# Deep Learning for Semantic Segmentation of Coral Reef Images Using Multi-View Information

Andrew King[1] Suchendra M. Bhandarkar[1,2] Brian M. Hopkinson[3]
[1]Institute for Artificial Intelligence [2]Department of Computer Science [3]Department of Marine Sciences
The University of Georgia, Athens, Georgia 30602, USA
andrewking@uga.edu suchi@cs.uga.edu bmhopkin@uga.edu

## Abstract

*Two major deep learning architectures, i.e., patch-based convolutional neural networks (CNNs) and fully convolutional neural networks (FCNNs), are studied in the context of semantic segmentation of underwater images of coral reef ecosystems. Patch-based CNNs are typically used to enable single-entity classification whereas FCNNs are used to generate a semantically segmented output from an input image. In coral reef mapping tasks, one typically obtains multiple images of a coral reef from varying viewpoints either using stereoscopic image acquisition or while conducting underwater video surveys. We propose and compare patch-based CNN and FCNN architectures capable of exploiting multi-view image information to improve the accuracy of classification and semantic segmentation of the input images. We investigate extensions of the conventional FCNN architecture to incorporate stereoscopic input image data and extensions of patch-based CNN architectures to incorporate multi-view input image data. Experimental results show the proposed TwinNet architecture to be the best performing FCNN architecture, performing comparably with its baseline Dilation8 architecture when using just a left-perspective input image, but markedly improving over Dilation8 when using a stereo pair of input images. Likewise, the proposed nViewNet-8 architecture is shown to be the best performing patch-based CNN architecture, outperforming its single-image ResNet152 baseline architecture in terms of classification accuracy.*

**Index terms:** *Underwater imaging, coral reef imaging, coral reef classification, deep learning, semantic segmentation, 3D reconstruction, multi-view integration*

## 1. Introduction

The challenge of generating an accurate and repeatable map of the underlying ecosystem has been a significant limiting factor in ecological studies of marine environments, especially coral reefs. Manual *in situ* mapping performed underwater by human divers is extremely time consuming, whereas aerial photography and satellite remote sensing are both severely limited by the fact that seawater absorbs light strongly, thereby limiting monitoring to very shallow coral reefs [11]. Acoustic methods, although capable of mapping the ocean floor at a large spatial scale, are simply unsuitable for mapping marine ecosystems at finer spatial scales.

Within the larger field of marine ecology, the subfield of coral reef ecology has been receiving increasing attention in recent times on account of the fact that coral reefs across the globe are facing increasing threats from both, natural and anthropogenic stressors. These stressors, which include climate change, ocean acidification, sea level rise, pollutant runoff, and overfishing [2, 13], have combined to cause rapid declines in coral reef ecosystems worldwide over the past three decades, resulting in a global marine environmental crisis [3]. The precarious state of coral reef ecosystems worldwide lends special urgency to the task of advancing current mapping and monitoring technologies so that accurate detection and precise quantification of changes in coral reef ecosystems at appropriate scales of temporal and spatial resolution is possible.

With recent advances in autonomous underwater vehicles (AUVs) equipped with high-resolution cameras, *in situ* surveys are being increasingly replaced by image- and video-based robotic surveys. This has led to growing interest within the research communities in marine biology and marine ecology in exploiting computer vision and machine learning techniques for high-throughput automatic analysis and annotation of benthic images [5, 6, 23, 24, 26, 28]. Computer vision and machine learning techniques are now enabling the generation of detailed, large-scale maps of underwater environments [14]. AUVs traveling systematically through coral reef environments are able to continuously acquire high-quality images of small portions of the coral reef ecosystem. Using computer vision and machine learning algorithms, the individual images are automatically assembled into a large-scale, 3D reconstruction (or map) of the coral reef ecosystem accompanied by semantic classification of the various coral taxa, thereby permitting one to es-

timate the spatial distribution of these taxa on the coral reef.

