Journal of *Marine Science and Engineering*

MDPI

Article

Coral Image Segmentation with Point-Supervision via Latent Dirichlet Allocation with Spatial Coherence

Xi Yu [1,*], Bing Ouyang [2] and Jose C. Principe [1,*]

[1] Computational NeuroEngineering Laboratory (CNEL), Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA
[2] Harbor Branch Oceanographic Institute HBOI 5600 U.S. 1 North, HB18 130, Fort Pierce, FL 34946, USA; bouyang@fau.edu
* Correspondence: yuxi@ufl.edu (X.Y.); principe@cnel.ufl.edu (J.C.P.)

Abstract: Deep neural networks provide remarkable performances on supervised learning tasks with extensive collections of labeled data. However, creating such large well-annotated data sets requires a considerable amount of resources, time and effort, especially for underwater images data sets such as corals and marine animals. Therefore, the overreliance on labels is one of the main obstacles for widespread applications of deep learning methods. In order to overcome this need for large annotated dataset, this paper proposes a label-efficient deep learning framework for image segmentation using only very sparse point-supervision. Our approach employs a latent Dirichlet allocation (LDA) with spatial coherence on feature space to iteratively generate pseudo labels. The method requires, as an initial condition, a Wide Residual Network (WRN) trained with sparse labels and mutual information constraints. The proposed method is evaluated on the sparsely labeled coral image data set collected from the Pulley Ridge region in the Gulf of Mexico. Experiments show that our method can improve image segmentation performance against sparsely labeled samples and achieves better results compared with other semi-supervised approaches.

Keywords: coral image segmentation; point-supervision; label-efficient; latent dirichlet allocation (LDA); mutual information; iterative training

Citation: Yu, X.; Ouyang, B.; Principe, J.C. Coral Image Segmentation with Point-Supervision via Latent Dirichlet Allocation with Spatial Coherence. *J. Mar. Sci. Eng.* **2021**, *9*, 157. https://doi.org/10.3390/jmse9020157

Academic Editor: Michael O'Byrne
Received: 9 January 2021
Accepted: 30 January 2021
Published: 5 February 2021

1. Introduction

Semantic image segmentation is the process of assigning a categorical label to each image pixel automatically. There are many critical applications that require this procedure, such as marine species detection and conservation, object localization, and scene understanding. For instance, coral detection in reef imagery is one such applications because coral reefs are struggling due to global warming and pollution. However, the quantification of coral abundance is currently completed by humans and it is a time-consuming, boring, and expensive task. For example, it takes 16 people to work for several months to analyze the abundance of corals in hundreds of images collected from one typical two-week cruise. Hence, semantic segmentation can be used to quantify the abundance of each species by counting the number of pixels belonging to that category. In recent years, this topic has been widely investigated using deep learning based methods such as SegNet [1], Unet [2] and fully convolutional network (FCN) [3]. However, such models require full pixel-level annotation to train. Unfortunately, existing marine species and biomedical images data sets lack annotated labels due to the cost of pixel-level labels. In our work, humans will provide labels only for 50 pixels per image. Figure 1 shows the sparse point-level labels in the coral images data set, where different colors represent different classes.

Semi-supervised semantic segmentation can be framed as semi-supervised image classification with sliding window patch to identify the class of the patch's central pixel. Prior works on semi-supervised classification are divided into two main categories. The first is consistency regularization which adds a regularizer into the loss function. This term

is applied to either all images or only the unlabeled samples, and designed based on the assumption that if a realistic perturbation was applied to the unlabeled data samples, the network prediction should not change significantly. Π-model [4] encourages that the distance between a network output with original input and its corresponding standard transformation (i.e., flipping, cropping) should be small. Virtual adversarial training (VAT) [5] approximates a tiny perturbation to the corresponding input data that would most significantly affect network output, then they put consistency regularization into the objective function to penalize the difference in the network outputs for the perturbed and unperturbed samples. Methods in the second category are called pseudo labeling because they assign pseudo-labels to the unlabeled samples based on either a network trained by predictor or the similarity between labeled and unlabeled samples. The pseudo-labeled examples augment the human labels in the training process with supervised loss, such as cross entropy. Both categories use a standard loss term that is trained with supervision from labeled samples. Our method belongs to the pseudo-labeling methodology.

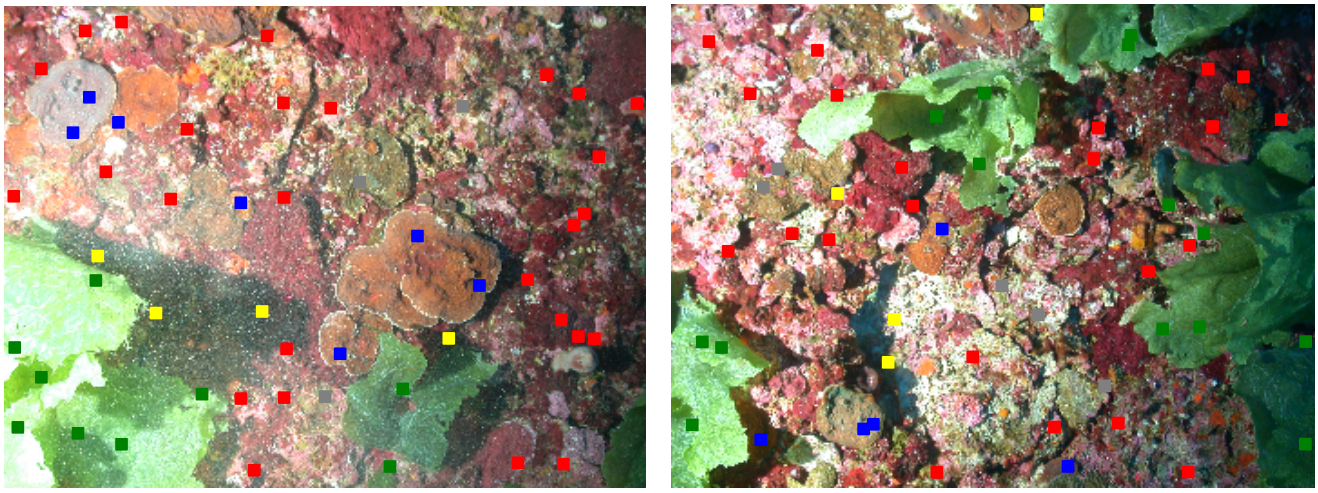

Figure 1. Point-supervision : there are only 50 points pixel labels in each image, blue color represents coral, green color represents green algae, red color represents red algae, gray color represents rock and yellow color represents other species.

There are many different ways to assign pseudo-labels on unlabeled data. The simplest way to generate pseudo-labels is based on the distance from the true labels, as exemplified by our previous work [6,7] to generate pseudo-label using superpixels in the input images. Lee et al. [8] was the first, to our knowledge, to use the trained network to infer pseudo-labels of unlabeled examples effectively by choosing the most confident class. Similarly, entropy minimization (EntMin) [9] encourages the network to make "confident" predictions for all unlabeled samples. The same principle was adopted by Shi et al. [10], where the authors further add contrastive loss to the consistency loss in the feature space, combined with a Mean Teacher approach [11]. Blundell et al. [12] and Kendall et al. [13] infer the pseudo-labels using Bayesian neural network (BNN) rather than the traditional neural network. Other methods for generating pseudo-labels employ a graph model, which consider samples as nodes and find the labels of unlabeled nodes from labeled nodes. Zhu et al. [14] proposes label prorogation and Ahmet et al. [15] applies label propagation into a deep neural network. Carlini et al. [16] achieves the better performance by incorporating ideas of consistency regularization, entropy minimization and Mixup operation [17]. Recently, deep learning has been applied to coral images. Gonzalez-Rivero et al. [18] employ convolutions neural networks in coral image patch classification, but without data augmentation. Akbari Asanjan et al. [19] develop a deep learning model for extracting domain invariant features from multimodal remote sensing imagery to create high-resolution coral images. Modasshir et al. [20] focus on coral images video and uses forward and backward tracking

algorithms to generate labels. Our method is different from all above methods, and our pseudo-labels are inferred from a latent class distribution.

Our method focus on the feature space because the input space is high dimensional which is hard to do clustering. It is obvious that a good feature representation plays a critical role in our proposed method. To this end, we apply information maximization criterion, which maximizes the mutual information between the input and latent features, in the training process to obtain a good representation. In this paper, we use matrix-based Rényi's α-order entropy functional proposed by Giraldo et al. [21] to estimate the mutual information and Yu et al. [22] extend it to multivariate condition. The main advantage of this approach is that it estimates the entropy and joint entropy directly from data without PDF estimation. This methodology is different from variational information bottleneck (VIB) [23] and mutual information neural estimation (MINE) [24], which either approximate the variational lower bound of mutual information or find the function to maximize the lower bound, but their accuracy in complex imagery is unclear.

