remote sensing

MDPI

Article

Review and Evaluation of Deep Learning Architectures for Efficient Land Cover Mapping with UAS Hyper-Spatial Imagery: A Case Study Over a Wetland

Mohammad Pashaei [1], Hamid Kamangir [1] and Michael J. Starek [1,2,*] and Philippe Tissot [2]

[1] Department of Computing Sciences, Texas A&M University-Corpus Christi, Corpus Christi, TX 78412, USA; mpashaei@islander.tamucc.edu (M.P.); hkamangir@islander.tamucc.edu (H.K.)

[2] Conrad Blucher Institute for Surveying and Science, Texas A&M University-Corpus Christi, Corpus Christi, TX 78412, USA; philippe.tissot@tamucc.edu

* Correspondence: michael.starek@tamucc.edu

Received: 31 January 2020; Accepted: 4 March 2020; Published: 16 March 2020

Abstract: Deep learning has already been proved as a powerful state-of-the-art technique for many image understanding tasks in computer vision and other applications including remote sensing (RS) image analysis. Unmanned aircraft systems (UASs) offer a viable and economical alternative to a conventional sensor and platform for acquiring high spatial and high temporal resolution data with high operational flexibility. Coastal wetlands are among some of the most challenging and complex ecosystems for land cover prediction and mapping tasks because land cover targets often show high intra-class and low inter-class variances. In recent years, several deep convolutional neural network (CNN) architectures have been proposed for pixel-wise image labeling, commonly called semantic image segmentation. In this paper, some of the more recent deep CNN architectures proposed for semantic image segmentation are reviewed, and each model's training efficiency and classification performance are evaluated by training it on a limited labeled image set. Training samples are provided using the hyper-spatial resolution UAS imagery over a wetland area and the required ground truth images are prepared by manual image labeling. Experimental results demonstrate that deep CNNs have a great potential for accurate land cover prediction task using UAS hyper-spatial resolution images. Some simple deep learning architectures perform comparable or even better than complex and very deep architectures with remarkably fewer training epochs. This performance is especially valuable when limited training samples are available, which is a common case in most RS applications.

Keywords: coastal wetland; land cover mapping; semantic image segmentation; machine learning; deep learning; convolutional neural networks; transfer learning; unmanned aircraft systems

1. Introduction

Remote sensing (RS) is the major source of spatial information related to the earth's surface, offering a wide range of sensors and platforms to monitor land cover and its spatial distribution. Recently, Unmanned Aircraft Systems (UASs) are widely employed in numerous RS applications including natural resource management [1–3]. In comparison with traditional RS, UAS technology stands out for its low-cost operation and ability to acquire image data with high spatial and temporal resolution in a flexible fashion at local scales. UAS usually flies at low altitudes and captures high spatial resolution (few cm to sub-cm) images. In combination with the recent advancement in image analysis algorithms, those high-quality images may significantly improve the overall accuracy of image-derived products in many different RS tasks. For instance, pixel-level labeling, which is frequently used in computer vision

tasks such as semantic image segmentation and instance segmentation, is eminently applicable to UAS hyper-spatial resolution imagery. Semantic image segmentation refers to the process of associating each individual pixel of an image with a predefined class label [4]. On the other hand, instance segmentation refers to the task that treats multiple objects of the same class as distinct individual objects (instances) [5].

Wetlands are known as one of the most important ecosystems on our planet. They can be characterized as transitional areas between permanently flooded deep water environments and well-drained highlands, where the water table is usually at or near the surface and the land is inundated by shallow water [6]. Coastal wetlands are important as highly dynamic natural ecosystems offering remarkable services essential to people and the environment including, wildlife habitat for myriad species of marine and terrestrial plants and animals, storm protection, erosion control, nutrient filtering, and recreation as tourist stops. These services are estimated to value at billions of dollars [7]. Authors in [8] highlight the need for monitoring wetland vegetation and its distribution to detect changes in the terrestrial-aquatic transition. Studies show that world wetland loss and degradation has been accelerated for the last three decades mostly due to both anthropogenic and natural factors. According to a report published by the US Fish and Wildlife Service (FWS) and the National Oceanic and Atmospheric Administration's (NOAA) National Marine Fisheries Service (NMFS), a net loss of about 361,000 acres of coastal wetlands in the eastern United States occurred between 1998 and 2004 -an average net reduction of 59,000 acres per year [6]. Sustainable management of any dynamic ecosystem requires, among other parameters, a thorough understanding of its different types of land cover.

Coastal wetland classification is challenging because vegetation and other land cover objects modulate with water level fluctuation and other environmental stressors, leading to sometimes rapid and frequent changes in the type and spatial distribution of a certain land cover [9,10]. The ability to accurately and quickly monitor and predict land cover undergoing rapid and seasonal variations in response to changing environmental factors, including seasonal and climate changes, topography, sea-level rise, water temperature, altered flooding and salinity patterns, etc., [11–13], is crucial for updated and/or continuous land cover monitoring systems. Wetland land cover processes as well as other dynamic landscapes are further complicated by the need for frequent data collection methods, and the subsequent demands for faster and automatic algorithms analyzing very high spatial, temporal, and spectral resolution imagery by the monitoring system with the lowest level of human intervention. In particular, achieving such continuous or near-real time land cover monitoring systems becomes more challenging where expert knowledge is required for designing and extracting the most efficient and discriminative features for different states of the land cover due to the change in participating factors. Furthermore, pixel-wise labeling using mere spectral information in natural environments usually gives rise to unsatisfactory results due to higher inter-class spectral similarity and intra-class spectral variability among natural targets [14]. This issue is highlighted especially where high-spatial resolution imagery from a lower spectral resolution sensor is employed for classification, such as consumer-grade digital RGB cameras commonly employed on small UAS for mapping purposes [15]. Moreover, natural targets such as vegetation or water bodies are not usually enclosed by well-defined boundaries in airborne images resulting in more uncertainties in the pixel-wise labeling for the land cover prediction. In addition, due to high spatial autocorrelation among natural targets, the relationship between the target pixel and its neighboring pixels need to be incorporated into subsequent analyses [16,17]. Thus, to take full advantage of the UAS-based high-spatial resolution imagery, image analysis algorithms exploiting spatial, spectral, contextual, and textural information, collectively, are highly recommended for precise land cover prediction [17–21].

Exploiting sophisticated techniques and algorithms along with some level of field operations for ground truthing and results validation are often a few required components for accurate monitoring of the wetland or other natural environments through remote sensing image classification [12,13,19]. In traditional RS classification techniques, pixel-wise classification methods assume each pixel is pure and typically labeled to the most likely land cover category. Object-based image analysis (OBIA) techniques, on the other hand, provided a new paradigm to classify RS images, where, by utilizing

both spectral and contextual image features, it can outperform the pixel-based techniques [14,17]. By exploiting OBIA techniques, geographical objects, instead of individual pixels, form the basic unit for image analysis [14]. Unlike pixel-based analysis, in OBIA, a certain image is segmented into relatively homogeneous and semantically coherent objects based on a predefined homogeneity criteria at different scales [18]. In other words, spectral information is aggregated per object, where other textural and contextual information become available for conducting image classification on objects rather than pixels [22]. Several studies have already shown the higher performance of object-based image classification techniques than pixel-based methods, especially when high-spatial resolution images are employed [14,22,23]. In general, both pixel-wise and OBIA strategies for land cover or land use classification, take advantage of a wide variety of supervised or unsupervised machine learning (ML) classification algorithms [24–29].

In recent years, however, due to the striking achievement of deep learning models in outperforming almost all state-of-the-art techniques in a wide range of applications, the RS community is shifting its attention to deep learning models. The large number of publications exploiting these models in different RS image analyses and the reported accuracies demonstrate the potential of deep learning in this field of study [30–33]. The recent success of deep convolutional neural networks (CNNs) has enabled substantial progress in many image understanding tasks including pixel-wise semantic image segmentation due to a rich hierarchical feature learning process. Hierarchical features are learned through an end-to-end trainable framework in which higher levels of the feature hierarchy are formed by the precise composition of the lower level features [34–37]. Learned features, at multiple levels of abstraction, provide a unified, highly complex mapping function from input to output taking only as input the raw data. Such complex mapping not only considers the spectral information of each individual pixel in the image, but also takes all textural, contextual, and spatial information related to each individual pixel into account. Thanks to the recent rise of transfer learning techniques, it is possible to take a pre-trained deep CNN model, trained over a large dataset in a supervised or unsupervised manner, and leverage high complex mappings learned by very deep CNN models to perform effectively on downstream tasks [38]. In addition, due to exploiting end-to-end trainable models within the deep learning framework, efficient feature engineering, which is the biggest concern for almost all traditional classification techniques, is entirely eliminated. This paves the path for developing fully autonomous and online land cover prediction systems. All these characteristics are extremely important in many image analyses in different RS tasks. Specifically, deep CNN models have been successfully used for RGB, multispectral, and hyperspectral RS image analyses in various applications [39–42]. Very recently, deep CNNs have been specifically applied to wetland studies, including land cover classification. Results and findings confirm where adequate labeled training samples are available, deep CNN models usually outperform the traditional and machine learning classification techniques [3,43–46].

The objectives of this paper include: (1) employing some of the most popular deep CNN architectures extensively used in computer vision community for semantic image segmentation on hyper-spatial resolution UAS images acquired over a coastal wetland for land cover prediction; (2) investigating the feasibility of deep learning architectures and evaluating the performance of different deep CNN models in pixel-wise image labeling where labeled training samples are limited and natural targets that appear in UAS images with high spatial resolution exhibit high complexity in their spectral and textural information without clear borders to distinguish other neighboring targets; (3) identifying a deep learning architecture representing, among others, a high performance CNN model from speed and accuracy points of view which can be effectively used in many RS applications where complex pixel-level analyses on high-spatial resolution imagery are required.

The author should emphasize that a comprehensive study on coastal wetland classification to perform detailed analyses of vegetation or other land cover properties is not the objective of this paper. Furthermore, the study of land cover changes over time in the coastal wetland setting due to changes in participating environmental factors is not a goal at this stage. Nonetheless, due to the complexity of

the coastal wetland setting relative to many other natural environments, in terms of providing higher inter-class spectral similarity and higher intra-class spectral variability, variable target boundaries and spatial distributions, and mixed pixels, this environment has been chosen as a suitable and challenging case study. For evaluating the efficiency of the employed deep CNN models, performance metrics commonly employed for evaluating model performance of semantic image segmentation tasks in computer vision are utilized. These metrics usually take the ground truth images as the existing reality and compare the predicted images with the corresponding ground truth images based on manual labeling of the image data.

The remainder of this paper is organized as follows. Section 2 explains the most popular deep learning architectures in the computer vision community for image understanding tasks and briefly describes transfer learning as a widely used technique to leverage a deep learning architecture trained for a certain task in a different task. Furthermore, different metrics that are usually used in machine learning and deep learning for performance evaluation of a typical classification technique are described at the end of the the section. Section 3 introduces the data collection and data pre-processing steps. It also provides some information about the chosen deep CNN architectures for land cover prediction task and brief details about the employed optimization algorithm and hardware configuration. Sections 4 and 5 report and discuss, respectively, experimental results achieved by implementing some of the most popular deep CNN architectures for pixel-wise labeling on the experimental dataset. Lastly, Section 6 provides conclusions and future work perspectives.

2. Deep Learning for Semantic Image Segmentation

Advancing deep learning architectures to tackle pixel-wise image labeling is a natural step in the progress from coarse to fine inference [4]. The origin of convolutional neural networks could be located at handling classification tasks where a certain category was predicted for the entire image [47]. Target localization and detection in computer vision tasks was the next necessary step towards fine-grained inference providing further information, other than classes. Instance segmentation which joins detection and segmentation is an additional improvement towards fine-grained inference [48]. Fully Convolutional Network (FCN) [4] is considered a milestone in transforming classification-purposed CNNs for semantic image segmentation by replacing fully connected layers with convolutional ones to output spatial maps instead of classification scores. Moreover, to compensate for low resolution prediction maps due to several down-sampling steps within pooling layers, FCN includes several fractionally-strided convolutions, also known as deconvolutions or transposed convolution [49,50], combined with a simple bilinear or any learnable interpolation allowing per-pixel labeled output. FCN can be trained end-to-end to efficiently learn to predict pixels' categories for an image of arbitrary size. This approach achieved significant improvement over traditional methods on the PASCAL Visual Object Classes (VOC) [51] standardized image dataset with high efficiency at inference time. Despite its simplicity and flexibility, FCN architecture suffers from some critical limitations when it is applied for certain applications. FCN has a fixed receptive field which makes the network unable to capture contextual information appropriate for pixel-wise labeling for objects that are substantially smaller or larger than the predefined fixed receptive field [37]. As a result, predictions are more uncertain for local ambiguous regions. Feature maps that are used for prediction in several layers of the CNN architecture have contextual information appropriate for the classification task, not the pixel-wise labeling. Additionally, the entire network is usually trained to be spatially invariant, which does not let the network take useful global context information into account. Furthermore, the network suffers from lack of instance-awareness which is very important in some image understanding tasks [37].

Since the introduction of FCN in 2015, a wide range of research has focused on how to provide dense segmentation maps with pixel-level accuracy from arbitrary sized images. Recently introduced deep learning architectures owe their high performances in precise semantic segmentation to several factors including:

1. introduction of more advanced and deeper CNN feature encoders that are efficiently trained using recently developed advanced optimization algorithms.
2. utilizing a more advanced decoding strategy to the final low-resolution encoded feature maps in an encoder–decoder architecture using deconvolution or dilated convolution to efficiently increase their resolution for pixel-wise prediction.
3. using the skip connection to introduce low-level abstract information to the high-level abstract information to build highly accurate feature maps representing pixel-level feature information.