In spite of recent advances in automating the map generation process using computer vision and machine learning techniques, significant challenges remain to be addressed. Conventional computer vision and machine learning techniques entail the segmentation of regions of interest in the input image or video stream, characterization of these regions using features, classification of these regions into predefined categories using the extracted features, and 3D reconstruction and recognition of objects from the results of the classification. Traditional computer vision and machine learning algorithms employ hand-crafted or pre-engineered features, models and classifiers for the purpose of 3D object reconstruction and recognition. A major shortcoming of pre-engineered features and classifiers is their lack of adaptability when faced with underwater images that typically exhibit significant variability due to changes in illumination, water turbidity, strong ocean currents and presence of algal blooms [4]. One of the promises of deep learning neural network architectures is their ability to replace hand-crafted or pre-engineered features with efficient algorithms for hierarchical feature learning and feature extraction from the underlying image data, which is especially important in computer vision [18].

Modern deep learning neural network architectures for semantic image segmentation typically fall into one of two major categories. The first category comprises of the *fully convolutional neural network* (FCNN) architectures, which segment and classify an image on a per-pixel basis using a single end-to-end trainable network. The second category comprises of patch-based *convolutional neural network* (CNN) architectures that classify image segments or image patches into predefined classes, thus generating a semantically segmented image. In the case of the patch-based CNNs, the image segments or image patches are typically generated using well known algorithms such as simple linear iterative clustering (SLIC) [1] or graph cuts [7, 8].

When semantic image segmentation is performed in the context of mapping tasks, such as in remote sensing or underwater imaging, the scene objects or entities of interest are often captured from multiple viewpoints. However, in conventional approaches to semantic image segmentation, only a single image is utilized for the purpose of classification. In the context of coral reef monitoring, a typical semantic image segmentation pipeline comprises of image acquisition and image filtering followed by image registration and mosaicking where a composite 2D image is created in order to generate a full map of the underlying coral reef. In conventional computer vision-based approaches to automatic annotation of benthic data, further analysis is done on the composite 2D image wherein most of the multi-view information is discarded. In this work we propose methods for utilizing this often discarded multi-view information with the aim of further improving the accuracy and precision of semantic image segmentation. Figure 1 depicts a typical computational pipeline for 3D reconstruction of a coral reef accompanied by the semantic classification of its constituent taxa.

In the case of the FCNN architecture, we investigate the possibility of using stereoscopic image pairs as input to achieve more accurate and precise semantic segmentation of coral reef images. We employ a well-known stereo imaging algorithm to generate a disparity map from the left- and right-perspective rectified coral reef images [12]. The disparity map is added as a fourth channel (complementing the existing three RGB color channels) to each coral reef image that is input to the FCNN. The disparity map, which encodes 3D disparity information, is expected to guide the semantic segmentation of the coral reef images. In addition, we propose the *TwinNet* architecture which is based roughly on the Siamese network architecture [16, 19] and accepts both the left- and right-perspective images as input. From these stereo images the *TwinNet* architecture can learn the disparity map and/or relevant spatial features that may prove useful for semantic image segmentation and classification.

In the case of the patch-based CNN, we explore the possibility of using input images taken from $n$ distinct viewpoints (where $n \geq 2$) to achieve a single-entity classification. We generate a 3D mesh representation of the coral reef surface and perform classification on each mesh face/facet to arrive at a 3D semantic segmentation of the coral reef surface. We investigate the impact of different voting schemes on the final classification accuracy. Furthermore, we propose the *nViewNet* architecture, which is capable of receiving a variable number of images (subject to a maximum number) and learning an optimal combination of these images to yield an accurate a single-entity classification.

## 2. Background and Related Work

In recent years, CNNs have continued to push the limits of accuracy for large-scale image classification tasks [17, 27, 31]. In the context of coral reef image classification, we have previously investigated the *VGG16* architecture [27] which represents a significant improvement over previous networks by its use of small $3 \times 3$ kernel filters instead of the larger kernel filters common at the time. We have also investigated the *InceptionV3* architecture [31] which improves upon previous CNN architectures via incorporation of an *inception module* that approximates an optimal sparse CNN, thereby allowing the *InceptionV3* architecture to deepen (i.e., add layers) while staying within common GPU memory constraints.