The main idea of assigning pseudo-label in our method is to find the probability of the image patch given the class and assign the label to the image patch corresponding to the highest probability. To obtain the latent class distribution over the image patches, we need to fit the feature space with a statistical model. Latent Dirichlet Allocation (LDA) [25] is a good choice, which is a three-level hierarchical Bayesian model. Each item of a collection is modeled as the mixture of topics and each topic is modeled as mixture of the codebook. To apply LDA for image processing, we regard the whole images as documents, categories as topics, and small image patches as visual words. However, traditional LDA is a "bag-of-words" model and doesn't consider the spatial information at all, which is essential for image processing, therefore, we add spatial information in LDA by calculating the frequency of the category around the image patches. Different from Wang et al [26], which adds another layer between codebook and category, our method is simpler and easy to train. Motivated by active learning which allowed human in the loop to annotate data at each iteration, we also propose an iterative strategy to generate pseudo-labels. The key idea of our strategy is to use previously learned knowledge to improve the model learning by adding pseudo-labels inferred from previous knowledge.

In this paper, we propose a novel framework to generate pseudo-labels iteratively depending only on the original sparsely labels. To summarize, the contributions of this paper are as following. Firstly, we propose a simple yet effective framework to image semantic segmentation based on the sparsely point-supervision. Secondly, we modify the Latent Dirichlet allocation (LDA) by adding spatial coherence and use latent distribution as the criterion to generate pseudo-labels iteratively. Finally, we add mutual information constraint between the input and feature space to get a good representation.

The rest of this paper is organized as follows. Section 2 provides the overview of our method and describes each part of our framework in detail. Section 3 shows the results of the proposed method in coral images dataset compared with other semi-supervised approaches, and ablation study for the impact of different components of our method. Conclusion and some future works are mentioned in Section 4.

2. Materials and Methods

In this section, we first provide the overview of the proposed method and then formulate coral image segmentation as the semi-supervised image classification problem. Detail description for each part in Figure 2 are also demonstrated.

As we can see, our framework, summarized in Figure 2, consists of three steps. Starting from a randomly initialized network. The first step is to train the network from labeled samples and mutual information constrain between input and latent features. The second step is to employ spatial coherence LDA in the embedding of the network trained in the previous step to infer the category distribution over latent features and generate pseudo-labels. The third step is to train the neural network on the entire training set, with labeled

samples, pseudo-labeled samples and unlabeled samples. The pseudo-labeled samples are weighted per samples and per class.

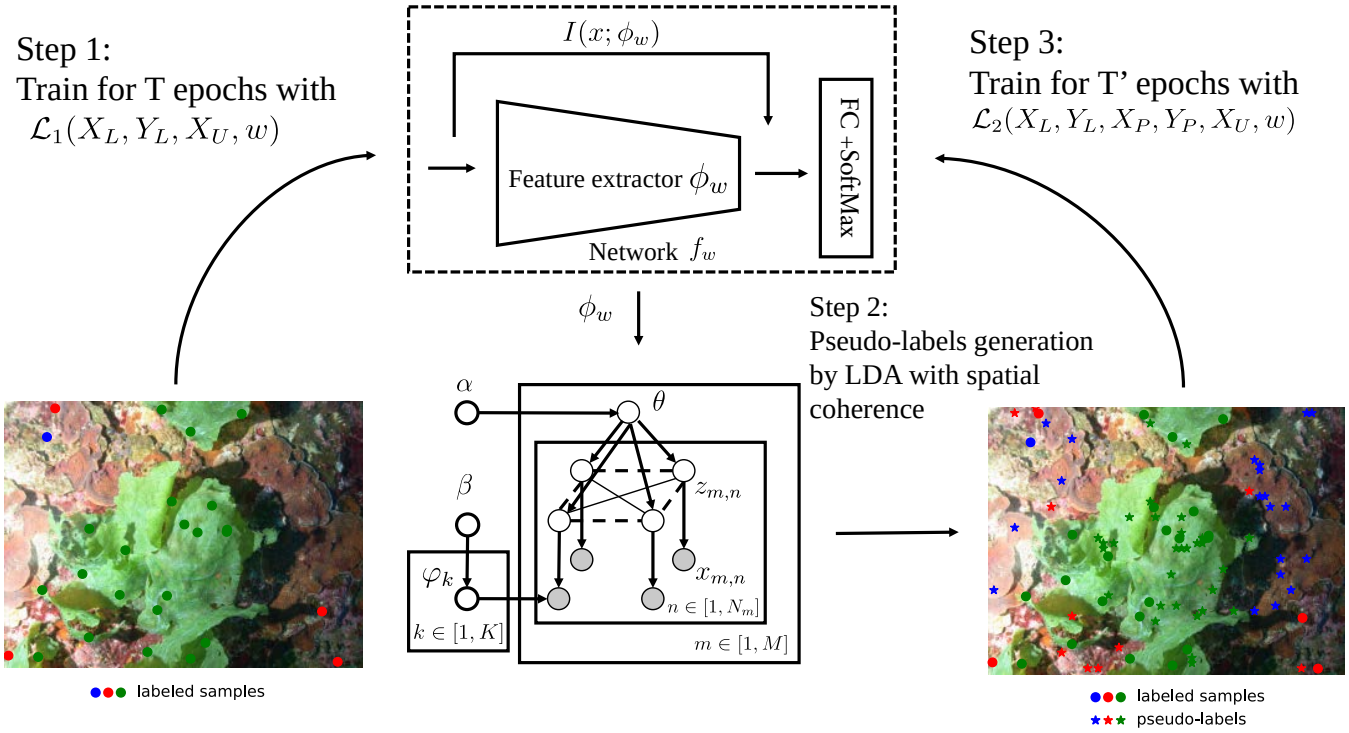

Figure 2. Framework of the proposed method.

2.1. Preliminaries

In this section, we first formulate the semi-supervised coral images segmentation and then we discuss the loss function that used in our work. For semi-supervised classification, we assume a collection of n examples $X = (x_1, x_2, ..., x_l, x_{l+1}, ..., x_n)$ with $x_i \in \mathcal{X}$. The first l examples x_i for $i \in L = \{1, ..., l\}$ denoted by X_L are labeled by $Y_L = (y_1, y_2, ..., y_l)$ with $y_i \in C$, where $C = \{1, ..., c\}$ is a discrete label set for c class. The remaining $u = n - l$ examples x_i for $i \in U = \{l + 1, ..., n\}$, denoted by X_U, are unlabeled. The goal is to use all X (image patches) and only small label size Y_L (point-supervision) to train a classifier to identify the class of unlabeled samples X_U. In practical conditions, the number of samples in label set is much smaller than that in the unlabeled set. For our coral images dataset, there are only 0.0015% labeled pixels.

The neural network takes input examples from $\mathcal{X}$ and produce a vector of class probability. We denote it by $f_w : \mathcal{X} \to \mathbb{R}^c$, where w represents the parameters of network. Function f_w is the mapping from the input space to the class space. The output of the network for ith example is $f_w(x_i)$ and the prediction is the index of maximum probability, which is shown in Equation (1).

$$\hat{y}_i = \underset{j}{\operatorname{argmax}} f_w(x_i)_j, \tag{1}$$

where subscript j denotes the j-th dimension of probability vector corresponding to the j-th class. Basically, we need an objective function and the goal is to minimize it, which is nothing but to take the derivative of the loss function respect to the parameters w. There are two stages for our method. First, we train a classifier with labels and mutual information constraint to get a good feature representation. Then, we generate pseudo-labeled samples via spatial LDA in the feature space extracted in the first stages and add them in the training set.

The objective function ($\mathcal{L}_1$) for the first stage consists of two component: supervised loss($\mathcal{L}_s$) and mutual information constraint loss ($\mathcal{L}_{MI}$) shown in Equation (2). where the minus sign is added before mutual information constraint loss because we want maximize the mutual information.

$$\mathcal{L}_1(X_L, Y_L, X_U, w) = \mathcal{L}_s(X_L, Y_L, w) - \lambda_{MI}\mathcal{L}_{MI}(X_U, w), \tag{2}$$

The objective function ($\mathcal{L}_2$) for the second stage consists of three component: supervised loss ($\mathcal{L}_s$), pseudo-label loss ($\mathcal{L}_p$) and mutual information constraint loss ($\mathcal{L}_{MI}$), which is shown in Equation (3), we bring in the pseudo-labeled samples information in loss function.

$$\mathcal{L}_2(X_L, Y_L, X_p, Y_p, X_U, w) = \mathcal{L}_s(X_L, Y_L, w) + \lambda_p\mathcal{L}_p(X_p, Y_p, w) - \lambda_{MI}\mathcal{L}_{MI}(X_U, w), \tag{3}$$

The network is trained by minimizing a supervised loss term ($\mathcal{L}_s$) on labeled samples in X_L, which is shown in Equation (4). A standard choice of l_s in classification is cross-entropy loss. Pseudo-label loss ($\mathcal{L}_P$) is the second component in $\mathcal{L}_2$, which is applied only to pseudo-labeled samples. Y_p represents the pseudo-labels of X_p and $\hat{y}_i$ in Equation (5) denotes the pseudo-labels for each example x_i for $i \in U$. This label is assigned according to the latent class distribution from LDA with spatial information described in Section 2.3.

$$\mathcal{L}_s(X_L, Y_L, w) = \frac{1}{l}\sum_{i=1}^{l} l_s(f_w(x_i), y_i), \tag{4}$$

$$\mathcal{L}_p(X_p, Y_p, w) = \frac{1}{n_p}\sum_{i=1}^{n_p} l_s(f_w(x_i), \hat{y}_i), \tag{5}$$

The third component in $\mathcal{L}_2$ is the mutual information between input space and latent features space shown in Equation (6). The reason why add such term is that we want to obtain a good representation combining not only the label information but also the input structure information. The classifier (dotted rectangle box in Figure 1) is conceptually divided in two parts. The first part is feature extraction network $\phi_w : \mathcal{X} \to \mathbb{R}^d$, mapping the input to a d dimension feature vector, we denote it by $\mathbf{v_i} = \phi_w(x_i)$ for i-th input sample x_i. The second classifier part typically consists of a fully connected layer appied on the top of ϕ_w followed by softmax layer.

$$\mathcal{L}_{MI}(X_U, w) = \frac{1}{n_u}\sum_{i=1}^{n_u} I(x_i; \phi_w(x_i)), \tag{6}$$

The classifier of choice is Wide Residual Networks (WRN) [27] which is widely used in many semi-supervised methods for image classification. It consists of an initial convolutional layer and three groups of residual blocks followed by average pooling and final fully connected layer. The main difference between WRN and ResNet [28] is that the number of kernels is larger than that of ResNet, which achieves better representation.