2.1. Feature Encoders

Feature encoders are simply described as a stack of convolution layers in combination with activation functions, usually *Rectified Linear Unit* (*ReLU*) [52] , and pooling layers, usually Max-Pooling, which construct a hierarchical representation of the input data containing low-level to high-level abstract information [50]. LeNet [47] is considered as the fist CNN-based feature encoder introduced by LeCun et al. in 1998. However, AlexNet [53], the first deep CNN architecture, introduced by Alex Krizhevsky in 2012 is a landmark in deep learning history. Several key factors are contributing in this progress: (1) the efficient training procedure implemented on the modern GPUs [53], (2) the proposal of the *ReLU* activation function, which had significant contribution in boosting training and made convergence much faster, and (3) the availability of a huge dataset, e.g., ImageNet [54] to train models with high capacity which include millions of trainable parameters. VGG-Net [55], GoogLeNet [56], Residual Network (ResNet) [57], and Densely Connected Network (DensNet) [58] are a few examples of popular architectures that are frequently employed for feature extraction in very deep CNN models.

- ***VGG-Net.*** VGG-Net [55] was invented in 2014 by Oxford's Visual Geometry Group as a successful effort to build and train a very deep CNN. VGG-Net showed that the depth of a network is a critical component in CNNs to achieve high performance in recognition or classification. By shrinking the convolution kernels to 3×3 yet increasing the number of sequences of convolutional layers and feature maps in each convolution layer, VGG is able to train deeper architecture with appropriate receptive field comparable with AlexNet for recognition tasks.
- ***GoogLeNet.*** GoogLeNet [56] (a.k.a. Inception Net) from Google in 2015 was proposed by Szegedy et al. with the objective of reducing computation complexity compared to the traditional CNNs. Inception module, which makes building block for the network, is a combination of 1×1, 3×3, and 5×5 convolutional kernels and a pooling layer. The motivation behind inception module is to increase the receptive field without losing fine information. By learning and combining features with different scales in parallel in each inception module, GoogLeNet is able to learn feature hierarchy in a multi-scale manner while its innovative architecture reduces the number of trainable parameters in a really deep framework (22 layers) to less than 5 million parameters in comparison to 62 million and 138 million parameters in AlexNet and VGG-Net, respectively. To train a deep stack of inception modules in an efficient way, bottleneck approach is exploited in which extra 1×1 convolutions reduce the dimensionality of feature maps that enter the inception module from the previous layer. This helps to avoid parameter explosion in inception modules and the overfitting problem in the whole network. Figure 1 illustrates the architecture of the inception module. Other versions of inception modules including BN-Inception [59], Inception V2, and Inception V3 [60] were later proposed. In order to increase the efficiency and performance of inception modules, in 2017, Szegedyetal et al. proposed a combined version of inception modules and residual network (ResNet) modules known as Inception-ResNet [61]. Xception [62], which stands for extreme version of inception, was proposed by Chollet et al. in 2017. The motivation behind it is to disjointly map cross-channels and spatial information in feature maps as their correlation is sufficiently decoupled. As a result, the depthwise separable convolutions from inception modules are modified in Xception modules as separable pointwise convolutions follow by depthwise convolutions.

- ***ResNet.*** As mentioned above, deeper networks can improve the performance of deep learning approach to solve complex visual tasks, but they are more prone to the notorious problem of vanishing/exploding gradients during training as well. It may lead to not only saturated accuracy, but also degradation of training accuracy. ResNet [57] designed by He et al. in 2015 exploits residual blocks to overcome the vanishing gradient problem in very deep CNNs by introducing identity shortcut connections to successive convolution layers as shown in Figure 2. The shortcut connections in residual blocks help gradients flow easily in back propagation step which leads to gaining accuracy during the training phase in a very deep network. Referring to Figure 2, each unit calculates a residual function $F(x) = H(x) - x$, in which x is the output of the previous residual unit and $H(x)$ denotes the desired underlying mapping. More precisely, if y_l is the output of the lth residual unit with weights w_l, then

$$y_l = x_l + F(x_l, w_l) \tag{1}$$

$$x_{(l+1)} = f(y_l) \tag{2}$$

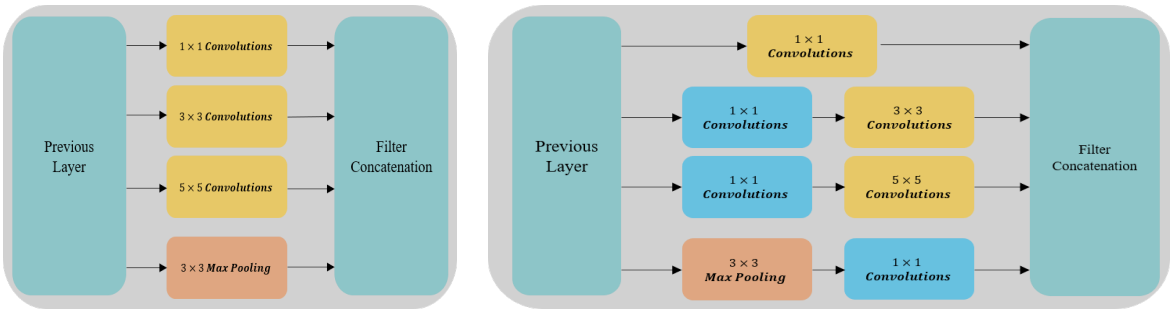
  where $f()$ is the activation function.

  According to Figure 3, different variants of residual unit were proposed, which consists of different combinations of convolutional layers, batch normalization (BN) [59], and rectified linear unit [63] activation function [57,64]. In our experiment, we use the full pre-activation variant of residual unit proposed by He et al. [57,64] to build our architectures, which use ResNet as their feature encoder. ResNeXt [65] proposed by Saining Xie in 2017 is a highly modularized version of ResNet architecture based on split transform aggregate strategy as an inception module for image classification. Its innovative, simple design results in homogeneous, multi-branch architecture that has only a few hyper-parameters to set. This approach exposes a new dimension called cardinality, the size of the set of transformations, as an essential factor in addition to other critical factors such as depth and width. The network is constructed by stacking repeating building blocks that aggregate a set of transformations with the same topology. Inspired by a residual network, several modifications, new designs, and architectures were proposed for different image understanding tasks [57,66–68]. For instance, Figure 4 illustrates an inception-ResNet block called Inception ResNet-A module of the Inception ResNet-v2 network [61]. Other variants of inception-ResNet blocks including Inception ResNet-B and Inception ResNet-C modules were also proposed by Szegedy et al. [61] in 2017.
- ***DenseNet.*** Inspired by ResNet and the idea that shorter connections between layers close to the input and those close to the output can help to train substantially deeper CNNs more accurately and efficiently, Huang et al. proposed DenseNet [58] in 2017. The architecture consists of densely connected CNN blocks in which the output feature maps of each layer are concatenated with the output feature maps of all successor layers in a dense block as shown in Figure 5. If lth layer receives all the feature maps from all preceding layers, x_0, x_1, ..., x_{l-1}, as input then:

$$x_l = H_l([x_0, x_1, ..., x_{l-1}]) \tag{3}$$

  where $[x_0, x_1, ..., x_{l-1}]$ represent simple concatenation of feature maps produced in layers $0, 1, ..., l-1$ and H_l is defined as a composite function of three consecutive operations including BN, followed by a *ReLU* and a 3×3 convolution. A transition layer composed of a a batch normalization layer and a 1×1 convolution followed by a 2×2 pooling operation is introduced between two consecutive dense blocks to reduce the dimensionality and spatial resolution of derived feature maps. DenseNet architecture consists of several densely connected blocks and transitional blocks, which are placed between two adjacent densely connected blocks. DenseNet concept alleviates the vanishing gradient problem, encourages feature propagation and feature reuse while substantially reducing network parameters.

- ***MobileNet.*** Since the advancement of deep learning, the general trend has been to make deeper and more complicated networks to improve model performance [55,60,61]. However, these advances to improve accuracy are not necessarily making networks more efficient with respect to size and speed. In many real world applications such as self-driving car, robotics, and augmented reality, the timely-fashioned or almost real-time prediction and recognition tasks need to be carried out on a computationally limited platform. Inspired by depthwise separable convolutions [69] to reduce the computation in the first few layers, a class of efficient models, called MobileNets [70,71], for mobile and embedded vision applications was introduced by Howard et al. in 2017. This class of models presents a streamlined-base architecture that uses depthwise separable convolutions to build lightweight deep neural networks. According to Figure 6, the depthwise separable convolution is a form of factorized convolutions factorizing a standard convolution into a depthwise convolution, which applies a single filter to each input channel, and a 1×1 convolution called a pointwise convolution to change the dimensions and linearly combine the output feature maps from depthwise convolutions. The depthwise separable convolution technique results in a drastic reduction in computation complexity and model size. Figure 7 illustrates two variants of MobileNet architectures. According to Figure 7, in MobileNetV1 [70], there are two layers including depthwise and pointwise convolutions. M and N are the number of input and output channels, respectively, and D_F and D_K are the sizes of feature maps and filter size, respectively. BN and *ReLU* activation function are both applied after convolutional layers. MobileNet introduces two hyper-parameters to the network including width multiplier, $\alpha \in (0, 1]$, to control the input width of a convolutional layer and resolution multiplier $\rho \in (0, 1]$, to control the input image resolution of the network. $\alpha = 1$ and $\rho = 1$ are hyper-parameters for the baseline MobileNets and $\alpha < 1$ and $\rho < 1$ are considered for any reduced computation MobileNets. Computational cost and the number of parameters are reduced by roughly α^2. However, the accuracy drops off as α and ρ decrease. MobileNetV2 [71] is a significant improvement over MobileNetV1 with high potential of reaching the state-of-the-art performance for mobile visual recognition tasks. It was also built upon the idea of depthwise separable convolution already applied in MobileNetV1 as efficient building blocks. In MobileNetV2, there are two types of blocks. One block is a residual block with stride of 1 and a second block with stride of 2 for downsampling. Both blocks include three layers. The first layer of each block in MobileNetV2 includes a 1×1 convolution with *ReLU* activation function. The second layer is a depthwise convolution, and the third layer is another 1×1 convolution but without any activation function.



(**a**) Naive Inception. (**b**) Inception V1.

Figure 1. Inception modules.

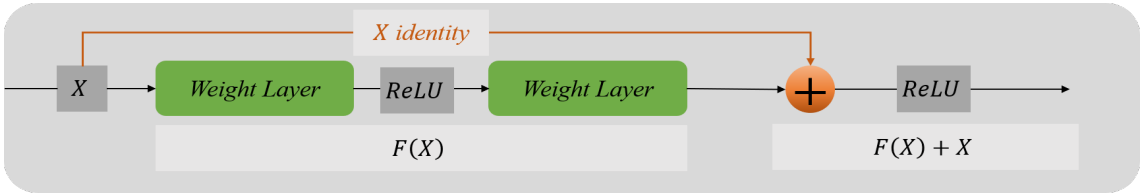


Figure 2. Basic diagram of residual unit.

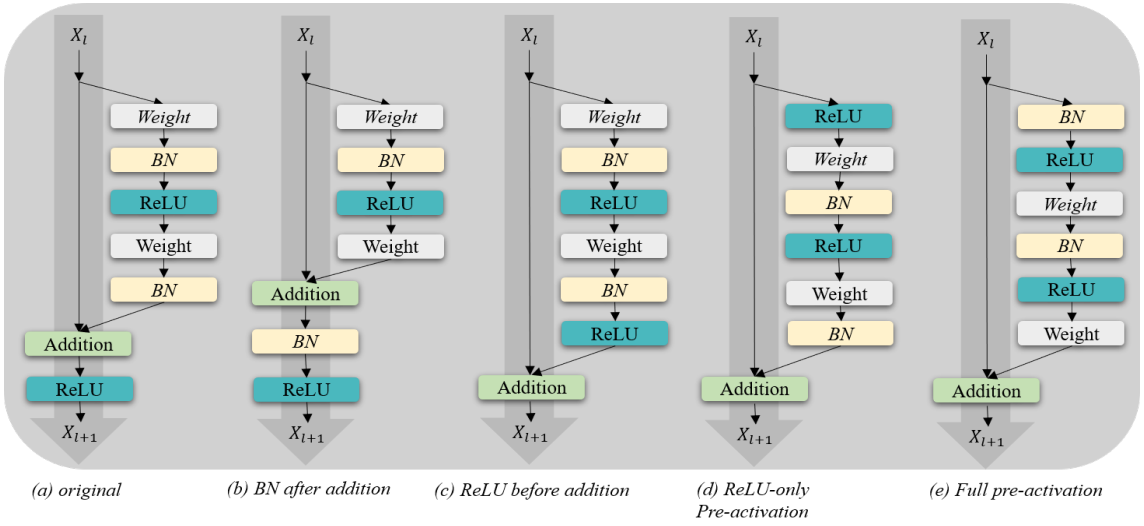


Figure 3. Different variants of residual units.