As CNNs have grown deeper over time allowing them to learn more complex patterns, dealing with progressively smaller gradient updates (i.e., the *vanishing gradient prob-*

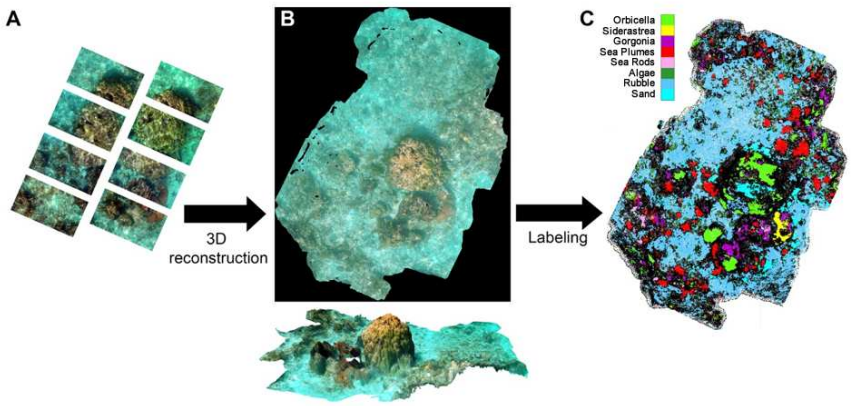



Figure 1: Computational pipeline for 3D reconstruction and annotation of a coral reef ecosystem.

*lem*) has become critical. The *ResNet* architecture [10] addresses the vanishing gradient problem by incorporating residual convolution blocks that attempt to fit a mapping of the residual as opposed to computing a direct mapping. By using *skip connections* that pass information directly from the first layer of the block to the last, the *ResNet* architecture preserves the gradient across several CNN layers thus allowing the network to deepen without the gradients becoming vanishingly small. In particular, the 152-layer *ResNet152* architecture has been shown to perform particularly well on the task of coral reef classification [15]. Recent work has shown the *ResNet152*-derived features (termed as *ResFeats*) to outperform conventional CNN-derived features for binary and multi-class classification of coral reefs [21, 22]. We adopt *ResNet152* as the baseline architecture for many of the models proposed in this paper. Finally, we also consider the *Inception-ResNetV2* architecture [30], which combines the *Inception* architecture with the *ResNet* residual block architecture.

Among the existing FCNN architectures for simultaneous semantic image segmentation and object classification, we consider the *FCN8s* architecture [20] and the *Dilation8* architecture [33]. The *FCN8s* architecture represents the first successful attempt to repurpose an existing CNN architecture designed for image classification, i.e., *VGG16*, for the task of semantic image segmentation. Long et al. [20] eliminate the fully connected CNN layers in the *VGG16* architecture, replacing them with convolution layers, and employ transposed convolution to upsample the output. This results in an end-to-end trainable model for semantic image segmentation, eliminating the need for separate segmentation and patch-wise classification phases. However, the *FCN8s* architecture requires whole-image ground truth segmentation maps for the purpose of training where the training loss is evaluated by comparing the network output against the ground truth segmentation map.

The *Dilation8* architecture [33] incorporates enhancements to the *FCN8s* architecture to improve accuracy. *Dilation8* removes many of the max pooling layers in the *VGG16* base of *FCN8s*, thereby eliminating the need for transposed convolution for upsampling. Rather than using iteratively larger kernels to maintain a large receptive field, *Dilation8* effectively dilates the convolution kernel. Since the number of kernel parameters are unchanged, the computational requirements of *Dilation8* are almost the same as those of *FCN8s*. In this paper, we compare the performance of the *FCN8s* and *Dilation8* architectures with and without the inclusion of a disparity channel.

Su et al. [29] formulate a multi-view CNN (MVCNN) as a classifier network for 3D shape-based object recognition from multiple images. The MVCNN architecture accepts input from an array of 12 equidistant cameras and pools the resulting views using an element-wise max operation. Experimental results show that pooling of multiple views in the MVCNN yields improved accuracy over single-view CNNs when attempting 3D shape-based classification of an object in an image. In our work, we relax the constraints of the MVCNN to allow for a variable ($n$) number of views from randomly-placed cameras.

In our work, we use weight sharing schemes similar to those used in Siamese networks [16, 19] and the MVCNN [29]. Siamese networks learn a similarity function between two inputs instead of a simple classifier. Both inputs are fed through identical networks with the same learned weights. The similarity function is learned via weight sharing and by using a contrastive loss function for similarity comparison. We draw on this general idea in the design of the proposed *TwinNet* and *nViewNet* architectures, which take more than one image as input and share weights in their underlying base architecture.