2.2. Feature Extraction with Information Maximization

In order to get a good feature representation for the input samples, we require that the feature space not only contains the label information but also preserves the input sample structure as well. Therefore, we maximize the mutual information between the input space and feature space. The loss function is in Equation (7):

$$\mathcal{L}_1(X_L, Y_L, X_U, w) = \frac{1}{|\mathcal{D}_l|}\sum_{x_l, y_l \in \mathcal{D}_l} \mathrm{H}(f_w(x_l), y_l) - \lambda\frac{1}{|X_U|}\sum_{x_u \in X_U} I(x_u; \phi_w(x_u)), \tag{7}$$

The first term is the cross entropy loss between predict and true labels, the second term is mutual information between input and its corresponding features.

For completeness, we review briefly bellow the matrix-based Rényi's α-order entropy functional on positive definite matrices and how to use it for calculating mutual information. We first give the definition of entropy and joint entropy and then provide the equation to calculate the mutual information.

Definition 1. *Let $\kappa : \chi \times \chi \mapsto \mathbb{R}$ be a real valued positive definite kernel that is also infinitely divisible. Given $\{\mathbf{x}_i\}_{i=1}^{n} \in \chi$, each $\mathbf{x}_i$ can be a real-valued scalar or vector, and the Gram matrix $K \in \mathbb{R}^{n \times n}$ computed as $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$, a matrix-based analogue to Rényi's α-entropy can be given by the following functional:*

$$H_\alpha(A) = \frac{1}{1-\alpha} \log_2((A^\alpha)) = \frac{1}{1-\alpha} \log_2 \left(\sum_{i=1}^{n} \lambda_i(A)^\alpha \right), \tag{8}$$

where $\alpha \in (0,1) \cup (1,\infty)$. A is the normalized version of K, i.e., $A = K/tr(K)$. $\lambda_i(A)$ denotes the i-th eigenvalue of A.

Definition 2. *Given n pairs of samples $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{n}$, each sample contains two measurements $\mathbf{x} \in \chi$ and $\mathbf{y} \in \gamma$ obtained from the same realization. Given positive definite kernels $\kappa_1 : \chi \times \chi \mapsto \mathbb{R}$ and $\kappa_2 : \gamma \times \gamma \mapsto \mathbb{R}$, a matrix-based analogue to Rényi's α-order joint-entropy can be defined as:*

$$H_\alpha(A, B) = H_\alpha \left(\frac{A \circ B}{(A \circ B)} \right), \tag{9}$$

where $A_{ij} = \kappa_1(\mathbf{x}_i, \mathbf{x}_j)$, $B_{ij} = \kappa_2(\mathbf{y}_i, \mathbf{y}_j)$ and $A \circ B$ denotes the Hadamard product between the matrices A and B.

Given Equations (8) and (9), the matrix-based Rényi's α-order mutual information $I_\alpha(A; B)$ in analogy of Shannon's mutual information is given by:

$$I_\alpha(A; B) = H_\alpha(A) + H_\alpha(B) - H_\alpha(A, B), \tag{10}$$

Throughout this work, we use the Gaussian kernel $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2})$ to obtain the Gram matrices. For each sample, we evaluate its k ($k = 10$) nearest distances and take the mean. We choose kernel width σ as the average of mean values for all samples. Further information and the analytical gradient of Equation (10) are shown in Appendix A.

2.3. LDA with Spatial Information

In this section, we first give a briefly introduction of traditional LDA and then we modify the LDA by adding local spatial information. LDA is one of the most popular generative models originally developed for natural language processing, which contains a three-level hierarchical structure. Recently, it has developed rapidly in the field of image processing such as image segmentation, classification and annotation. When LDA is applied to image processing, we treat the classes of objects as topics, local patches of images as words and the whole image as a document. A codebook is created by clustering all the local descriptors in the image set using K-means. Each local patch is quantized into a visual word according to the codebook. The graphical model of traditional LDA is shown in Figure 3. There are M images in the dataset. Each image m has N_m image patches. $\mathbf{v}_{m.n}$ is the observed feature value of the local image patch n in image m, $z_{m,n}$ denotes the hidden class for $\mathbf{v}_{m,n}$. All the local image patches in the corpus will be clustered into K classes. Each image m is modeled as a multinomial distribution ($p(z_{m,n} \mid \vec{\theta}_m)$) with parameter $\vec{\theta}_m$ over classes and similarly each category k is modeled as a multinomial distribution ($p(\mathbf{v}_{m,n} \mid \vec{\varphi}_z)$) with parameter $\vec{\varphi}_z$ over the visual codebook, and α, β are Dirichlet prior

for multinormal distribution. Equation (11) shows the LDA model, $\vec{\theta}_m$ and $\vec{\varphi}_z$ are hidden variables to be inferred. The generative process of LDA is shown in Algorithm 1.

$$P(\mathbf{v}_{m,n} \mid \vec{\theta}_m, \varphi) = \sum_z p(\mathbf{v}_{m,n} \mid z_{m,n}, \varphi_z) p(z_{m,n} \mid \vec{\theta}_m), \tag{11}$$

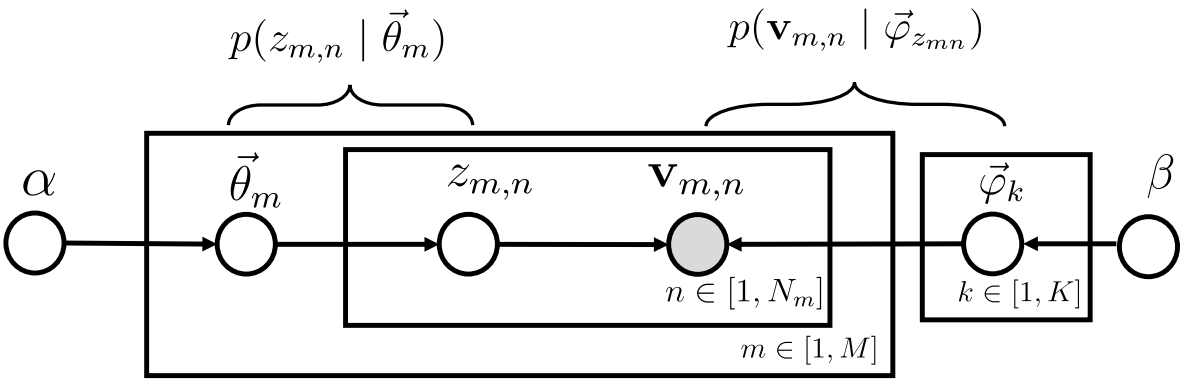

Figure 3. Graphical model of traditional LDA.

Algorithm 1 Generative process of LDA

1: Select the α and β, which are the parameters of Dirichlet distribution.
2: For a image m, a multinomial parameter θ_m is sampled from Dirichlet prior $\theta_m \sim Dirichlet(\alpha)$
3: For a category k, a multinomial parameter φ_k is sampled from Dirichlet prior $\varphi_k \sim Dirichlet(\beta)$.
4: For a image patch n in image m, its category z_{mn} is sampled from the image to category Multinomial distribution $z_{mn} \sim Multinomial(\theta_m)$.
5: The features $\mathbf{v}_{mn}$ of image patch n in image m, is sampled from the category to image patch features Multinomial distribution of topic z_{mn}, $\mathbf{v}_{mn} \sim Multinomial(\varphi_{z_{mn}})$.