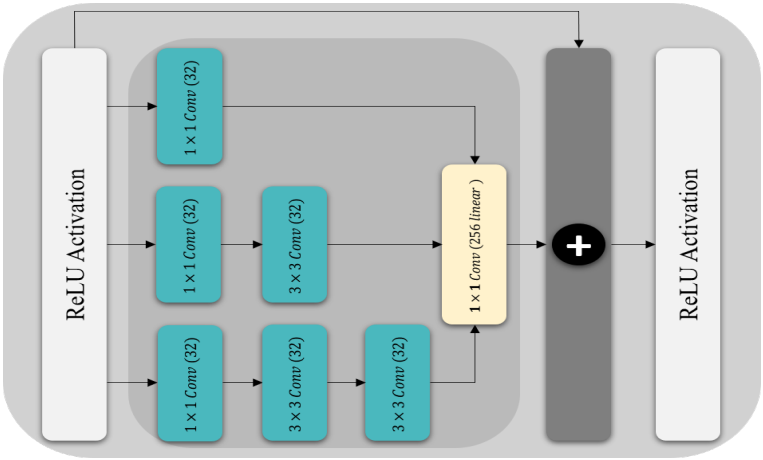


Figure 4. Inception-ResNet block.

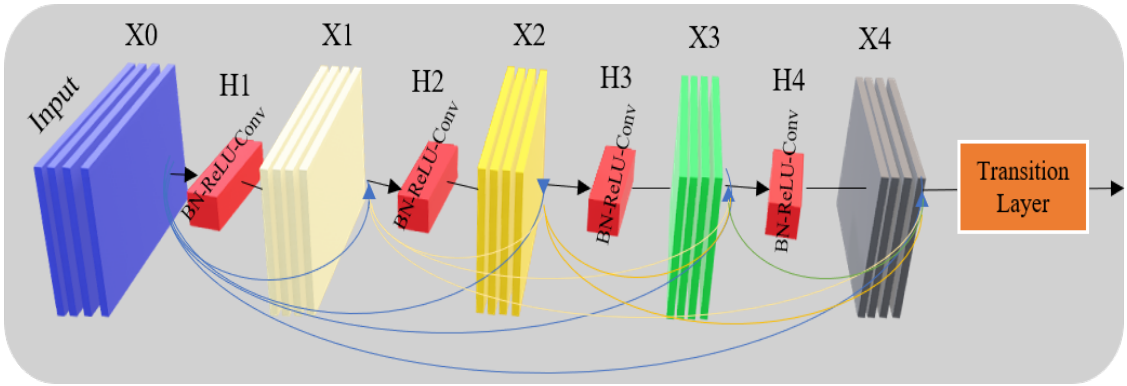


Figure 5. A 5-layer dense block with a growth rate of k = 4.

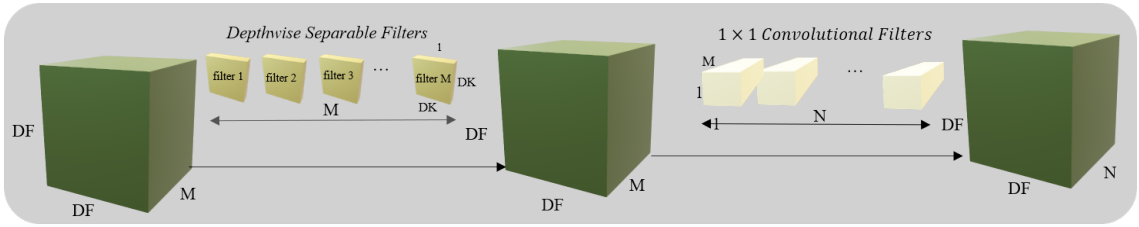
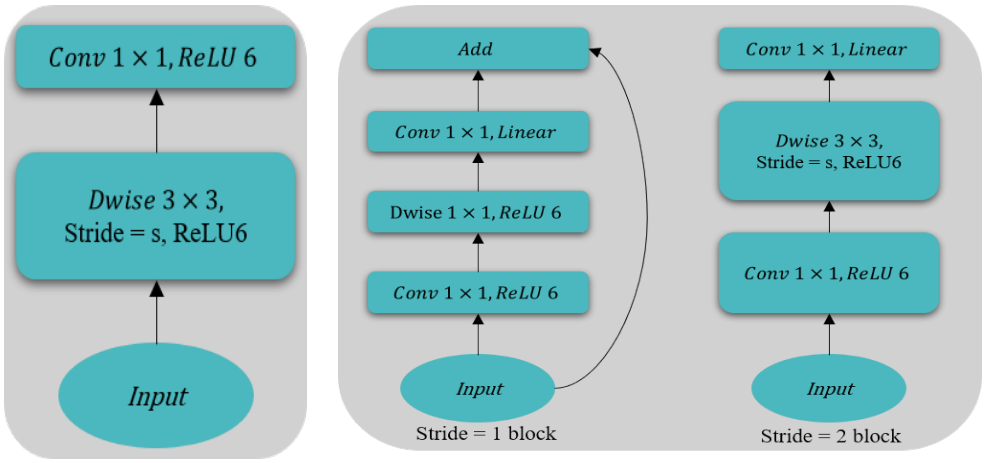


Figure 6. Depthwise separable convolution concept.



(**a**) MobileNetV1. (**b**) MobileNetV2.

Figure 7. MobileNet architecture modules.

2.2. Decoding Approaches

As explained earlier, an encoder is simply a deep learning architecture such as VGG-Net, GoogLeNet, ResNet, etc., making a hierarchical representation of input data. The final feature maps derived from encoders are usually coarse representations of the input image which needs to be upsampled to higher resolution feature maps. Decoding, on the other hand, is a strategy that aims to efficiently exploit encoded feature maps provided by the encoder to form an output that is the closest match to the intended output, usually corresponding ground truth. Deconvolution or transposed convolution [49,72] is conceptually required in deep CNN architectures for pixel-wise predictions as feature maps are continuously down-scaled within several convolution and pooling layers. As mentioned earlier, FCN architecture enables upsampled feature maps with resolution comparable to the input image through a fractionally-strided convolution step in combination with a simple bilinear interpolation. However, due to the lack of an efficient trainable deep deconvolution network, FCN fails to achieve the high accuracy in pixel-wise labeling, especially when it is required to reconstruct highly nonlinear structures of object boundaries. [73].

The deconvolution network was first discussed for image reconstruction from its feature representation by Zeiler et al. [50]. To resolve ambiguity induced by Max-pooling layers, the network stored the pooled locations, which need to be retrieved in an unpooling operation. To predict pixel-wise segmentation map, in 2015, Noh et al. proposed a trainable deep deconvolution network composed of deconvolution and unpooling layers [73]. SegNet [74] designed by Badrinarayanan et al. in 2015 consists of a deep encoder network and a hierarchy of decoders—one corresponding to each encoder followed by a pixel-wise classification layer. Appropriate decoders are fed by Max-pooling indices computed in the pooling steps of the corresponding encoder to perform deconvolution with nonlinear upsampling of their input feature maps. To produce dense feature maps in the decoder, the resulting sparse upsampled feature maps are, then, convolved with trainable filters. U-Net [75] developed in 2015 is an innovative deep learning architecture first developed for biomedical image segmentation by Ronneberger et al. and was then extensively used for image segmentation in many other fields

with different encoders such as ResNet, DenseNet, and Inception modules. The network has a symmetrical architecture characterized by an encoder with a series of convolution and Max-pooling layers in the contracting path and a decoder containing a mirrored sequence of convolution and upsampling layers in the expanding path of the network. U-Net is able to concatenate low level abstract information, extracted from the first convolutional layers of the encoder (contracting path) and high level semantic abstraction information, extracted from the final layers of encoder, in the decoder (expanding path), resulting in a finer and more accurate prediction map. This strategy resulted in high performance especially when only a limited training dataset is available [75]. Motivated by a Laplacian pyramid developed for compact image coding [76], in 2016, Ghiasi et al. proposed a network called Laplacian Pyramid Reconstruction (LRR) in which low-resolution feature maps are used to reconstruct a low-frequency segmentation map. Feature maps are, then, refined by adding high-frequency details. Refinement network (RefineNet) [77], proposed by Lin et al. in 2017, is a generic multi-path network which explicitly exploits all available information along the downsampling path to enable high-resolution image labeling using long-range residual connections. This network consists of three main components: Residual convolution unit (RCU), which exploits features at multiple scales, multi-resolution fusion, which merge multi-resolution features, and chained residual pooling, which aims to capture background context from a large image region by fusing the output feature maps of all pooling blocks together with the input feature map.

Inspired by DenseNet, in 2017, Jegou et al. proposed a One Hundred Layers Tiramisu network, commonly called Fully Convolutional DenseNet (FC-DenseNet) [78]. The architecture extends the DenseNet to a fully convolutional network for a semantic segmentation task. The upsampling path includes convolution, upsampling operations called transition up, and skip connections. Transition up modules consist of a transposed convolution to upsample the previous feature maps. Upsampled feature maps are then concatenated with corresponding feature maps in the downsampling path using skip connections to prepare the input for the next upsampling dense block. To mitigate the parameter explosion problem, the input of a dense block is not concatenated with its output in the upsampling path e.g., transposed convolution is applied only on feature maps derived by the last dense block instead of the concatenation of all derived feature maps so far.

Other innovative techniques were also proposed for dense semantic segmentation, which, unlike the convolution/deconvolution design, do not introduce new parameters to upsample feature maps. Atrous convolution [79,80], usually called dilated convolution, originally developed for computing undecimated wavelet transform (UWT) [81] is employed to effectively enlarge the field of view of feature maps without increasing the number of parameters or computation complexity. Atrous or dilated convolution in the context of CNNs aims for expanding the receptive field of the network. They generate high-resolution feature maps capturing multi-scale contextual information from the input data. Dilated convolution introduces a new hyper-parameter called dilation rate to the convolution layers, which specifies the expansion rate of receptive field enabling the network to exploit a larger receptive field without losing spatial information.

In 2014, DeepLab [79], introduced by Chen et al. from Google, proposes atrous convolution instead of deconvolution for feature upsampling. Atrous convolution offers an efficient mechanism to control the receptive field of the network and finds the best trade-off between precise localization, with the small receptive field, and context assimilation, with the large receptive field. The output of the network is interpolated, with bilinear interpolation, and goes through the fully connected conditional random fields (CRF), which fine-tune the result for a more accurate and detailed segmentation map. Different variants of DeepLab architecture were later proposed with some modification on the original network. Atrous Spatial Pyramid Pooling (ASPP) was proposed in DeepLabV2 [34] to robustly segment objects at multiple scales. ASPP probes incoming feature maps at multiple sampling rates and field-of-views capturing objects and image context in multiple scales. In DeepLabV3 [82], to handle the problem of multi-scale object segmentation, a cascade or parallel atrous convolution design is employed to capture multi-scale context by adopting multiple dilation rates. DeepLabV3 outperformed its predecessors

without dense CRF post-processing and attained comparable performance with other state-of-the-art models. Authors in DeepLabV3+ [83] decided to add a decoder module to the former variant in which the encoded features are first upsampled by a factor of 4, instead of 16 as in [82], and then the resulting feature maps were concatenated with corresponding mid-level features from the network backbone. Moreover, to reduce computational complexity, they adopted the Xception module [62] and applied depthwise separable convolution to both the ASPP and decoder.

Yu et al. [80] developed a deep learning architecture in 2015 specifically designed for dense prediction based on dilated convolution concept [80]. This convolutional network module combines multi-scale contextual information without losing spatial resolution. Pyramid scene parsing network (PSPNet) [84] introduced in 2017 exploits the capability of global context information by different region-based context aggregation methods by employing a pyramid pooling module in combination with the proposed pyramid scene parsing network. To do pixel-wise prediction, PSPNet extends pixel-level feature to a specially designed global pyramid pooling one. Then, the local and global clues jointly form the final prediction.

2.3. Transfer Learning

The idea of transfer learning was motivated by the fact that people can intelligently apply knowledge previously learned to solve a task in one domain to solve a new problem in the same or different domain [85]. In the deep learning context, features learned by a CNN architecture to solve a problem in a certain domain are reusable for solving problems in some other domains, as the first layers of the network in related domains usually tend to learn the same sorts of features. Transfer learning is a highly practical approach to tackle the issue of training a very deep architecture where a limited supply of target training data is available. This could be due to data scarcity, or methods to collect and label the data may be time consuming and expensive requiring expert knowledge. In contrast to many computer vision tasks that can take advantage of thousands of freely available images related to the underlying task, in most RS applications, e.g., land cover mapping, satellite or aerial imagery missions can be very expensive or time consuming. Data collections are a function of many participating factors including flight height, ground sampling distance (GSD), environmental conditions at the time of observation, and camera/sensor settings. Furthermore, a limited number of aerial images are acquired in every flight mission and the acquired images are not always available to the public to enable generation of large labeled data repositories for a specific type of environment or land cover. UAS provides a cost effective and flexible means to collect high-resolution aerial imagery over localized geographic extents; however, dense repositories of UAS imagery acquired over a specific type of natural environment that is expertly labeled for training deep CNNs to perform land cover prediction are presently non-existent.

