M. Advani, A. Kolchinsky, B. D. Tracey, and D. D. Cox, "On the information bottleneck theory of deep learning", in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=ry_WPG-A-.

[42] S. Scardapane, D. Comminiello, A. Hussain, and A. Uncini, "Group sparse regularization for deep neural networks", *Neurocomputing*, vol. 241, pp. 81–89, 2017.

[43] R. Soltani and H. Jiang, "Higher-order recurrent neural networks", *arXiv*, 2016.

[44] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies", *Evol. Comput.*, vol. 10, no. 2, pp. 99–127, Jun. 2002, ISSN: 1063-6560. DOI: 10.1162/106365602320169811. [Online]. Available: https://doi.org/10.1162/106365602320169811.

[45] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions", *Proc. IEEE CVPR*, 2015.

[46] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision", *Proc. IEEE CVPR*, 2016.

[47] H. Touvron, A. Vedaldi, M. Douze, and H. Jegou, "Fixing the train-test resolution discrepancy", *Proc. NIPS*, 2019.

[48] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need", *ArXiv*, vol. abs/1706.03762, 2017.

[49] A. Veit, M. Wilber, and S. Belongie, *Residual networks behave like ensembles of relatively shallow networks*, 2016. arXiv: 1605.06431 [cs.CV].

[50] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification", *Proc. IEEE CVPR*, 2017.

[51] K. Wasson, K. Raposa, M. Almeida, K. Beheshti, and J. C. et al, "Pattern and scale: Evaluating generalities in crab distributions and marsh dynamics from small plots to a national scale", *Ecology*, 2019.

[52] B. Weinstein, "A computer vision for animal ecology", *Jour. Animal Ecology*, 2018.

[53] ——, "Scene-specific convolutional neural networks for video-based biodiversity detection", *Methods in Ecology and Evolution*, 2018.

[54] B. Weinstein, S. Marconi, S. Bohlman, A. Zare, and E. White, "Individual tree-crown detection in rgb imagery using semi-supervised deep learning neural network", *Remote Sensing*, 2019.

[55] "What learning systems do intelligent agents need? complementary learning systems theory updated", *Trends in Cognitive Sciences*, vol. 20, no. 7, pp. 512–534, 2016, ISSN: 1364-6613. DOI: https://doi.org/10.1016/j.tics.2016.05.004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364661316300432.

[56] I. Williams, C. Couch, O. Beijbom, T. Oliver, B. Vargas-Angel, B. Schumacher, and R. Brainard, "Leveraging automated image analysis tools to transform our capacity to assess status and trends of coral reefs", *Frontiers in Marine Science*, 2019.

[57] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks", *Proc. IEEE CVPR*, 2017.

[58] H. Zhou, J. Lan, R. Liu, and J. Yosinski, "Deconstructing lottery tickets: Zeros, signs, and the supermask", in *NeurIPS*, 2019.

[59] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun, "Graph neural networks: A review of methods and applications", *ArXiv*, vol. abs/1812.08434, 2018.