Hidden category variable $z_{m,n}$ can be sampled through a Gibbs sampling [29] procedure which integrates out $\vec{\theta}_m$ and $\vec{\varphi}_k$. We fist randomly assign the class to each image patch and then determine the class according to Equation (12). More details about Gibbs sampling for LDA are shown in Appendix B.

$$\begin{aligned} & P(z_{mn} = k \mid \mathbf{v}_{mn} = t, \vec{z}_{mn}, \vec{x}_{mn}, \vec{\alpha}, \vec{\beta}) \\ & \propto P(\mathbf{v}_{mn} = t \mid z_{mn} = k) \cdot P(z_{mn} = k \mid \vec{z}_{mn}, \mathbf{v}_{mn}, \vec{\alpha}, \vec{\beta}) \\ & = \underbrace{\frac{n^k_{t,mn} + \beta_t}{\sum_{t'=1}^{T}(n^k_{t',mn} + \beta_{t'})}}_{\boldsymbol{\varphi}^k_t} \cdot \underbrace{\frac{n^m_{k,mn} + \alpha_k}{\sum_{k'=1}^{K}(n^m_{k',mn} + \alpha_{k'})}}_{\boldsymbol{\theta}^m_k}, \end{aligned} \tag{12}$$

where $n^k_{t,mn}$ is the number of visual words in the corpus with value t assigned to category k excluding visual word n in document m, and $n^m_{k,mn}$ is the number of visual words in document m assigned to category k excluding word m in document n. Equation (12) is the product of two ratios: the probability of visual word $\mathbf{v}_{mn} = t$ under category k (φ^{k_t}) and the probability of category k in document m (θ^m_k).

However, traditional LDA is a "bag of words" model and does not consider spatial information at all, which is essential for image processing. Therefore, we want to add spatial information in the original formulation based on the assumption that if visual words are from the same class of objects, they should also be close in space. So we group image patches which are close in space. One straightforward way is to calculate the frequency

of the category in the neighborhood of the image patch and add it to the corresponding conditional category distribution. Therefore we change the category distribution term from $p(z_{m,n} \mid \vec{\theta}_m)$ to $p(z_{m,n} \mid \vec{\theta}_m, z_{m,n_i})$ and bring the local information, which is shown in Equation (13). The LDA graphical model with spatial coherence is shown in Figure 4.

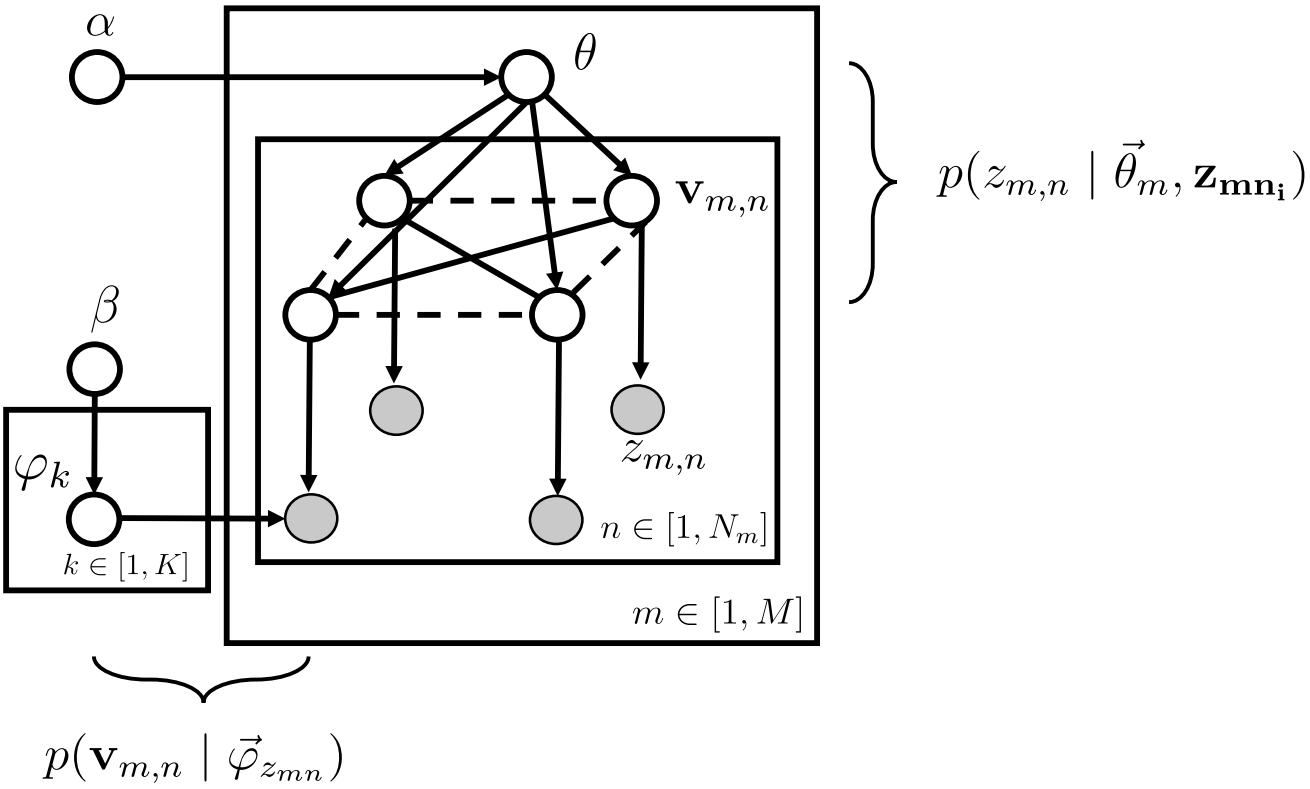

Figure 4. Graphical model of LDA with spatial information.

$$
\begin{aligned}
&P(z_{mn} = k \mid x_{mn} = t, \vec{z}_{mn}, \vec{x}_{mn}, \vec{\alpha}, \vec{\beta}) \\
&\propto P(x_{mn} = t \mid z_{mn} = k) \cdot P(z_{mn} = k \mid z_{mn_i}, \vec{z}_{mn}, \vec{x}_{mn}) \\
&= \underbrace{\frac{n^k_{t,mn} + \beta_t}{\sum_{t'=1}^{T}(n^k_{t',mn} + \beta_{t'})}}_{\boldsymbol{\varphi^k_t}} \cdot \underbrace{\left((1-\lambda) \cdot \frac{n^m_{k,ji} + \alpha_k}{\sum_{k'=1}^{K}(n^m_{k,mn} + \alpha_{k'})} + \lambda \cdot \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}(z_{mn_i} = k)\right)}_{\boldsymbol{\theta^m_k}},
\end{aligned} \tag{13}
$$

where λ is a trade-off parameter to change the weight of the local spatial information, z_{mn_i} represents the i-th image patch's category of N neighborhoods for z_{mn}. Recall that the indicator function $\mathbf{1}(z_{mn_i} = k)$ equals 1 if and only if $z_{mn_i} = k$. Equation (13) shows that the category of the image patch is more likely to belong to the neighborhood's category than Equation (12). In this paper, we set $N = 8$ denoting eight connected neighborhoods of the center image patch. The LDA generative process with spatial coherence is almost the same to original LDA (Algorithm 1), except that category z_{mn} is sampled from $P(z_{mn} = k \mid z_{mn_i}, \vec{z}_{mn}, \vec{x}_{mn})$. Algorithm 2 demonstrates inference for parameter θ_m and φ_k of LDA with spatial information using Gibbs sampling.

Algorithm 2 Gibbs sampling for LDA with spatial coherence

Input: image patch feature values matrix ($M \times H \times W$), the number of categories K, initial category of each image patch features.
Output: $\vec{\theta}_m$ and $\vec{\varphi}_k$.
for each iteration T **do**:
  for each image m **do**:
    for each image patch n **do**:
      Sampling category of nth image patch based on Equation (13).
    end for
  end for
end for
Estimate the $\vec{\theta}_m$ and $\vec{\varphi}_k$.

2.4. Pseudo-Label Generation

In this section, we will introduce how to generate pseudo-labels based on LDA illustrated in Figure 5. The three heatmaps in the middle column represent higher probability over image patch codebooks in areas with coral, red algae and green algae, respectively (from top to bottom) according to the category distribution (left-hand side of Figure 5). We annotate the pseudo-labels (star point) in the sample image at right-hand side. We calculate the distance between the pseudo-labeled samples and the original labeled samples to determine the class for each cluster.

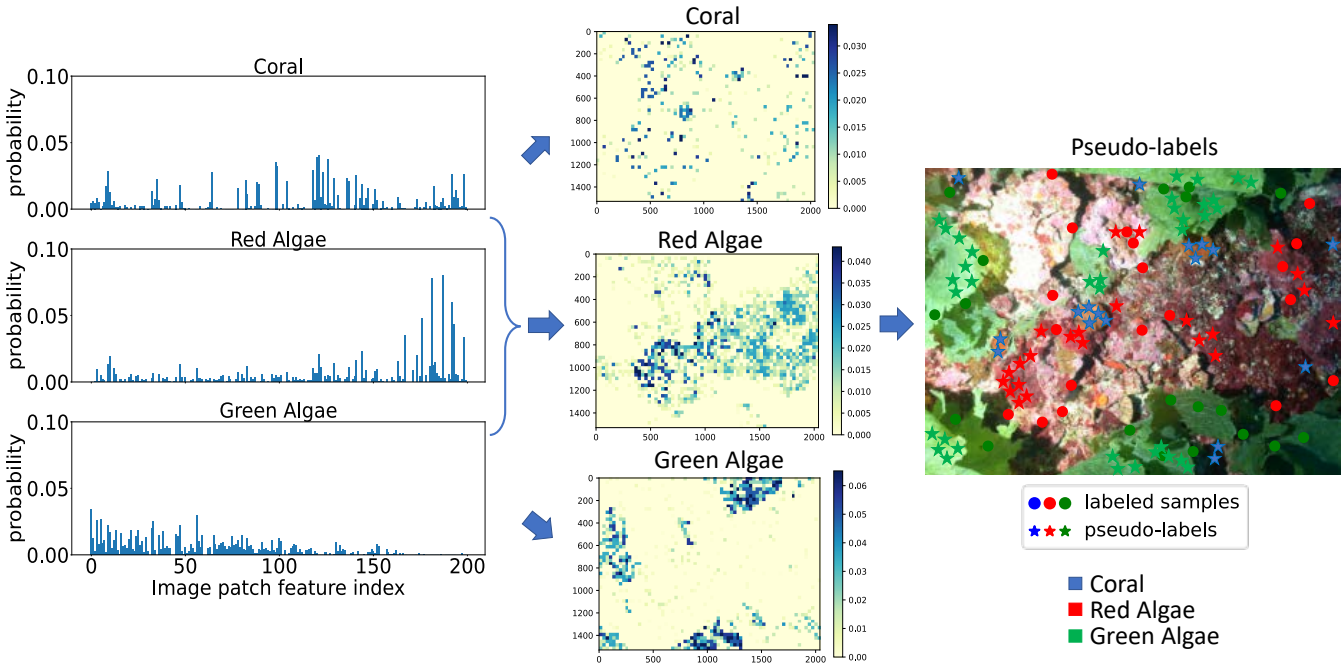

Figure 5. Pseudo-labels generation.