Common practice in transfer learning is to copy the whole or just the first n layers of a pre-trained network, already trained on a huge dataset, to exploit them in a new task and then back-propagate the errors from the new task into the copied features to fine-tune them to the new task. In another approach, especially where the training sample size is significantly limited or the new task is closely related to the task from which a transferred feature is derived, the first n feature layers can be left frozen, meaning that they do not change during training on the new task. The choice of whether or not to fine-tune the copied first n feature layers depends on the size of the available dataset for the new target task. In a case where the target dataset is small, fine-tuning may lead to overfitting, especially when the network contains a large number of parameters. On the other hand, if the target dataset is rich enough or the number of network's parameters is small, where overfitting does not seem to be a problem, then fine-tuning copied features to the new task can highly improve the performance [38]. In such case, training the network from scratch may also be taken as an option.

2.4. Performance Metrics

This section describes the most common performance or evaluation metrics used in the context of semantic image segmentation. Usually, overall performance of a deep learning architecture in semantic image segmentation task is described in terms of overall accuracy of pixel-wise labelling, time, and memory usage. Overall accuracy of a network is a measure which usually describes the correctness of labelling as a simple ratio representing the number of correctly classified pixels over the total number of manually classified pixels in the ground truth. Pixel-wise or per-class accuracy is another measure that usually aims to report the percent of correctly classified pixels for each individual class. Pixel-wise accuracy is closely related to overall accuracy. In fact, binary mask employed in pixel-wise accuracy assessment may return quantities more than just true positive (TP), which represents the number of correctly labeled pixels, and true negative (TN), which represents the number of pixels that are correctly identified as not belonging to a certain class. False positive (FP) represents the number of pixels belonging to other classes misclassified as the target class, and false negative (FN) represents the number of pixels that belong to the target class but are misclassified as belonging to other classes. They are two of the most important quantities for which the binary mask may be designed to account. Accordingly, the overall accuracy per-class can be formulated as [86]:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{4}$$

Pixel-wise accuracy metric is not reliable and may provide misleading results when a certain class representation is small within the whole dataset. Precision and recall are two metrics that can help to interpret the overall accuracy of each class more accurately even in the case of unbalanced classes. Precision or positive predictive value (PPV) describes the purity of positive detection procedure relative to all pixels that have already been truly classified in the ground truth [86]:

$$Precision = \frac{TP}{(TP + FP)} \tag{5}$$

Recall, or true positive value (TPV), on the other hand, effectively describes the completeness of the positive predictions relative to all pixels that have already been truly classified in the ground truth [86]:

$$Recall = \frac{TP}{(TP + FN)} \tag{6}$$

The F-score is a widely used performance metric for classification and segmentation tasks, which consists of the harmonic mean of precision and recall metrics [86]:

$$F - \beta = \frac{(\beta^2 + 1)TP}{(\beta^2 + 1)TP + \beta^2 FN + FP} \tag{7}$$

where B is a scaling factor between the precision and recall. F1 score, one of the more widely used F-measure metrics is formulated by setting $B = 1$ [86]:

$$F1 = \frac{2TP}{2TP + FN + FP} \tag{8}$$

Intersection over Union (IoU), also known as Jaccard index, is a standard performance measure for the object category segmentation. IoU measure represents the similarity ratio between the predicted region and the corresponding ground truth region for an object presented in the dataset [87]:

$$IoU = \frac{TP}{FP + TP + FN} \tag{9}$$

Mean Intersection over Union (mIoU) is a common performance metric for semantic segmentation that is calculated by averaging over all IoU values computed for all existing semantic classes. Other performance metrics, such as time, memory, and power, are highly dependent on the available hardware, software, and the specific deep learning architecture chosen for solving a classification task. Providing such metrics becomes more crucial when a deep learning framework is employed in online applications such as autonomous driving and mobile systems where the memory and power is more limited.

3. Materials and Methods

3.1. Study Site

The study site is a coastal marsh located on a barrier island along the southern portion of the Texas Gulf Coast, USA, bounded by Corpus Christi Bay, the Laguna Madre, and the Gulf of Mexico called the Mustang Island Wetland Observatory as shown in Figure 8. The study area as imaged by the UAS is 11 hectares. Elevation within the wetland slopes gradually and is nearly flat, with the highest elevation in the study area at about 0.8 m (NAVD88). The wetland is located on the bay side of the island Figure 8 and is oriented in a northeast to southwest trend, with the Gulf of Mexico located to the east and Corpus Christi Bay to the west. The dominant vegetation species are *Schizachyrium littorale* (Nash) (coastal bluestem) and *Spartina patens* (Aiton) (gulf cordgrass) commonly found growing in mats. The second most prevalent environment of this study area is tidal flat; it ranges in elevation from −0.05 m to 0.5 m (NAVD88) [88]. Low regularly flooded tidal/algal flats are significantly less abundant than high flats in this area. These local tidal flats are designated as wind-tidal flats because flooding occurs mainly due to wind-driven tides [89,90]. Blue-green algae can be prevalent in the lower portion of the tidal flats after long periods of inundation. Furthermore, salt marsh vegetation can be found sparingly in portions of the tidal flat areas. Low marsh areas are very high in biologic productivity usually ranging in elevation from −0.1–0.3 m (NAVD88). More frequently inundated areas near tidal creeks are dominated by taller vegetation, primarily *Avicennia germinans* (L.) (black mangrove). High marsh environment is the least abundant in the study area imaged by the UAS. It varies in range from approximately 0.2–0.8 m (NAVD88) well above the mean high tide; therefore, it is rarely inundated. These characteristics briefly illustrate the highly dynamic and complex nature of the coastal wetland and the need for applying accurate algorithms for detailed land cover mapping through analyzing UAS hyper-spatial imagery.

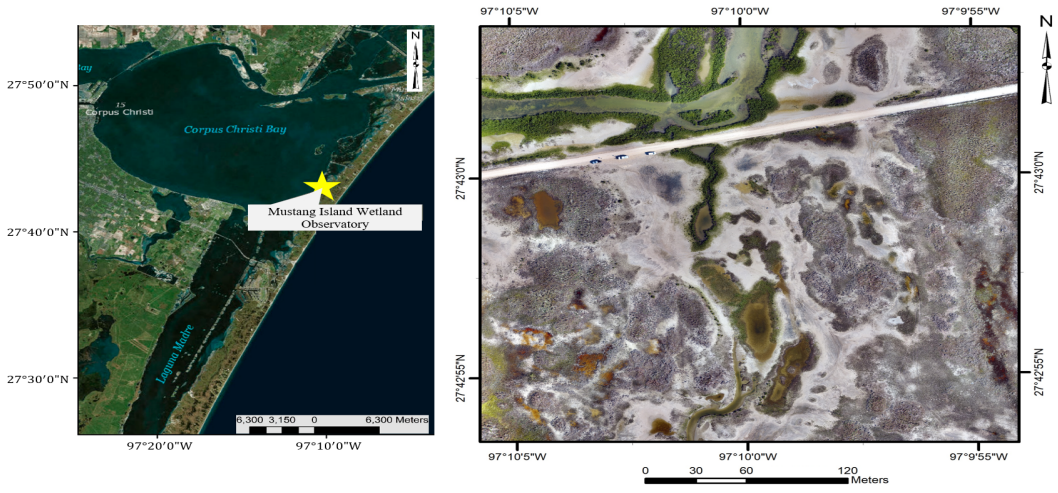


Figure 8. Left, Mustang Island Wetland Observatory study site location; Right, UAS orthoimage of the study area showing the dirt road, exposed tidal flats, water bodies, and surrounding vegetated land cover including mangroves.

3.2. Data Collection and Preparation

Phantom 3 multi-rotor UAS, manufactured by Shenzhen DJI Sciences and Technologies Ltd (SZ DJI Technology Co., Ltd.) headquartered in Shenzhen, Guangdong province, China, was employed to collect required images for this study. This platform is equipped with a CMOS RGB sensor to capture 12 megapixel images with a resolution of 4000 × 3000 pixels. The flight was designed at an altitude of 90 m above the ground resulting in an average GSD of around 3 *cm*. Imagery was collected at 80% sidelap and endlap flown in a grid pattern with parallel flight lines and a 90-degree (nadir) camera orientation. This high amount of overlap was used to perform Structure-from-Motion (SfM) photogrammetry processing and orthorectify the imagery to remove perspective and relief distortion and generate a large orthoimage that covers the study area. The performance and visual quality of land cover prediction using different deep CNN models is evaluated on a certain part of the study area that most original images belonging to that area are kept for validation purpose. Because in RS applications, land cover is usually predicted on orthorectified images, the visual quality of land cover prediction is illustrated on an orthoimage mosaic of validation images. The reader is referred to [90,91] for more details on SfM photogrammetry.

In this work, 300 images were manually selected from the total set of acquired UAS images (about 500 images) that cover the whole study area to reduce repetitive information from image overlap. Due to the high resolution of the original imagery, the image set can rapidly exhaust the whole GPU's memory when directly fed to any deep convolutional network. Therefore, we randomly extract 10,000 image patches of resolution 512 × 512 pixels from the set of 300 raw images. From those image patches, 1000 image patches are held as a validation data set for evaluating the model performance after each training epoch. Every image, at most, represents four classes: vegetation, water bodies, tidal flat, and road. In our experiment, tidal flat is assigned to surfaces exposed within intertidal areas. All temporarily flooded areas or permanently submerged lands are considered water bodies. Areas covered by any type of vegetation is called vegetated area. Finally, road represents the artificially elevated dirt surface of exposed ground that has not been affected by tides. The different land cover classes can be observed in the orthoimage mosaic displayed in Figure 8, which was generated from all UAS images acquired over the study area using the SfM photogrammetry software. All needed ground truth data for training and validation were manually prepared through supervised labeling by interpretation and delineation of land cover boundaries in the image patches. This was done by color labeling of all existing pixels in each original image patch to a representative class using a labeling app in MATLAB software for pixel-level image labeling. According to our predefined color for each target, pixels belonging to vegetation, tidal flat, water, and road are represented by green, orange, blue, and brown, respectively. It should also be mentioned that a set of 64 raw UAS images from a portion of the study site that had a representative distribution of the land cover classes were set aside for independent evaluation of model performance results as presented below in Section 4. These images, or patches extracted from them, were not used as part of the training set described above.

3.3. Deep Learning Architectures

This subsection introduces the deep learning architectures evaluated in this study for performing pixel-wise image segmentation task (i.e., land cover mapping) with UAS hyper-spatial imagery acquired over a complex coastal wetland environment. The chosen architectures are extensively used in a wide range of applications beyond RS including computer vision and medical image processing.

- ***Encoder–Decoder (SegNet).*** SegNet architecture, displayed in Figure 9, is examined in this study, which is a relatively old deep learning network for semantic image segmentation task. It uses VGG network as its encoder to hierarchically extract features from input images. The encoder network consists of 13 convolutional layers corresponding to the first 13 convolutional layers in the VGG-16 network. In our experiment, we use weights from pre-trained VGG-16 network

to initialize the training process. Each encoder layer has a corresponding decoder layer that upsamples the feature maps by using the stored pooled indices.

- ***U-Net.*** U-Net is a famous deep architecture based on an encoder–decoder principle that instead of using pooling indices, it transfers and exploits the entire feature maps from encoder to decoder. Upsampling strategy can have a great impact on the final accuracy of pixel-wise image classification. Comparing the performance of SegNet and U-Net architecture can tell us more about the effectiveness of those two upsampling strategies. Figure 10 illustrates U-Net architecture with ResNet-34 network for feature extraction in this study.
- ***FC-DenseNet.*** To explore the efficiency of DensNet architecture in feature learning for pixel-wise classification of coastal wetland images, the one hundred layer tiramisu model (FC-DenseNet), as shown in Figure 11, is employed which uses 56 convolutional layers, with four layers per dense block and a growth rate of 12. Similar to U-Net architecture, FC-DenseNet exploits U-shape encoder–decoder structure with skip connections between the downsampling and the upsampling paths to add higher resolution information to the final feature map. Unique characteristics of feature reuse, compactness, and substantially reduced number of parameters in FC-DenseNet architecture is evaluated in our experiment based on its performance when training the network from scratch using a limited dataset, which is the case here.
- ***DeepLabV3+.*** Effectiveness of ASPP to encode multi-scale contextual information in images acquired over complex coastal wetland is investigated by examining DeepLabV3+ architecture illustrated in Figure 12. This architecture is able to perform several parallel atrus convolution with different rates.
- ***PSPNet.*** As illustrated in Figure 13, PSPNet, which uses pyramid pooling module for more reliable prediction, is also investigated for this study. Specifically, this module is able to extract global context information through aggregating different regional context information.
- ***MobileU-Net.*** Considering the idea of depthwise separable convolution in MobileNet and feature map upsampling in U-Net architecture, MobileU-Net architecture, illustrated in Figure 14, is implemented in this study. The performance of this architecture in pixel-wise image labeling of hyper-spatial UAS images may give us an estimation of the accuracy achievement in real-time land cover mapping.

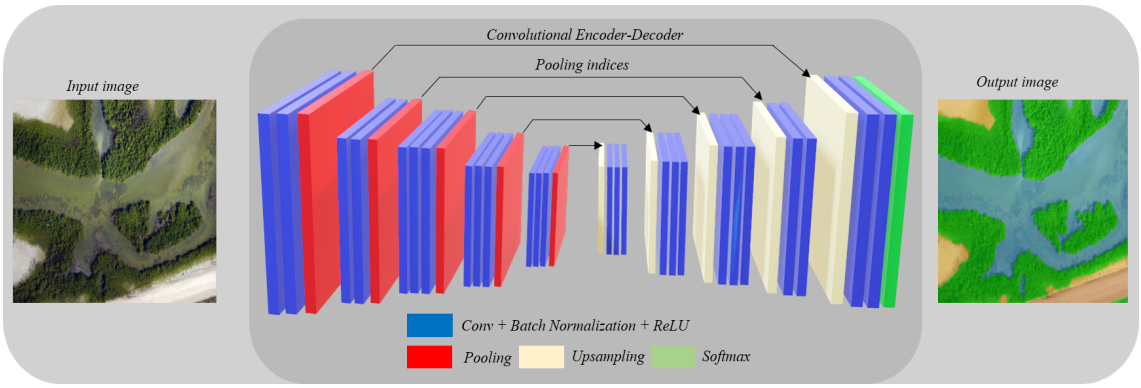


Figure 9. An illustration of the Encoder–Decoder (SegNet) architecture.