One of the problems for generating pseudo-labels is that low-quality features extracted by the neural network at early training stages may mislead the training process into a wrong direction and such wrong information can spread to the following training process. To overcome this problem, we come up with a confidence level for each pseudo-labeled sample, which indicates how reliable the pseudo-label is. For each labeled sample $x_i \in X_L$, we always set its confidence level $r = 1$. For each pseudo-labeled sample $x_p \in X_U$, we

compute r using Equation (14), based on the assumption that x_p will be more reliable if it is located in densely populated regions.

$$r_{x_p} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\sqrt{2\pi}\sigma}exp\Big(-\frac{(\phi_w(x_p)-\phi_w(x_i))^2}{2\sigma^2}\Big), \tag{14}$$

where, x_i is the original labeled sample and x_p is the pseudo-labeled sample we generated. We adopt kernel density estimation to estimate the probability of pseudo-labeled samples within the label samples in the feature space. We use Gaussian kernel and for each sample, we evaluate its $k(k = 10)$ nearest distances and take the mean. We choose the average of mean values for all samples as the kernel size σ. When the pseudo-labeled samples are far away from the original labeled samples, we can get the small confidence level r.

In addition, we also introduce the class weight (ζ_j of class j) to deal with the issue of class imbalance. ζ_j is defined in Equation (15), which is inversely proportional to class population.

$$\zeta_j = (|L_j| + |P_j|)^{-1}, \tag{15}$$

where $|L_j|$ denotes the number of class j in labeled samples and $|P_j|$ represents the number of class j of generated pseudo-labels.

2.5. Iterative Training

After pseudo-label generation, we will train the neural network with labeled samples, pseudo-label samples and unlabeled samples together using objective function shown in Equation (16).

$$\begin{aligned}\mathcal{L}_2(X_L, Y_L, X_P, Y_P, X_U, r, \zeta) = &\frac{1}{|\mathcal{D}_l|}\sum_{x_l, y_l \in \mathcal{D}_l}\zeta_y \mathrm{H}(f_w(x_l), y_l) + \lambda_p \frac{1}{|\mathcal{D}_p|}\sum_{x_p, y_p \in \mathcal{D}_p}\zeta_{y_p} r_{x_p}\mathrm{H}(f_w(x_p), y_p)\\ &- \lambda_{MI}\frac{1}{|\mathcal{D}_U|}\sum_{x_u \in \mathcal{D}_U} I(x_u; \phi_w(x_u)).\end{aligned} \tag{16}$$

As can be seen, there are three terms in Equation (16). The first term is cross-entropy between predict of labeled samples and its corresponding true labels, the second term is cross-entropy between predict of unlabeled samples and its corresponding pseudo-labels, and the last term is mutual information between unlabeled samples and its corresponding features. λ_p and λ_{MI} are the hyper-parameters to adjust the importance of them.

Given the image patch feature extraction, pseudo-labels generation and neural network training with labeled samples, pseudo-labeled samples and unlabeled samples, we plug these components into an iterative learning process. First, we train the network for T epochs with labeled samples and mutual information constraint using Equation (7). Second, we obtain the class distribution over feature visual words via spatial LDA. Third, we assign pseudo-labels to unlabeled image patches by selecting higher probability in class distribution. Finally, we train the network on the entire dataset using Equation (16) for $T^{'}$ epochs. We repeat this iterative process for M iterations. The above steps are summarized in Algorithm 3.

Algorithm 3 GeneratePseudo-labels iteratively via spatial LDA

$w \leftarrow$ initialize randomly
for epoch $\in [1, ..., T]$ **do**
  $w \leftarrow \text{Optimize}(\mathcal{L}_1(X_L, Y_L, X_U, w))$ Equation (7) ▷ mini-batch optimization
for Iteration $\in [1, ..., M]$ **do**
  for $i \in [1, ..., n]$ **do** $\mathbf{v}_i \leftarrow \phi_w(x_i)$ ▷ feature descriptors
  $V_w \leftarrow \text{K-means}(\mathbf{v}_i)$ ▷ visual feature codebook
  $P_c \leftarrow \text{Gibbs sampling}(V_w)$ Equation (13) ▷ probability of class given unlabeled samples
  for $c \in [1, ..., C]$ **do** $y_p^c \leftarrow \text{argmax}_c P_{c|x_p}$ ▷ pseudo-labels
  for $x_p \in [1, ..., X_p]$ **do** $r_{x_p} \leftarrow$ Equation (14) ▷ confidence level
  for $j \in C$ **do** $\zeta_j \leftarrow (|L_j| + |P_j|)^{-1}$ ▷ class weight
  for epoch $\in [1, ..., T]$ **do**
    $w \leftarrow \text{Optimize}(\mathcal{L}_2(X_L, Y_L, X_P, Y_P, X_U, r, \zeta))$ Equation (15) ▷ mini-batch optimization
  end for
end for

3. Results

In this section, we first describe coral image data set used in our experiments and semi-supervised image segmentation setup. Then, we discuss the training details for our method. Finally, we perform the experiments to compare with other semi-supervised image classification approaches and show the impact of different components involved in the our proposed method.

3.1. Dataset

For the coral image data set, which is collected from Pulley Ridge region in the Gulf of Mexico. There are 120 images with only 50 labeled pixels for each image, the size of each image is 2048×1536. For each human label, we select 30×30 pixel patch centered at the label. We use 100 images for training and 20 images for testing. The number of image patches for training is 5000. We select 4000 image patches samples for training and 1000 for validation. There are five classes: corals, rock, green algaes, red algaes and others.

3.2. Experiments Setup and Training Details

Experiments on coral image dataset are performed with Wide Residual Networks (WRN). Specifically, we used "WRN-28-2", i.e., ResNet with 28 convolutional layers and the number of kernels is twice as that of ResNet, including average pooling, batch normalization and leaky ReLU nonlinearities. For training, the size of input image patch is 30×30 and we chose the Adam optimizer [30], with 0.001 learning rate and 64 batch size for labeled samples, 128 batch size for unlabeled samples. We set the $\lambda_{MI} = 0.1$, and linearly ramp up λ_p to its maximum value (we set it as 10 in our experiment) over the 500 epochs during the training. We employ the mean intersection over union criterion (mIOU) [31] to quantify our proposed method.

We first train the network for 100 epochs with only sparse point-level labels and mutual information constraint between input and the output of the last layer before the softmax. Then, we use K-means to construct the visual codebook in the feature space. The codebook size is 200 and the dimension of feature visual word is 128. The way to assign the pseudo-labels for unlabeled image patches is as follows: we first find the 10 highest probability features for each class based on the class over feature visual word distribution obtained by spatial LDA and assign such features as that class label. Then we go back to whole image to search the image patches and give them the same pseudo-labels as its corresponding features. Finally, we train the neural network with labeled samples, pseudo-labeled samples and unlabeled samples together for 500 epochs. We repeat the above steps 5 to 10 times and generate about 5000 pseudo-labeled samples for each iteration.

3.3. Parameter Analysis and Performance Comparison

We first show the performance with different codebook size which is essential for our experiments. It is obvious that the small codebook cannot represent all image patches, while a large codebook size will improve the computational complexity in inferring the parameters of LDA. So, we select an appropriate codebook size according to the image segmentation results shown in Figure 6. As can be seen, when the codebook size is larger than 200, the performance starts decreasing slowly. Therefore, we set codebook size as 200 in our experiments.

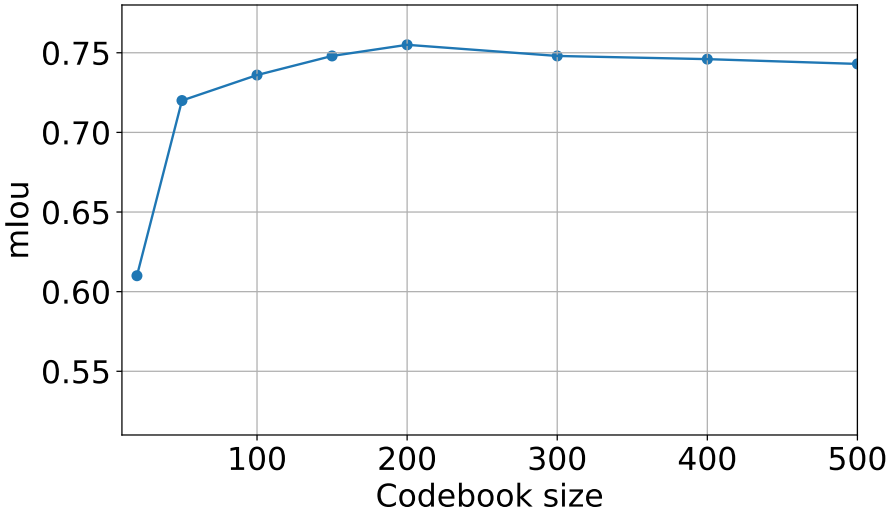

Figure 6. Performance with different codebook size.

Then, we compare our method with other semi-supervised methods in Table 1. As we can see, pseudo-labeling methods are more accurate than supervised approach (use sparse labels only). Entropy minimization, virtual adversarial training (VAT) and Π-model work better than pseudo-labeling. Our proposed method performs better than other competing methods and when combined with VAT, we can achieve the best performance against others. The way we combine VAT is to add another adversarial consistency loss term (mean square error between original sample and its corresponding adversarial example) in Equation (4) at stage 2 training. Figure 7 shows the results of coral images segmentation for different methods. As can be seen, our proposed method can detect coral well and the areas are more smooth than other approaches.

Table 2 shows the abundance of coral, green algae and red algae detected by different methods on the coral images test dataset. Our proposed method performs much better than others especially for coral and red algae detection.

Table 1. Performance on coral images dataset with different semi-supervised model.

Method	mIOU
Supervised	60.8%
EntMin	69.3%
Π-model	68.5%
Pseudo-labeling	61.7%
VAT	74.7%
Ours	74.8%
Ours + VAT	75.1%

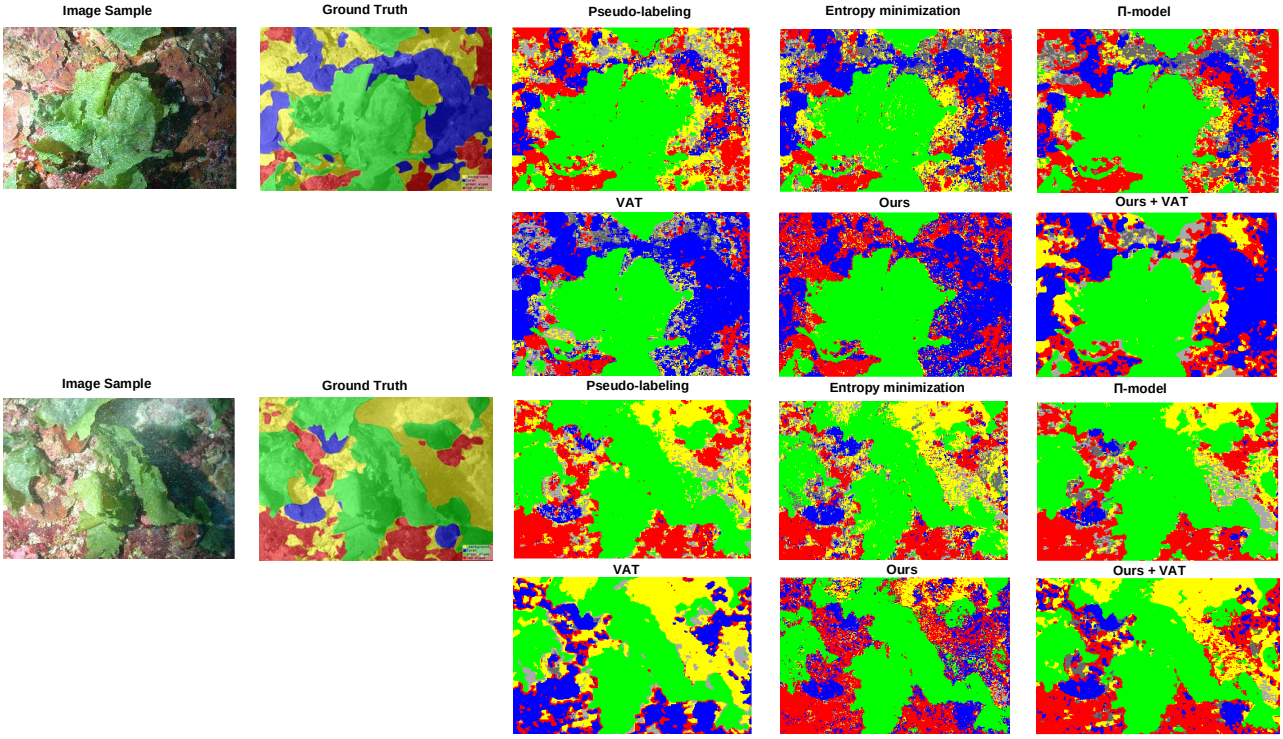

Figure 7. Coral image segmentation on test dataset with different methods. Blue color represents coral, green color represents green algae, red color represents red algae, gray color represents rock and yellow color represents other species.

Table 2. Coral, green algae and red algae abundance detected with different method on test dataset.

Method	Coral	Green Algae	Red Algae
Ground Truth	15.7%	43.6%	20.4%
Supervised	10.2%	40.2%	22.5%
EntMin	11.6%	42.5%	22.3%
Π-model	12.5%	41.8%	25.3%
Pseudo-labeling	10.8%	42.6%	22.9%
VAT	20.6%	43.3%	17.5%
Ours	17.0%	43.8%	21.5%
Ours + VAT	16.8%	43.1%	21.8%

3.4. Ablation Study

We investigate the impact of different component of our proposed approach. First, we show the benefit of using weighting strategy (confidence level r_i for samples and class weights ζ_j for different classes) for generated pseudo-labeled samples. Green and orange curves in Figure 8 shows that our weight strategy has positive contribution.

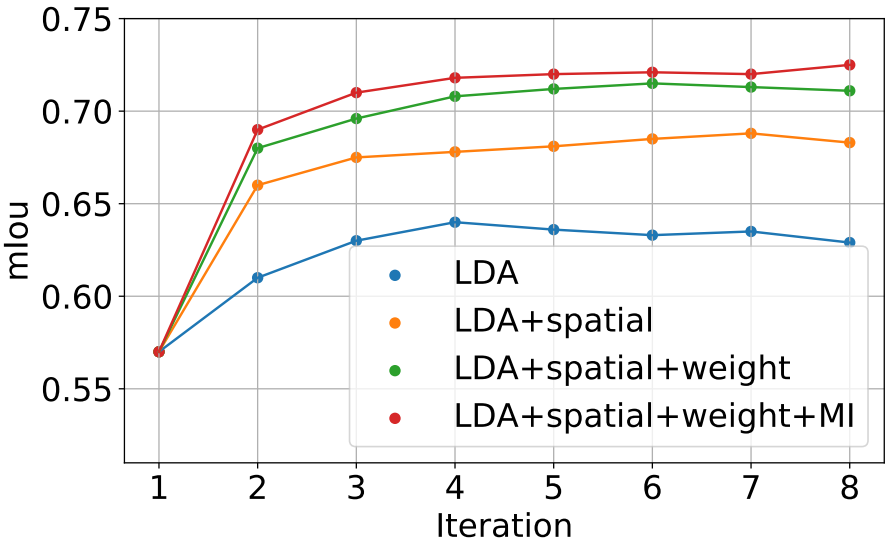

Figure 8. Performance on coral images test dataset with different component of proposed method.

Then, we study the effectiveness of including spatial coherence in LDA. Figure 9a shows the value of log-likelihood during the Gibbs sampling process for LDA with or without spatial coherence. λ denotes the weight to adjust the importance of spatial information. As can be seen, when adding spatial information, the performance improves (the higher log-likelihood the better), and we can achieve the best performance when $\lambda = 0.01$ corresponding green curve in Figure 9. Similarly, we also plot the log-likelihood for LDA with or without mutual information constraint in Figure 9b, which shows that the features extracted with MI constraints are better than without MI constraints.

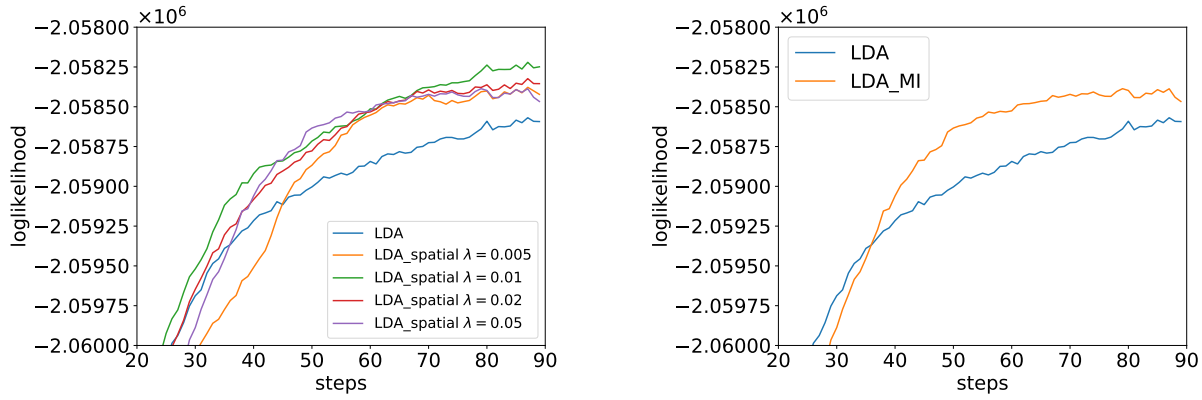

Figure 9. (**a**) log-likelihood for LDA with or without spatial coherence; (**b**) log-likelihood for LDA with or without MI regularization.