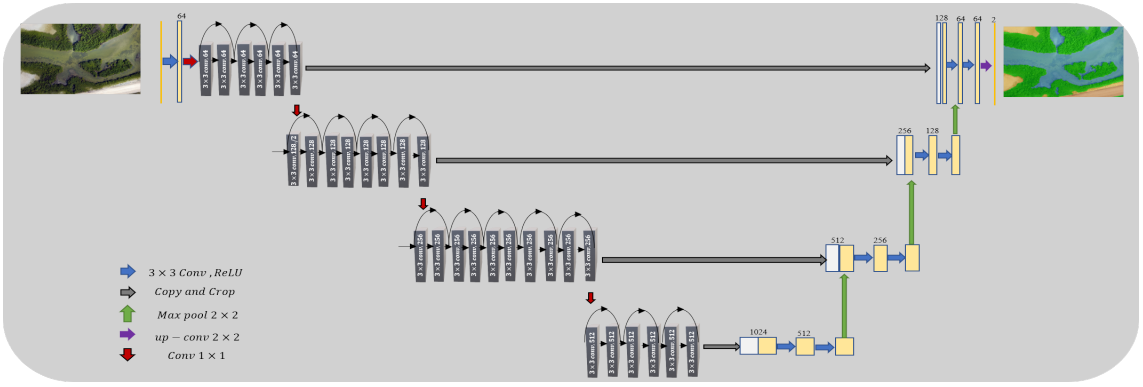


Figure 10. An illustration of U-Net architecture with ResNet34 as its encoder.

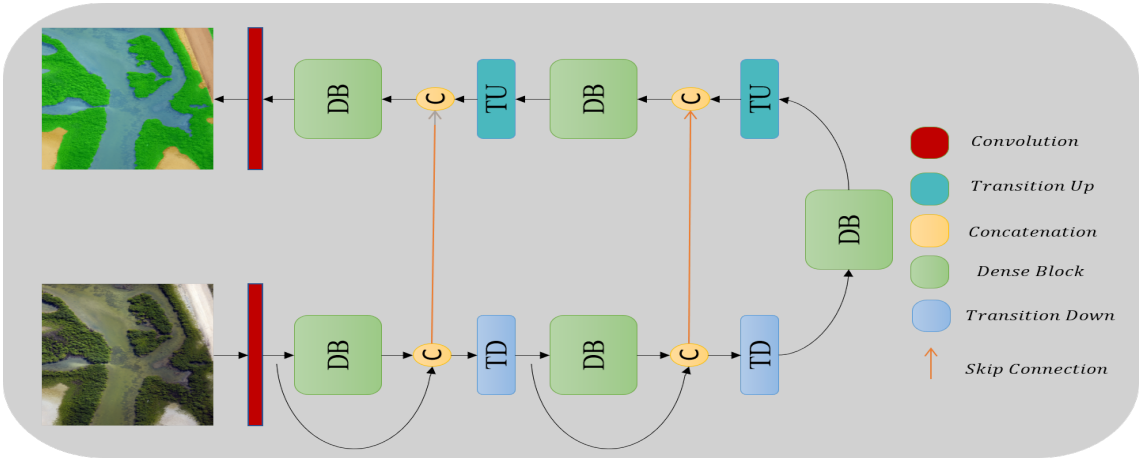


Figure 11. An illustration of the One Hundred Layers Tiramisu (FC-DenseNet) architecture.

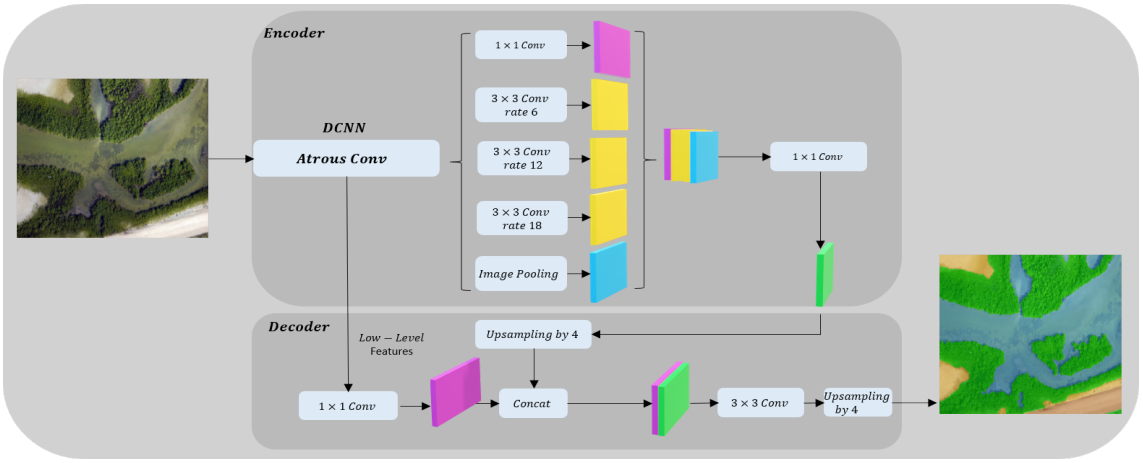


Figure 12. An illustration of DeepLabV3+ architecture.

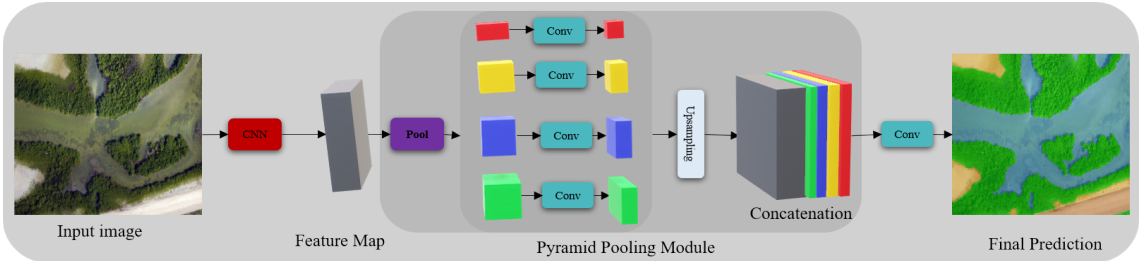


Figure 13. An illustration of PSPNet architecture.

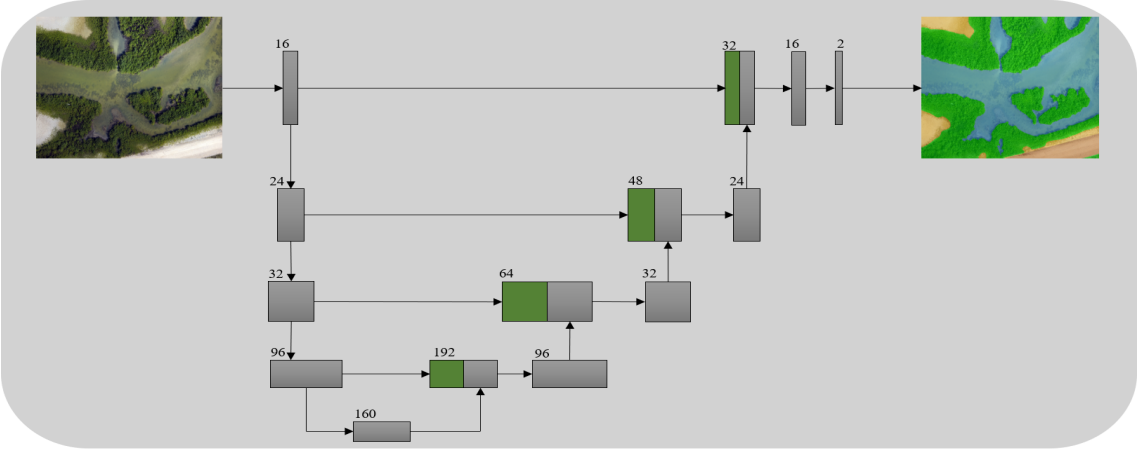


Figure 14. An illustration of MobileU-Net architecture.

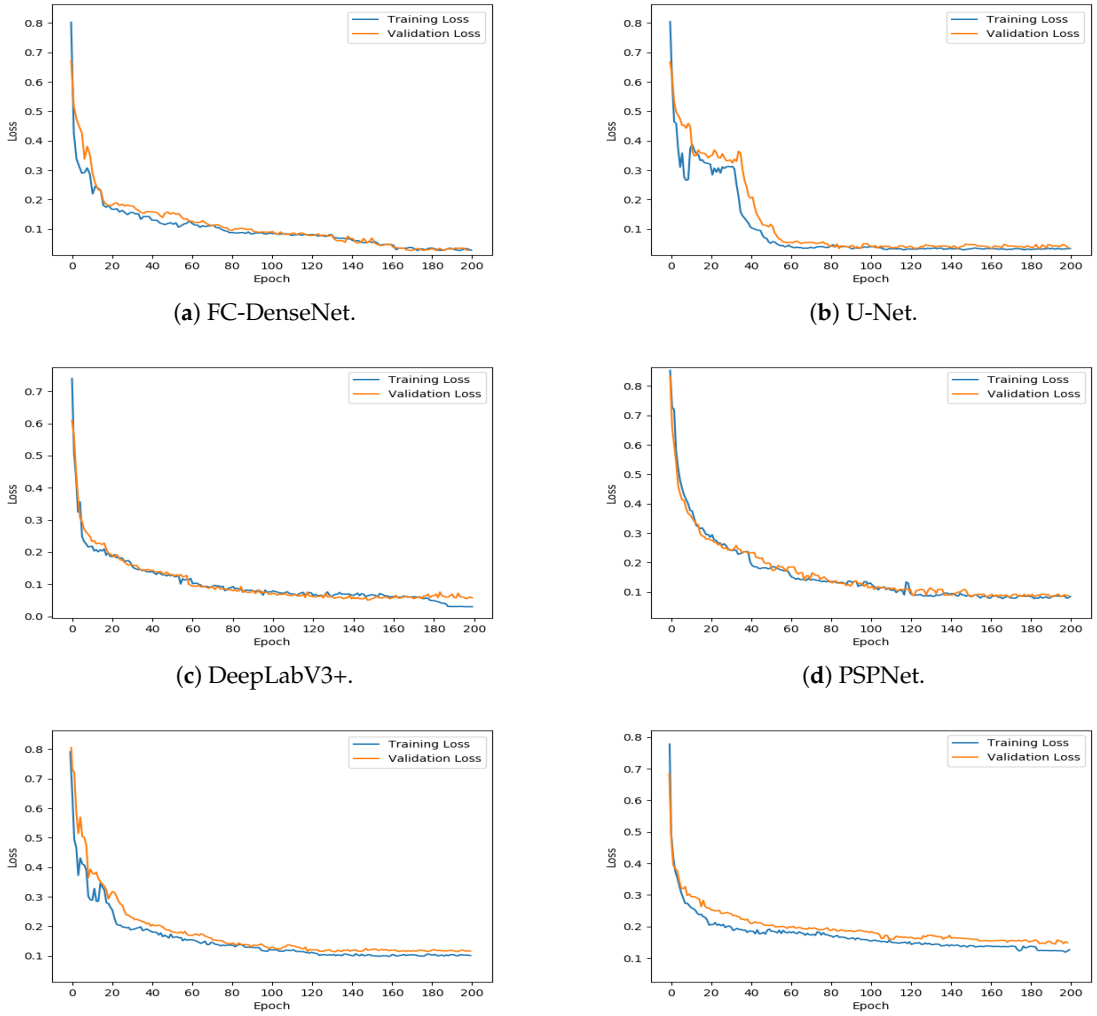
In our experiment, we use a pre-trained ResNet-34 network as a feature encoder in all employed architectures excluding Encoder–Decoder (SegNet) and FC-DenseNet architectures. To predict each image pixel's category, all employed deep architectures include a multi-class softmax classifier on top, which is fed by the output upsampled feature map from the final layer of the network to produce pixel-wise class probabilities. Cross-entropy and Adam optimizer [92] are selected as the loss function and optimization algorithm, respectively. Adam optimizer computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients [92] and realizes the benefits of both AdaGrad [93] and RMSProp [94]. It includes several parameters that need to be carefully set. Popular deep learning libraries generally use the default parameters recommended by the paper including learning rate parameter $\alpha = 0.001$, two exponential decay rate parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and $\epsilon = 1e^{-8}$, which prevents any division by zero in the implementation. In our experiment, we set all optimization parameters according to those recommended values.

Weight initialization is carried out for all employed networks. Except for FC-DenseNet, weight parameters in other networks are initialized by transfer learning. For FC-DenseNet, we decided to train the network from scratch since we did not find a pre-trained FC-DenseNet on large datasets such as ImageNet. FC-DenseNet has very few parameters, about 10 times less than recent state-of-the-art models; thus, it is worth it to train this network from scratch and compare its performance over our limited dataset with the performance of pre-trained encoders. All deep CNN models in this experiment were trained using the same training samples under the same conditions for 200 epochs.