Table 3 and Figure 8 demonstrate that weighting strategy, spatial coherence and MI constraints in our proposed method have positive contributions for coral images segmentation. Spatial coherence in LDA considers the local patch information, and bring weights for pseudo-labeled samples can reduce the bad effect of labeling errors. MI constraints introduced in the loss function achieve the better representation for feature extraction.

4. Conclusions

In this paper, we propose a novel and effective framework to generate pseudo-labels iteratively only depending on sparsely labels. The results in the coral image data set from

Table 3. Ablation study results on coral images test dataset.

Method	mIOU
LDA	63.8%
LDA + spatial	69.3%
LDA + spatial + weights	73.5%
LDA + spatial + weights + MI	74.8%

Pulley Ridge show that our approach can generate more correct pseudo-labels and help us get a better result for image segmentation against other semi-supervised method. The main advantage of generating pseudo-label iteratively is that previously learned knowledge can be incorporated to improve the model learning and final results. However, the limitation of our method is that for the under represented classes, i.e., classes that have a low percentage of the overall pixels, our method does not work well. Nevertheless, our method is a productive way to tell human experts what kind of classes should be more annotated, and which classes already have sufficient labels to yield good identification results.

Future works may follow four directions: First, we think that metric learning may quantify the uncertainty of the pseudo-labels by including distance in the input space, latent feature space and label space. Second, we want to improve the information theoretic methods to obtain more useful information besides the label information. Third, we want to change the current architecture for image patch classification to a fully convolutional network. One of the obvious weakness of the current architecture is that the network can only see the small size image patches but cannot obtain the whole image structure. Last but not least, we want to develop a graphical user interface (GUI) software to allow humans in the loop interaction to guide the annotation of more useful labels.

Author Contributions: Conceptualization, X.Y., B.O. and J.C.P.; methodology, X.Y. and J.C.P.; software, X.Y.; validation, X.Y.; formal analysis, X.Y. and J.C.P.; investigation, X.Y. and J.C.P.; writing original draft preparation, X.Y. and J.C.P.; supervision, B.O. and J.C.P.; project administration, B.O. and J.C.P.; funding acquisition, B.O. and J.C.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work is funded by Office of Ocean Exploration and Research (award NA14OAR4320260).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Data is available from the authors.

Acknowledgments: We would like to thank Stephanie Farrington, John Reed and Brian Cousin for preparing the coral image dataset.

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

Abbreviations

The following abbreviations are used in this manuscript:

LDA	Latent Dirichlet Allocation
MI	Mutual Information
MINE	Mutual Information Neural Estimation
VAT	Virtual Adversarial Training
mIOU	mean Intersection Over Union
WRN	Wide Residual Networks
EntMin	Entropy Minimization

Appendix A. Mutual Information and the Gradient of Matrix-Based Entropy Functional

$$\begin{aligned} I(\mathbf{x};\mathbf{y}) &= \int\int \mathsf{P}(\mathbf{x},\mathbf{y}) \log\left(\frac{\mathsf{P}(\mathbf{x},\mathbf{y})}{\mathsf{P}(\mathbf{x})\mathsf{P}(\mathbf{y})}\right) d\mathbf{x}d\mathbf{y} \\ &= -\int\left(\int \mathsf{P}(\mathbf{x},\mathbf{y})d\mathbf{y}\right)\log \mathsf{P}(\mathbf{x})d\mathbf{x} - \int\left(\int \mathsf{P}(\mathbf{x},\mathbf{y})d\mathbf{x}\right)\log \mathsf{P}(\mathbf{y})d\mathbf{y} \\ &+ \int\int \mathsf{P}(\mathbf{x},\mathbf{y})\log \mathsf{P}(\mathbf{x},\mathbf{y})d\mathbf{x}d\mathbf{y} \\ &= -\int \mathsf{P}(\mathbf{x})\log \mathsf{P}(\mathbf{x})d\mathbf{x} - \int \mathsf{P}(\mathbf{y})\log \mathsf{P}(\mathbf{y})d\mathbf{y} + \int\int \mathsf{P}(\mathbf{x},\mathbf{y})\log \mathsf{P}(\mathbf{x},\mathbf{y})d\mathbf{x}d\mathbf{y} \\ &= H(\mathbf{x}) + H(\mathbf{y}) - H(\mathbf{x},\mathbf{y}), \end{aligned} \tag{A1}$$

where $H(\cdot)$ denote the entropy and $H(\cdot,\cdot)$ denotes the joint entropy.

It is not hard to verify that our measure has analytical gradient. In fact, we have:

$$\frac{\partial S_\alpha(A)}{\partial A} = \frac{\alpha}{(1-\alpha)}\frac{A^{\alpha-1}}{\mathrm{tr}(A^\alpha)}, \tag{A2}$$

$$\frac{\partial S_\alpha(A,B)}{\partial A} = \frac{\alpha}{(1-\alpha)}\left[\frac{(A\circ B)^{\alpha-1}\circ B}{\mathrm{tr}(A\circ B)^\alpha} - \frac{I\circ B}{\mathrm{tr}(A\circ B)}\right] \tag{A3}$$

and

$$\frac{\partial I_\alpha(A;B)}{\partial A} = \frac{\partial S_\alpha(A)}{\partial A} + \frac{\partial S_\alpha(A,B)}{\partial A} \tag{A4}$$

Since $I_\alpha(A;B)$ is symmetric, the same applies for $\frac{\partial I_\alpha(A;B)}{\partial B}$ with exchanged roles between A and B.

Appendix B. Gibbs Sampling for the LDA Topic Model

There are two processes of LDA: one is $\vec{\alpha} \to \vec{\theta} \to z_{m,n}$, the other is $\vec{\beta} \to \vec{\varphi}_k \to x_m \mid k = z_m$. Hidden variable z follows multinomial distribution which is shown in Equation (A5) and we show the equations of the first process as follows.

$$P(\vec{z} \mid \vec{\theta}) = \prod_{i=1}^{K} \theta_i^{n_i} \tag{A5}$$

$$\begin{aligned} Dirichlet(\vec{\theta} \mid \vec{\alpha}) &= \frac{\Gamma(\alpha_1+\alpha_2+\cdots+\alpha_k)}{\Gamma(\alpha_1)+\Gamma(\alpha_2)+\cdots+\Gamma(\alpha_k)}\prod_{i=1}^{K}\theta_i^{\alpha_i-1} \\ &= \frac{1}{\Delta(\vec{\alpha})}\prod_{i=1}^{K}\theta_i^{\alpha_i-1} \end{aligned} \tag{A6}$$

Because the Dirichlet distribution is the conjugate prior of the multinomial distribution, so the form of the distribution for $\vec{\theta}$ given $\vec{z}$ has the same form as Dirichlet distribution, which is shown in Equations (A6) and (A7). We select the expectation value of the posterior as the value of the variable $\vec{\theta}$ which is shown in Equation (A8). In order to get the joint distribution, we also calculate the conditional distribution of x and z in Equations (A9) and (A10).

$$P(\vec{\theta} \mid \vec{z})) \sim Dirichlet(\vec{\theta} \mid (\vec{\alpha} + \vec{n})) \tag{A7}$$

$$\vec{\theta} = \left(\frac{n_1+\alpha_1}{\sum_{i=1}^{K}(n_i+\alpha_i)}, \frac{n_2+\alpha_2}{\sum_{i=1}^{K}(n_i+\alpha_i)}, \cdots, \frac{n_K+\alpha_K}{\sum_{i=1}^{K}(n_i+\alpha_i)}\right) \tag{A8}$$

$$\begin{aligned} P(\vec{z} \mid \vec{\alpha}) &= \int P(\vec{z} \mid \vec{\theta}) P(\vec{\theta} \mid \vec{\alpha}) d\vec{\theta} \\ &= \int \prod_{i=1}^{K} \theta_i^{n_i} Dirichlet(\vec{\theta} \mid \vec{\alpha}) d\vec{\theta} \\ &= \int \prod_{i=1}^{K} \theta_i^{n_i} \frac{1}{\Delta(\vec{\alpha})} \prod_{i=1}^{K} \theta_i^{\alpha_i - 1} d\vec{\theta} \\ &= \frac{1}{\Delta(\vec{\alpha})} \int \prod_{i=1}^{K} \theta_i^{n_i + \alpha_i - 1} d\vec{\theta} \\ &= \frac{\Delta(\vec{\alpha} + \vec{n})}{\Delta(\vec{\alpha})} \end{aligned} \tag{A9}$$