All experiments were carried out on Amazon Web Service (AWS) with one high-performance NVIDIA K80 GPU, with 2496 parallel processing cores and 12 GB of GPU memory and high frequency Intel Xeon E5-2686 v4 processors under CUDA version 10.0.

4. Results

Figure 15 illustrates the training and validation losses for all employed deep CNN architectures trained under the same training dataset. The validation loss curves closely follow corresponding training loss curves showing the ability of the deep CNN models in generalization. Normalized confusion matrices in Figure 16 display the performance of the deep CNN models on each individual land cover target.



(**a**) FC-DenseNet. (**b**) U-Net.

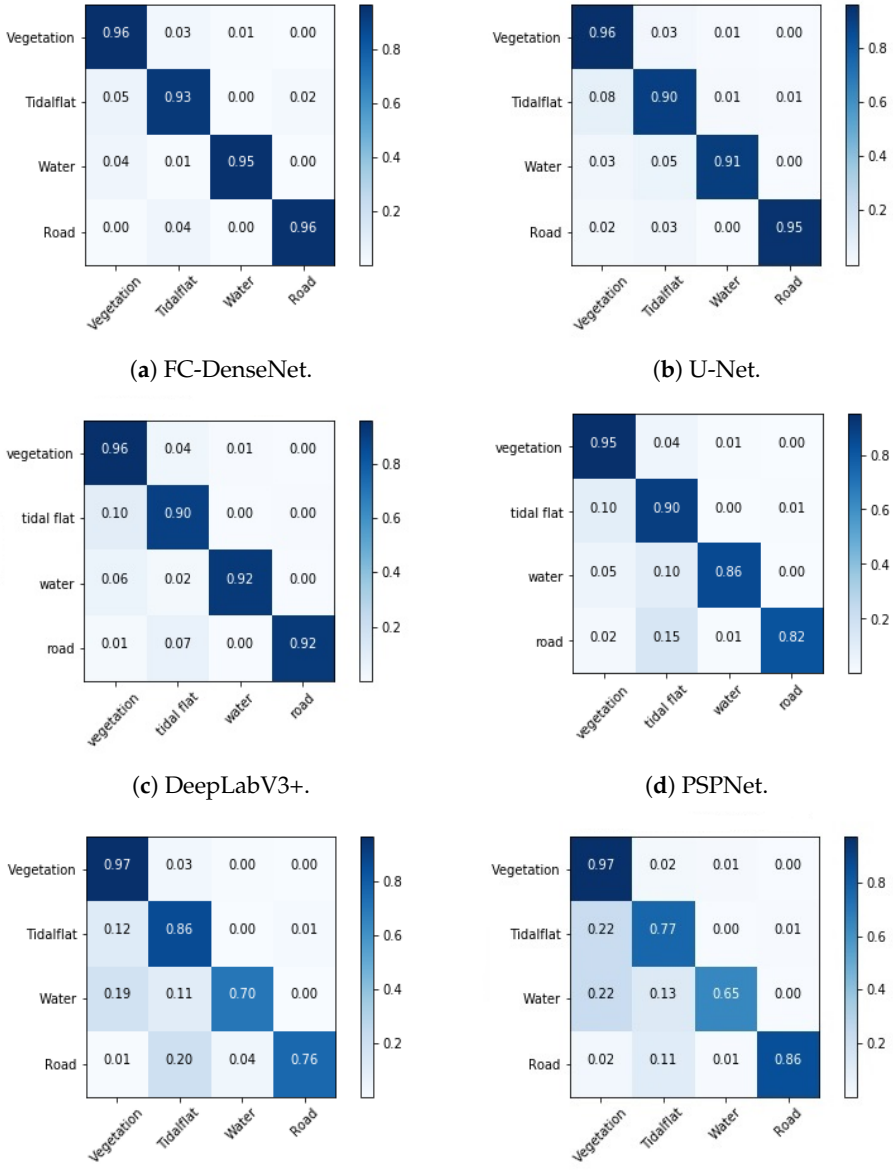
(**c**) DeepLabV3+. (**d**) PSPNet.

(**e**) MobileU-Net. (**f**) SegNet.

Figure 15. Average loss per epoch for training and validation steps.

Table 1 illustrates the land cover prediction results achieved for the different deep CNN architectures employed in the image segmentation experiment. The first two columns represent overall accuracy for training (OA-Train) and validation (OA-Val). Precision, recall, F1, and mIoU are included for evaluating the performance of each architecture as these are some of the most widely used metrics.

Figure 17 displays a cropped orthoimage from the upper portion of the study area, and its corresponding ground truth labels. This area was selected for model validation purposes because it provides a nice distribution of the different land cover classes. Images from this area were not included in the training samples. Figure 17 stems from a set of 64 overlapping UAS images that were orthorectifed and mosaicked together as part of the SfM photogrammetric processing used to create the full study area orthoimage (Figure 8). To classify the orthoimage, the full image is not fed directly into the model due to its large size. Small image patches (512 × 512 pixels) are extracted and then fed into the model to undergo pixel-wise labeling. After the land cover class(es) contained in each individual image patch are predicted by the model, they are then reassembled to generate the full resolution image. The land cover maps predicted for this orthoimage, using all employed deep CNN models in this study, are displayed in Figure 18a–f. Interestingly, land cover classes predicted by all employed CNN models closely resemble the ground truth image in Figure 17. However, FC-DenseNet, UNet, and DeepLabV3+ are the most accurate representations of the ground targets in this complex wetland environment.



(**a**) FC-DenseNet. (**b**) U-Net.

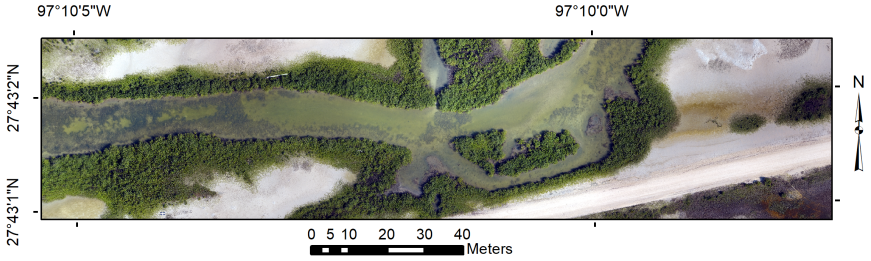
(**c**) DeepLabV3+. (**d**) PSPNet.

(**e**) MobileU-Net. (**f**) SegNet.

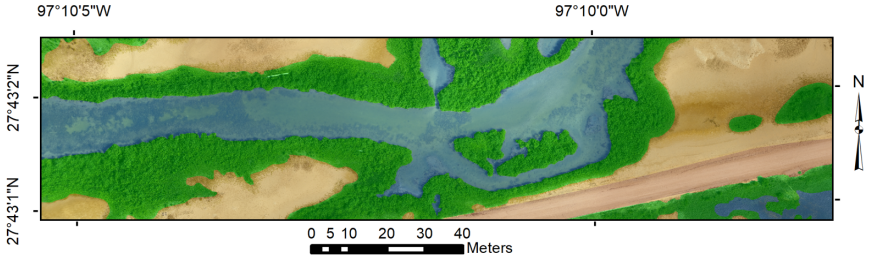
Figure 16. Normalized confusion matrices for the coastal wetland land cover prediction task using different deep CNN architectures.

Table 1. Coastal wetland land cover classification results.

Algorithm	OA-Train	OA-Val	Prec	Recall	F1	mIoU	Veg	TF	Water	Road
FC-DenseNet	0.97	0.95	0.95	0.95	0.95	0.90	0.96	0.93	0.94	0.96
U-Net	0.96	0.95	0.95	0.94	0.94	0.91	0.95	0.93	0.95	0.95
DeepLabV3+	0.94	0.93	0.91	0.90	0.90	0.89	0.94	0.88	0.87	0.89
PSPNet	0.91	0.89	0.89	0.88	0.88	0.83	0.96	0.89	0.87	0.83
MobileU-Net	0.89	0.85	0.88	0.79	0.84	0.75	0.97	0.85	0.69	0.76
SegNet	0.88	0.82	0.91	0.81	0.82	0.69	0.97	0.77	0.65	0.85

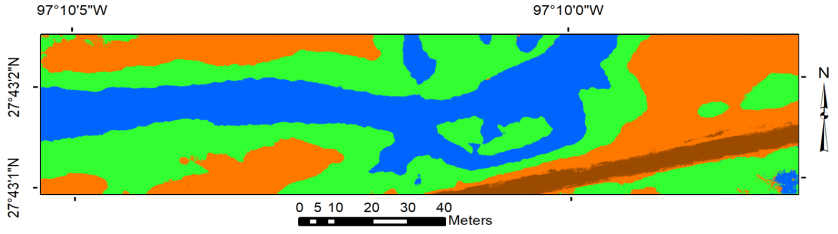


(**a**)

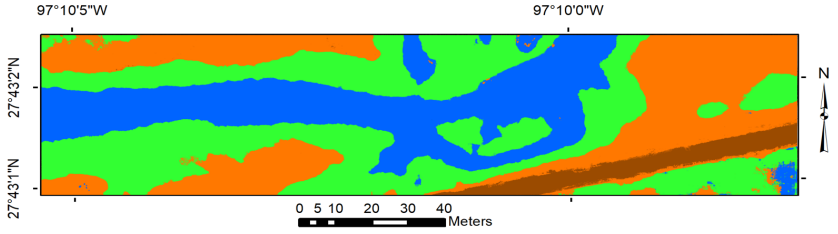


(**b**)

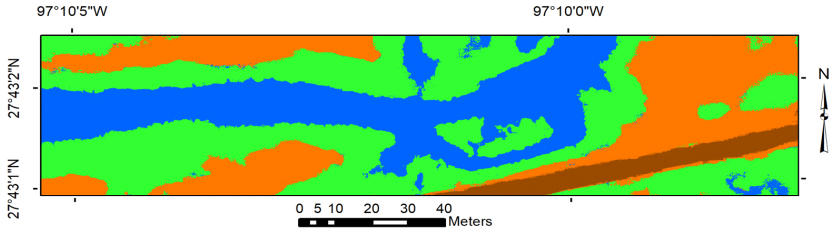
Figure 17. (**a**) original orthoimage generated by mosaicking 64 ortho-rectified UAS images over the wetland study site; (**b**) corresponding ground truth image.



(**a**) FC-DenseNet.

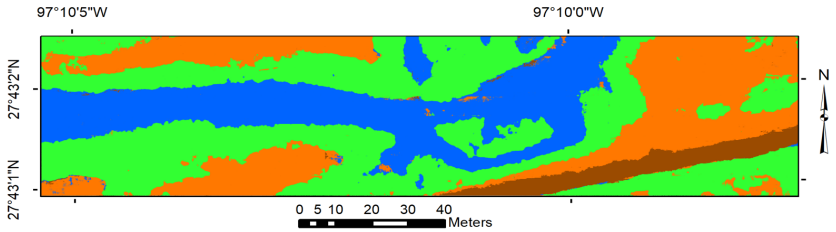


(**b**) U-Net.

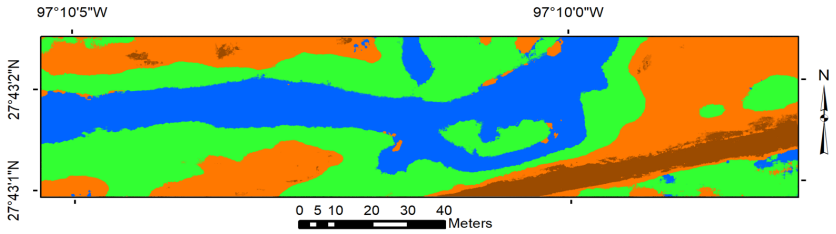


(**c**) DeepLabV3+.

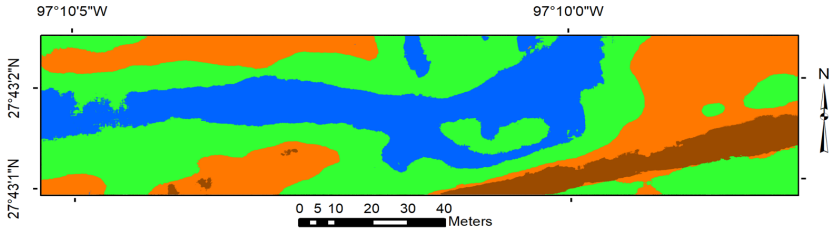
Figure 18. *Cont.*



(**d**) PSPNet.



(**e**) MobileU-Net.



(**f**) SegNet.

Figure 18. Land cover map prediction over prepared orthoimage for part of the coastal wetland test area.

5. Discussion

Referring to Figure 15, FC-DenseNet, U-Net, and DeepLabV3+ show lower loss values for both training and validation losses w.r.t MobileU-Net, PSPNet, and SegNet models, resulting in higher training and validation accuracies according to Table 1. For the SegNet model, the validation losses keep a certain distance above the training losses explaining the larger difference between validation and training accuracies reported for this model. Furthermore, still referring to Figure 15, U-Net is showing a higher speed of convergence during the training phase. This suggests that the skip connections from encoder to decoder have a high contribution in smoothing the gradient descent's path towards the global minimum in the high-dimensional weight space. Additionally, in comparison to FC-DenseNet, the fine-tuning strategy of the transfer learning technique employed by U-Net yielded reduced training epochs. This approach helps to exploit the advantages of deeper CNNs with a larger number of trainable parameters where the available training resources are limited (as is the case here due to manual labeling). FC-DenseNet also takes advantage of skip connections in its encoders, and between encoders and decoders, which helps with the flow and convergence of gradient descent through reuse of features. However, due to training the network from scratch, more training steps to converge is necessary.