Similarly, we can get the distribution of $\vec{x}$ given by $\vec{z}$ and $\vec{\beta}$.

$$P(\vec{x} \mid \vec{z}, \vec{\beta}) = \prod_{i=1}^{K} \frac{\Delta(\vec{\beta}_i + \vec{n}_i)}{\Delta(\vec{\beta}_i)} \tag{A10}$$

Therefore, we can get the joint distribution for the image patch feature $\vec{x}$ and its category $\vec{z}$, which is shown as follows.

$$\begin{aligned} P(\vec{x}, \vec{z} \mid \vec{\alpha}, \vec{\beta}) &= P(\vec{z} \mid \vec{\alpha}) P(\vec{x} \mid \vec{z}, \vec{\beta}) \\ &= \frac{\Delta(\vec{\alpha} + \vec{n})}{\Delta(\vec{\alpha})} \prod_{i=1}^{K} \frac{\Delta(\vec{\beta}_i + \vec{n}_i)}{\Delta(\vec{\beta}_i)} \end{aligned} \tag{A11}$$

We use the Gibbs sampling algorithm, which is one of the Markov chain Monte Carlo (MCMC) methods to estimate the parameters of the LDA model.

$$\begin{aligned} P(z_i = k \mid \vec{z}_i, \vec{x}) &= P(z_i = k \mid x_i = t, \vec{z}_i, \vec{x}_i) \\ &= \frac{P(z_i = k, x_i = t \mid \vec{z}_i, \vec{x}_i)}{P(x_i = t \mid \vec{z}_i, \vec{z}_i)} \end{aligned} \tag{A12}$$

$$P(\vec{\theta} \mid \vec{z}_i, \vec{x}_i) = Dirichlet(\vec{\theta} \mid \vec{\alpha} + \vec{n}_i) \tag{A13}$$

$$P(\vec{\varphi}_k \mid \vec{z}_{k,i}, \vec{x}_{k,i}) = Dirichlet(\vec{\varphi}_k \mid \vec{\beta}_k + \vec{n}_{k,i}) \tag{A14}$$

Finally, we can get the Gibbs sampling equation by combining Equations (A11)–(A13), which is shown as follows:

$$\begin{aligned} &P(z_i = k \mid \vec{z}_i, \vec{x}) \propto P(z_i = k, x_i = t \mid \vec{z}_i, \vec{x}_i) \\ &= \int P(z_i = k, x_i = t, \vec{\theta}, \vec{\varphi}_k \mid \vec{z}_i, \vec{x}_i) d\vec{\theta} d\vec{\varphi}_k \\ &= \int P(z_i = k, \vec{\theta} \mid \vec{z}_i, \vec{x}_i) d\vec{\theta} \int P(x_i = t, \vec{\varphi}_k \mid \vec{z}_{k,i}, \vec{x}_{k,i}) d\vec{\varphi}_k \\ &= \int \vec{\theta}_k Dir(\vec{\theta} \mid \vec{\alpha} + \vec{n}_i) d\vec{\theta} \int \vec{\varphi}_{k,t} Dir(\vec{\varphi}_k \mid \vec{\beta}_k + \vec{n}_{k,i}) d\vec{\theta} \\ &= E(\theta_k) E(\varphi_{k,t}) \\ &= \hat{\theta}_k \hat{\varphi}_{k,t} \end{aligned} \tag{A15}$$

In addition, we can estimate $\hat{\theta}_k$ and $\hat{\varphi}_{k,t}$ through the following equations.

$$\hat{\theta}_k = \frac{n_{k,ji}^{j} + \alpha_k}{\sum_{k'=1}^{K} (n_{k,ji}^{j} + \alpha_{k'})} \tag{A16}$$

$$\hat{\varphi}_{k,t} = \frac{n^k_{t,ji} + \beta_t}{\sum_{t'=1}^{T}(n^k_{t',ji} + \beta_{t'})} \tag{A17}$$

$$P(z_{ji} = k \mid x_{ji} = t, \vec{z}_{ji}, \vec{x}_{ji}, \vec{\alpha}, \vec{\beta}) \propto$$
$$= \frac{n^j_{k,ji} + \alpha_k}{\sum_{k'=1}^{K}(n^j_{k,ji} + \alpha_{k'})} \cdot \frac{n^k_{t,ji} + \beta_t}{\sum_{t'=1}^{T}(n^k_{t',ji} + \beta_{t'})} \tag{A18}$$

References

1. Vijay, B.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495.
2. 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; Springer: Cham, Switzland, 2015; pp. 234–241
3. 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.
4. Laine, S.; Aila, T. Temporal Ensembling for Semisupervised Learning. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
5. Miyato, T.; Maeda, S.H.; Ishii, S.; Masanori, K. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 1979–1993. [CrossRef] [PubMed]
6. Yu, X.; Ma, Y.; Farrington, S.; Reed, J.; Ouyang, B.; Principe, J.C. Fast segmentation for large and sparsely labeled coral images. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–6.
7. Yu, X.; Ouyang, B.; Principe, J.C.; Farrington, S.; Reed, J. Fast focus of attention for corals from underwater images. In *Ocean Sensing and Monitoring XI*; International Society for Optics and Photonics: Baltimore, ML, USA, 2019; Volume 11014, p. 1101408.
8. Lee, D.-H.Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In Proceedings of the Workshop on Challenges in Representation Learning, International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; Volume 3.
9. Grandvalet, Y.; Bengio, Y. Semi-Supervised Learning by Entropy Minimization. In Proceedings of the Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 5–8 December 2005.
10. Shi, W.; Gong, Y.; Ding, C.; Ma, Z.; Tao, X.Y.; Zheng, N. Transductive semi-supervised deep learning using min-max features. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018.
11. Antti, T.; Harri, V. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA; London, UK, 2017; Volume: 30, pp. 1195–1204.
12. Charles, B.; Cornebise, J.; Kavukcuoglu, K.; Wierstra, D. Weight uncertainty in neural networks. *arXiv* **2015**, arXiv:1505.05424.
13. Alex, K.; Badrinarayanan, V.; Cipolla, R. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv* **2015**, arXiv:1511.02680.
14. Zhu, X.; Lafferty, J.; Rosenfeld, R. Semi-Supervised Learning with Graphs. Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2005
15. Ahmet, I.; Tolias, G.; Avrithis, Y.; Chum, O. Label propagation for deep semi-supervised learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 5070–5079.
16. David, B.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; Raffel, C.A. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*; IRTF: Vancouver, BC, Canada, 2019; pp. 5049–5059.
17. Hongyi, Z.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv* **2017**, arXiv:1710.09412.
18. Gonzalez-Rivero, M.; Beijbom, O.; Rodriguez-Ramirez, A.; Bryant, D.E.; Ganase, A.; Gonzalez-Marrero, Y.; Hoegh-Guldberg, O. Monitoring of coral reefs using artificial intelligence: A feasible and cost-effective approach. *Remote Sens.* **2020**, *12*, 489. [CrossRef]
19. Akbari Asanjan, A.; Das, K.; Li, A.; Chirayath, V.; Torres-Perez, J.; Sorooshian, S. Learning Instrument Invariant Characteristics for Generating High-resolution Global Coral Reef Maps. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Conference, 2020; pp. 2617–2624. Available online: https://dl.acm.org/doi/abs/10.1145/3394486.3403312 (accessed on 3 February 2021).
20. Modasshir, M.; Rekleitis, I. Enhancing Coral Reef Monitoring Utilizing a Deep Semi-Supervised Learning Approach. In IEEE International Conference on Robotics and Automation (ICRA), Paris, France 31 May–31 August 2020; pp. 1874–1880.
21. Giraldo, L.G.S.; Rao, M.; Principe, J.C. Measures of entropy from data using infinitely divisible kernels. *IEEE Trans. Inf. Theory* **2014**, *61*, 535–548. [CrossRef]
22. Yu, S.; Giraldo, L.G.S.; Jenssen, R.; Principe, J.C. Multivariate Extension of Matrix-based Renyi's α-order Entropy Functional. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *42*, 2960–2966. [CrossRef] [PubMed]
23. Fischer, A.A.A.I.; Dillon, J.V.; Murphy, K. Deep variational information bottleneck. *arXiv* **2016**, arXiv:1612.00410.

24. Ishmael, B.M.; Baratin, A.; Rajeswar, S.; Ozair, S.; Bengio, Y.; Courville, A.; Hjelm, R.D. Mine: Mutual information neural estimation. *arXiv* **2018**, arXiv:1801.04062.
25. Blei, D.M.; Andrew, Y.N.; Jordan, M.I. Latent dirichlet allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
26. Xiaogang, W.; Grimson, E. Spatial latent dirichlet allocation. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA; London, UK, 2007; pp. 1577–1584.
27. Sergey, Z.; Komodakis, N. Wide residual networks. *arXiv* **2016**, arXiv:1605.07146.
28. Kaiming, H.; 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.
29. Ian, P.; Newman, D.; Ihler, A.; Asuncion, A.; Smyth, P.; Welling, M. Fast collapsed gibbs sampling for latent dirichlet allocation. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 569–577.
30. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
31. Intersection over Union (IoU) for Object Detection. Available online: https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/ (accessed on 3 February 2021).