The fine-tuning strategy of transfer learning yielded very good results in all models with pre-trained VGG-16 and ResNet-34 architectures as their encoder for feature learning. This is a promising result given that the structure of low-level and high-level natural/wetland terrain features in our dataset are noticeably different from those that appear in the ImageNet dataset used for training the deep CNN architectures. Furthermore, the overall accuracy achieved by training FC-DenseNet from scratch confirms that the dramatic reduction in the number of parameters of this architecture w.r.t. other state-of-the-art deep learning architectures enables it to learn optimum features when presented with relatively limited training samples.

Regarding the F1 score and mIoU values depicted in Table 1, the first three CNN models exhibit the highest performance among the others. According to the confusion matrices displayed in Figure 16, three of the employed networks, FC-DenseNet, U-Net, and DeepLabV3+, were successful in predicting labels for pixels belonging to all existing classes with accuracy above 90%. Almost all deep networks were successful in predicting pixels belonging to the vegetation class with an accuracy greater than 95%. Compared to the other classes of targets, vegetation represents the least confused class. Referring back to Figure 16, especially when SegNet, PSPNet, and MobileU-Net models were employed, road pixels are mostly confused with tidal flat pixels, and pixels belonging to water bodies are more likely to be misclassified as tidal flat and vegetation. It should be noted that discriminating pixels belonging to the tidal flat class from those belonging to the road class at this study site is a difficult task. These two classes exhibit very high inter-class similarities due to the road being a dirt road comprised of similar sand material to that of the exposed ground areas within tidal flat areas but with some mixed gravel.

The comparable overall accuracy of the FC-DenseNet architecture trained from scratch to that of the U-Net and DeepLab V3+ architectures, which use fine-tuned encoders, illustrates that the compactness in the number of parameters of FC-DenseNet makes it a good choice among many recently-developed CNN architectures for pixel-wise labeling for training from scratch under limited training samples. The high performance of the U-Net architecture, trained based on the transfer learning technique, provided the most accurate and efficient choice among the others for pixel-wise labeling. Its performance justifies that the employed transfer learning technique does very well when it is employed to learn hierarchical features in high-spatial resolution UAS or RS images over natural terrain like wetlands. Such image sets and features are significantly different from the features of standard image datasets, such as ImageNet [54]. High performance of the DeepLabV3+ architecture demonstrates the effectiveness of ASPP in this network, which is able to properly encode multi-scale contextual information of the coastal wetland land cover captured in the images. However, this network needs more training steps to reach comparable performance w.r.t U-Net. Our experiment with PSPNet at the wetland study site shows that the pyramid pooling module together with the pyramid scene parsing network is more effective in predicting vegetation and tidal flat areas than water and road areas. MobileU-Net and SegNet achieved less accuracy among all employed architectures for semantic image segmentation. Results achieved by our MobileNet architecture is based on baseline settings for its hyper-parameters, which include $\alpha = 1$ for width multiplier and $\rho = 1$ for resolution multiplier. Decreasing those two hyper-parameters can dramatically decrease the performance of the network. However, MobileNets have the potential to be employed effectively in some real-time RS applications. As mentioned earlier, MobileNets were built as small, low-latency, low-power models parameterized to meet the resource constraints of a variety of mobile and embedded vision applications. These type models require less computational power and capacity for near real-time applications compared to very deep architectures with a higher capacity for learning due to their larger number of parameters. SegNet, like other employed architectures in this experiment, performed very well in vegetation areas but was much less accurate in classifying other targets. It is suspected that SegNet's inefficiency for pixel-wise labeling of the other targets, which are more challenging, stems from the network's inefficiency for exploiting low-level and high-level abstract features throughout the network and in its inefficient upsampling method.

It is worth mentioning that the information needed for training any of the evaluated classification architectures was obtained through supervised labeling by interpretation and delineation of land cover boundaries in the UAS images. This interpretation includes labeling a relatively large number of images by a human operator. This may result in different types of errors in the labelling of land cover types, and most notably in those circumstances in which categories are very heterogeneous and the landscape is complex. This is especially worrisome for non-domain experts or practitioners of deep learning who may not be familiar with the key characteristics that differentiate one land cover type or boundary from another. In this case, training was limited to four relatively distinct classes of importance to our wetland monitoring efforts, as opposed to more refined classes, to try and

reduce those issues. Although different types of vegetation and land cover exist in the study area, this grouping aided our ability to efficiently label the data and serve the study purpose. However, the high level of classification accuracy reported here, to some degree, may be a function of this class structure. Efforts to classify the land cover into more distinct categories and capture more biodiversity will be posed with greater labeling and training challenges and require more domain expertise. Classification accuracy may be lower in such cases than those reported here, especially if relying on low spectral resolution RGB imagery alone as evaluated in this study. Inevitably, some mixing of classes will occur during the labeling process, regardless of expertise or attention to detail, and these challenges will grow over heterogeneous and complex natural landscapes like coastal wetlands. This problem can be exacerbated when attempting to perform pixel-level labeling using very high resolution imagery, such as created from a low-altitude UAS flight. This is due to a large amount of within class spectral variability when viewing land cover at zoomed in geographic scales (here cm-level). The errors in labeling are specifically maximized when pixels belonging to the borderlines are going to be labeled because natural targets do not usually express clear borders. In some landscapes, two or more different targets can be so mixed together that the operator cannot decide which label should be given to that specific pixel or area. Inevitably, it becomes highly subjective. Such areas can be seen in the lower right part of Figure 17 where a vegetation area has been submerged in shallow water. In this work, it was classified as a water body/submerged landscape. Additionally, at this specific study site, discriminating pixels belonging to edges of the tidal flat class from those belonging to the dirt road was a difficult task because those two classes exhibit very high inter-class similarities at their boundaries. As a result, the uncertainty for labeling road pixels close to the boundaries increased.

Lastly, coastal wetlands are among some of the most dynamic and complex ecosystems on the planet. Many different factors, such as seasonal and climate changes, water temperature, altered flooding and salinity patterns, sea-level rise, topography, etc., [11,13], contribute to the current state of the land cover and its physical properties at the time of recording the remote sensing observations. Thus, the authors emphasize that the classification results shown here, based on the classes chosen to be examined, are valid for the specific data set acquired at a certain time over the study site. The results cannot be necessarily generalized to the same coastal wetland area imaged at a different time, or at a different land cover state, without further analyses. Ambient environmental conditions, such as lighting or wind, can impact data captured in an UAS image. Similarly, flight design including altitude above ground and camera perspective (e.g., oblique versus nadir) will impact the GSD and appearance of land cover features. As a result, the visual representation of the same target may deviate from one exposure to another in a single UAS flight mission and across repeat data acquisitions. For this study, UAS data acquisition targeted calm winds and a bright, sunny day. The flight was conducted during the middle part of the day to reduce shadowing and enhance scene brightness. Furthermore, the entire scene was mapped in under thirty minutes so variation of ambient lighting during flight was minimal. Camera angles were kept at nadir to provide a top-down view for orthoimage generation and reduce shadowing of terrain from oblique perspectives.

Future efforts will need to examine the generalizability and stability of these models to perform repetitive classification using a time series of images captured from repeat UAS flights under varying conditions. However, we believe that the high capacity of deep CNN models to efficiently extract informative and discriminative features from the raw UAS images in an end-to-end manner have the potential to be extended further by training deep CNN models using a time-series of UAS images acquired over the same area. An efficient deep network trained using appropriate training samples acquired at different times and labeled by expert knowledge will be able to capture more properties about a certain land cover target at a different state of the wetland or other environment. Such models could provide a powerful framework for designing any automatic or online land cover prediction system aiming to offer high performance regardless of the conditions at the time of data acquisition.

6. Conclusions

Wetlands provide a challenging natural environment for performing high accuracy land cover prediction with hyper-spatial resolution UAS imagery due to high intra-class variability and low inter-class disparity often observed between classes. For decades, semantic image segmentation for land cover mapping tasks in the RS field has relied heavily on the tedious procedure of manually designing and extracting the most informative hand-crafted features from the available data, which are then fed into different machine learning techniques for classification or segmentation. On the other hand, the accuracy of any prediction technique is highly dependent on the contribution of those features for discriminating different targets that are captured in high-spatial to hyper-spatial resolution images, such as those acquired by UAS flying at low altitude.

In this research, we exploited state-of-the-art deep learning frameworks, commonly called deep CNNs, to automatically explore high-dimensional hierarchical feature spaces and find the most informative and discriminative features for performing a pixel-wise image labeling task for land cover mapping. Among the many available deep CNN architectures, this study investigated the performance of some of the more recent very deep CNN architectures that are heavily employed for pixel-level labeling in many different applications. Six different networks were evaluated, FC-DenseNet, U-Net, DeepLabV3+, MobileU-Net, PSPNet, and Encoder–Decoder (SegNet), for performing a pixel-wise classification task using UAS hyper-spatial resolution images acquired over a coastal wetland area. Results of the study revealed that hierarchical features learned by the deep learning frameworks are highly efficient for discriminating different targets in a complex wetland environment and providing accurate pixel-level land cover predictions for the target classes investigated (vegetation, tidal flat, water, road). Specifically, fine-tuning of deep architectures with tens of millions of parameters is the best strategy when there is a limited labeled dataset as was the case in this study. This is also the case for most current RS land cover mapping applications where large repositories of relevant labeled datasets are not available. In this study, FC-DenseNet trained from scratch outperformed the other architectures regarding the overall accuracy performances (Table 1) based on the validation dataset. However, U-Net architecture with ResNet34 encoder outperformed the other architectures based on training speed while achieving comparable accuracy to FC-DenseNet. These results suggest that U-Net is the most efficient architecture for the UAS hyper-spatial pixel-wise classification task explored here. Skip connections in FC-DenseNet and U-Net architecture play a significant role in these networks' ability for faster training and/or achieving higher overall accuracies. DeepLabV3+, which uses the ASPP technique to account for objects at multiple scales, was also very successful at pixel-level prediction in our study case. Furthermore, results from per-class accuracy revealed that almost all networks were able to successfully predict pixels belonging to the vegetation area with high accuracy.

The experiment with the U-Net architecture employing a ResNet34 encoder revealed that fine-tuning using the transfer learning technique works well for hyper-spatial UAS image analyses. Furthermore, the transfer learning technique in combination with skip connections applied to the architecture of CNNs significantly reduced the need for a large number of training epochs, and large labeled data resources, typically required for training deep CNNs without sacrificing their high classification performance. In this study, FC-DenseNet, with 56 convolutional layers, trained from scratch performed comparably well with the U-Net architecture regarding the overall classification accuracy evaluated on the training dataset. This suggests that the parameter-based compactness of FC-DenseNet makes it a good choice among other deep CNN architectures for accurate pixel-wise labeling in RS applications where transfer learning may not be efficiently applicable and/or higher level of generalization with a limited training sample is required. However, as long as training from scratch is applied to FC-DenseNet, it would need more training epochs to reach an overall accuracy comparable to U-Net using a pre-trained encoder with the same number of training samples.

In conclusion, the results of this study demonstrate the high potential for exploiting recent deep CNN architectures to perform pixel-wise land cover mapping with hyper-spatial resolution imagery acquired from a small UAS equipped with an RGB camera or other RS method. Transfer

learning is highly applicable for training deep CNNs in RS applications to help achieve state-of-the-art performances when faced with limited labeled data resources. Finally, coastal wetlands are highly diverse natural environments providing a range of complexities if attempting to identify more refined land covers, such as vegetation types. Such efforts will likely demand more advanced sensors to capture finer spectral information from the different targets. Future work will explore deep CNN architectures for pixel-wise labeling of multispectral and hyperspectral images to predict land cover in a coastal wetland setting.

Author Contributions: M.P. and M.J.S. conceived the overall study concept and approach; M.P. formulated experimental design; H.K. and P.T. assisted in experimental design; M.P. developed computational code, performed the experiments, and analyzed the results; M.P. and H.K. analyzed data and prepared training and validation datasets; H.K., M.J.S., and P.T. helped with results interpretation; M.P., H.K., and M.J.S. wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This publication was prepared by Texas A&M University-Corpus Christi using Federal funds under award NA18NOS4000198 from the National Oceanic and Atmospheric Administration, U.S. Department of Commerce. The statements, findings, conclusions, and recommendations are those of the author(s) and do not necessarily reflect the views of the National Oceanic and Atmospheric Administration or the U.S. Department of Commerce.

Acknowledgments: The authors gratefully acknowledge James Rizzo and Jacob Berryhill of the Conrad Blucher Institute for Surveying and Science for providing UAS field work support.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Boon, M.; Greenfield, R.; Tesfamichael, S. Wetland assessment using unmanned aerial vehicle (UAV) photogrammetry. In Proceedings of the XXIII ISPRS Congress, Prague, Czech Republic, 12–19 July 2016.
2. Laliberte, A.S.; Rango, A.; Herrick, J. Unmanned aerial vehicles for rangeland mapping and monitoring: A comparison of two systems. In Proceedings of the ASPRS 2007 Annual Conference, Tampa, FL, USA, 7–11 May 2007.
3. Pashaei, M.; Starek, M.J. Fully Convolutional Neural Network for Land Cover Mapping In A Coastal Wetland with Hyperspatial UAS Imagery. In Proceedings of the IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 6106–6109.
4. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
5. Hariharan, B.; Arbeláez, P.; Girshick, R.; Malik, J. Simultaneous detection and segmentation. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 297–312.
6. Stedman, S.M.; Dahl, T.E. Status and Trends of Wetlands in the Coastal Watersheds of The Eastern United States, 1998 to 2004. 2008. Available online: https://www.fws.gov/wetlands/Documents/Status-and-Trends-of-Wetlands-in-the-Coastal-Watersheds-of-the-Eastern-United-States-1998-to-2004.pdf (accessed on 13 March 2020).
7. Pendleton, L.H. *The Economic and Market Value of Coasts and Estuaries What's at Stake*; Restore America's Estuaries: Arlington, VA, USA, 2011.
8. Olmsted, I.C.; Armentano, T.V. *Vegetation of Shark Slough, Everglades National Park*; South Florida Natural Resources Center, Everglades National Park Homestead: Homestead, FL, USA, 1997.
9. Belluco, E.; Camuffo, M.; Ferrari, S.; Modenese, L.; Silvestri, S.; Marani, A.; Marani, M. Mapping salt-marsh vegetation by multispectral and hyperspectral remote sensing. *Remote Sens. Environ.* **2006**, *105*, 54–67. [CrossRef]
10. Smith, G.M.; Spencer, T.; Murray, A.L.; French, J.R. Assessing seasonal vegetation change in coastal wetlands with airborne remote sensing: An outline methodology. *Mangroves Salt Marshes* **1998**, *2*, 15–28. [CrossRef]
11. Cahoon, D.R.; Guntenspergen, G.R. Climate change, sea-level rise, and coastal wetlands. *Natl. Wetl. Newsl.* **2010**, *32*, 8–12.
12. Silvestri, S.; Marani, M.; Marani, A. Hyperspectral remote sensing of salt marsh vegetation, morphology and soil topography. *Phys. Chem. Earth Parts A/B/C* **2003**, *28*, 15–25. [CrossRef]

13. Taramelli, A.; Valentini, E.; Cornacchia, L.; Monbaliu, J.; Sabbe, K. Indications of dynamic effects on scaling relationships between channel sinuosity and vegetation patch size across a salt marsh platform. *J. Geophys. Res. Earth Surf.* **2018**, *123*, 2714–2731. [CrossRef]
14. Myint, S.W.; Gober, P.; Brazel, A.; Grossman-Clarke, S.; Weng, Q. Per-pixel vs. object-based classification of urban land cover extraction using high spatial resolution imagery. *Remote Sens. Environ.* **2011**, *115*, 1145–1161. [CrossRef]
15. Hsieh, P.F.; Lee, L.C.; Chen, N.Y. Effect of spatial resolution on classification errors of pure and mixed pixels in remote sensing. *IEEE Trans. Geosci. Remote Sens.* **2001**, *39*, 2657–2663. [CrossRef]
16. Tso, B.C.; Mather, P.M. Classification of multisource remote sensing imagery using a genetic algorithm and Markov random fields. *IEEE Trans. Geosci. Remote Sens.* **1999**, *37*, 1255–1260. [CrossRef]
17. Blaschke, T. Object based image analysis for remote sensing. *ISPRS J. Photogramm. Remote Sens.* **2010**, *65*, 2–16. [CrossRef]
18. Dronova, I.; Gong, P.; Wang, L. Object-based analysis and change detection of major wetland cover types and their classification uncertainty during the low water period at Poyang Lake, China. *Remote Sens. Environ.* **2011**, *115*, 3220–3236. [CrossRef]
19. Small, C.; Milesi, C. Multi-scale standardized spectral mixture models. *Remote Sens. Environ.* **2013**, *136*, 442–454. [CrossRef]
20. Pande-Chhetri, R.; Abd-Elrahman, A.; Liu, T.; Morton, J.; Wilhelm, V.L. Object-based classification of wetland vegetation using very high-resolution unmanned air system imagery. *Eur. J. Remote Sens.* **2017**, *50*, 564–576. [CrossRef]
21. Li, M.; Zang, S.; Zhang, B.; Li, S.; Wu, C. A review of remote sensing image classification techniques: The role of spatio-contextual information. *Eur. J. Remote Sens.* **2014**, *47*, 389–411. [CrossRef]
22. Whiteside, T.G.; Boggs, G.S.; Maier, S.W. Comparing object-based and pixel-based classifications for mapping savannas. *Int. J. Appl. Earth Obs. Geoinf.* **2011**, *13*, 884–893. [CrossRef]
23. Gao, Y.; Mas, J.F. A comparison of the performance of pixel-based and object-based classifications over images with various spatial resolutions. *Online J. Earth Sci.* **2008**, *2*, 27–35.
24. Rollet, R.; Benie, G.; Li, W.; Wang, S.; Boucher, J. Image classification algorithm based on the RBF neural network and K-means. *Int. J. Remote Sens.* **1998**, *19*, 3003–3009. [CrossRef]
25. Blanzieri, E.; Melgani, F. Nearest neighbor classification of remote sensing images with the maximal margin principle. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 1804–1811. [CrossRef]
26. Goncalves, M.; Netto, M.; Costa, J.; Zullo Junior, J. An unsupervised method of classifying remotely sensed images using Kohonen self-organizing maps and agglomerative hierarchical clustering methods. *Int. J. Remote Sens.* **2008**, *29*, 3171–3207. [CrossRef]
27. Civco, D.L. Artificial neural networks for land-cover classification and mapping. *Int. J. Geogr. Inf. Sci.* **1993**, *7*, 173–186. [CrossRef]
28. Vapnik, V.; Vapnik, V. Statistical Learning Theory; Springer: Berlin/Heidelberg, Germany, 1998.
29. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
30. Chen, Y.; Lin, Z.; Zhao, X.; Wang, G.; Gu, Y. Deep learning-based classification of hyperspectral data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2094–2107. [CrossRef]
31. Zou, Q.; Ni, L.; Zhang, T.; Wang, Q. Deep learning based feature selection for remote sensing scene classification. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 2321–2325. [CrossRef]
32. Chen, Y.; Zhao, X.; Jia, X. Spectral–spatial classification of hyperspectral data based on deep belief network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 2381–2392. [CrossRef]
33. Cheng, G.; Zhou, P.; Han, J. Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 7405–7415. [CrossRef]
34. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [CrossRef] [PubMed]
35. Lin, G.; Shen, C.; Van Den Hengel, A.; Reid, I. Efficient piecewise training of deep structured models for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 1 July 2016; pp. 3194–3203.

36. Dai, J.; He, K.; Sun, J. Instance-aware semantic segmentation via multi-task network cascades. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 1 July 2016; pp. 3150–3158.
37. Garcia-Garcia, A.; Orts-Escolano, S.; Oprea, S.; Villena-Martinez, V.; Garcia-Rodriguez, J. A review on deep learning techniques applied to semantic segmentation. *arXiv* **2017**, arXiv:1704.06857.
38. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 3320–3328.
39. Maggiori, E.; Tarabalka, Y.; Charpiat, G.; Alliez, P. Convolutional neural networks for large-scale remote-sensing image classification. *IEEE Trans. Geosci. Remote Sens.* **2016**, *55*, 645–657. [CrossRef]
40. Romero, A.; Gatta, C.; Camps-Valls, G. Unsupervised deep feature extraction for remote sensing image classification. *IEEE Trans. Geosci. Remote Sens.* **2015**, *54*, 1349–1362. [CrossRef]
41. Paoletti, M.; Haut, J.; Plaza, J.; Plaza, A. A new deep convolutional neural network for fast hyperspectral image classification. *ISPRS J. Photogramm. Remote Sens.* **2018**, *145*, 120–147. [CrossRef]
42. Palsson, F.; Sveinsson, J.R.; Ulfarsson, M.O. Multispectral and hyperspectral image fusion using a 3-D-convolutional neural network. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 639–643. [CrossRef]
43. Liu, T.; Abd-Elrahman, A.; Morton, J.; Wilhelm, V.L. Comparing fully convolutional networks, random forest, support vector machine, and patch-based deep convolutional neural networks for object-based wetland mapping using images from small unmanned aircraft system. *GIScience Remote Sens.* **2018**, *55*, 243–264. [CrossRef]
44. Liu, T.; Abd-Elrahman, A. Deep convolutional neural network training enrichment using multi-view object-based analysis of Unmanned Aerial systems imagery for wetlands classification. *ISPRS J. Photogramm. Remote Sens.* **2018**, *139*, 154–170. [CrossRef]
45. Pouliot, D.; Latifovic, R.; Pasher, J.; Duffe, J. Assessment of convolution neural networks for wetland mapping with landsat in the central Canadian boreal forest region. *Remote Sens.* **2019**, *11*, 772. [CrossRef]
46. Hu, Y.; Zhang, J.; Ma, Y.; An, J.; Ren, G.; Li, X. Hyperspectral coastal wetland classification based on a multiobject convolutional neural network model and decision fusion. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1110–1114. [CrossRef]
47. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
48. Romera-Paredes, B.; Torr, P.H.S. Recurrent instance segmentation. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 312–329.
49. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 818–833.
50. Zeiler, M.D.; Taylor, G.W.; Fergus, R.; others. Adaptive deconvolutional networks for mid and high level feature learning. *ICCV* **2011**, *1*, 6.
51. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
52. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.
53. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; NIPS, Inc.: Montreal, QC, Canada, 2012; pp. 1097–1105.
54. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
55. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
56. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
57. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

58. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
59. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.
60. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
61. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
62. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
63. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Lauderdale, FL, USA, 11–13 April 2011; pp. 315–323.
64. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 630–645.
65. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.
66. Huang, G.; Sun, Y.; Liu, Z.; Sedra, D.; Weinberger, K.Q. Deep networks with stochastic depth. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 646–661.
67. Veit, A.; Wilber, M.J.; Belongie, S. Residual networks behave like ensembles of relatively shallow networks. In *Advances in Neural Information Processing Systems*; NIPS, Inc.: Montreal, QC, Canada, 2016; pp. 550–558.
68. Wu, Z.; Shen, C.; Van Den Hengel, A. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognit.* **2019**, *90*, 119–133. [CrossRef]
69. Sifre, L.; Mallat, S. Rigid-Motion Scattering for Image Classification. Ph.D. Thesis, CMAP Ecole Polytechnique, Palaiseau, France, 2014.
70. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
71. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
72. Zeiler, M.D.; Krishnan, D.; Taylor, G.W.; Fergus, R. Deconvolutional networks. *CVPR* **2010**, *10*, 7.
73. Noh, H.; Hong, S.; Han, B. Learning deconvolution network for semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1520–1528.
74. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
75. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing And Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
76. Burt, P.; Adelson, E. The Laplacian pyramid as a compact image code. *IEEE Trans. Commun.* **1983**, *31*, 532–540. [CrossRef]
77. Lin, G.; Milan, A.; Shen, C.; Reid, I. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1925–1934.
78. Jégou, S.; Drozdzal, M.; Vazquez, D.; Romero, A.; Bengio, Y. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 11–19.
79. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv* **2014**, arXiv:1412.7062.
80. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. *arXiv* **2015**, arXiv:1511.07122.

81. Holschneider, M.; Kronland-Martinet, R.; Morlet, J.; Tchamitchian, P. A real-time algorithm for signal analysis with the help of the wavelet transform. In *Wavelets*; Springer: Berlin/Heidelberg, Germany, 1990; pp. 286–297.
82. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.
83. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.
84. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
85. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2009**, *22*, 1345–1359. [CrossRef]
86. Goutte, C.; Gaussier, E. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In Proceedings of the European Conference on Information Retrieval, Santiago de Compostela, Spain, 21–23 March 2005; pp. 345–359.
87. Rahman, M.A.; Wang, Y. Optimizing intersection-over-union in deep neural networks for image segmentation. In Proceedings of the International Symposium on Visual Computing, Las Vegas, NV, USA, 12–14 December 2016; pp. 234–244.
88. Paine, J.G.; White, W.A.; Smyth, R.C.; Andrews, J.R.; Gibeaut, J.C. Mapping coastal environments with lidar and EM on Mustang Island, Texas, US. *Lead. Edge* **2004**, *23*, 894–898. [CrossRef]
89. Nguyen, C.; Starek, M.; Tissot, P.; Gibeaut, J. Unsupervised clustering method for complexity reduction of terrestrial lidar data in marshes. *Remote Sens.* **2018**, *10*, 133. [CrossRef]
90. Nguyen, C.; Starek, M.J.; Tissot, P.; Gibeaut, J. Unsupervised Clustering of Multi-Perspective 3D Point Cloud Data in Marshes: A Case Study. *Remote Sens.* **2019**, *11*, 2715. [CrossRef]
91. Westoby, M.J.; Brasington, J.; Glasser, N.F.; Hambrey, M.J.; Reynolds, J.M. 'Structure-from-Motion' photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology* **2012**, *179*, 300–314. [CrossRef]
92. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
93. Duchi, J.; Hazan, E.; Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **2011**, *12*, 2121–2159.
94. Tieleman, T.; Hinton, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA Neural Netw. Mach. Learn.* **2012**, *4*, 26–31.