## 3. Underwater Image Data Acquisition

The underwater stereoscopic coral reef survey image data used in the work reported in this paper were manually

collected from coral reefs off the Florida Keys by a team of swimmers and divers. An underwater stereo camera rig, comprising of a GoPro Dual Hero camera system, was used to collect the underwater video data while swimming over sections of the coral reef. The stereo camera rig was carried over the reef in a serpentine (i.e., lawn mower) pattern in order to capture a complete section of the seafloor. Stereo image pairs were extracted from the video data at a rate of two frames per second. The resulting 2,391 stereo image pairs were deemed to comprise the Underwater Stereoscopic Coral Reef Survey of the Florida Keys image data bank (USCSF) and used for all the experiments reported in this paper. The work reported in this paper was conducted under permits granted by the Florida Keys National Marine Sanctuary (FKNMS-2016-042, FKNMS-2017-035).

## 4. Stereoscopic FCNN Architecture

In this section we describe the proposed extensions to the conventional FCNN architecture, enabling it to accept and analyze stereoscopic image data as input. We propose two extensions; the first involves feeding the stereoscopic disparity as a distinct channel to the conventional FCNN, complementing the conventional RGB input channels, and the second, termed as the *TwinNet* architecture is based roughly on the Siamese network architecture [16, 19] and accepts both the left- and right-perspective images as input.

### 4.1. Image Data Collection

During image data collection for the proposed FCNN models, it is necessary to generate dense pixelwise ground truth segmentation maps that could be subsequently used for training the models. Since we deal with stereoscopic image data, we acquire both, a left-perspective image and a right-perspective image. However, we create ground truth segmentation maps only for the left-perspective images. The creation of the ground truth segmentation maps is a time-consuming process, especially in the case of underwater image data, wherein the image clarity is often compromised on account of high water turbidity, ocean currents and floating algae and debris. Consequently, we designed a customized image annotation tool to streamline the process of generating ground truth segmentation maps.

The customized image annotation tool provides two methods for image segmentation: (a) superpixel generation via simple linear iterative clustering (SLIC) [1] and (b) computation of graph cuts [7, 8]. The image annotation tool allows the user to segment the input images and annotate the image regions using predefined class labels. It has a tunable parameter to allow for over-segmentation or under-segmentation of the input image, and also offers modes for automated annotation as well as manual annotation. A user can quickly generate segmentation maps upon segmenting the input image and annotating a subset of the resulting regions. The annotation tool uses RGB histograms and Gabor filter features to measure region similarity and performs $k$-means clustering based on the similarity measure. If a single region within a cluster is observed to possess a class label, then that class label propagated to the remaining regions within the cluster.

Using the aforementioned annotation tool, we were able to quickly generate 413 dense semantic segmentation and classification maps for training the FCNN models [20]. The ground truth semantic segmentation maps entail classification of each image pixel into one of the following 10 classes: (1) *Acropora palmata*, (2) *Orbicella spp.*, (3) *Siderastrea siderea*, (4) *Porites astreoides*, (5) *Gorgonia ventalina*, (6) sea plumes, (7) sea rods, (8) algae, (9) rubble, and (10) sand. Furthermore, we employ an *ignore* class for regions that do not fall into one of the aforementioned categories. These include objects such as *fish* or regions that are deemed unclassifiable by an expert. The *ignore* class does not contribute to the loss calculations and is therefore never a classification output generated by the FCNN. Furthermore, regions from the *ignore* class are not used in computing the classification accuracy on the validation set. The first four classes, i.e., *A. palmata*, *Orbicella spp.*, *Siderastrea siderea*, and *P. astreoides*, represent the different species of coral commonly found on reefs in the Florida Keys. The remaining single-species class, i.e., *Gorgonia ventalina*, represents the common sea fan. The remainder of the classes are multi-species classes or general coral classes.

### 4.2. Image Data Preprocessing

Upon collection, we process the image data before input to the FCNN models. Due to the large size of each image in the dataset ($2700 \times 1400$ pixels), it was deemed necessary to split each image into four quadrants for further processing on an Nvidia GTX 1080 GPU with a batch size of one. Since the coral reef environment contains 10 classes of interest, the preprocessing stage generates output images with pixel values between 0 and 9 in the respective color channels. Furthermore, we also subtract the mean color value of the training set from each input image to normalize the image data before input to the FCNN.

### 4.3. FCNN with Disparity Channel

We first examine the use of stereoscopic disparity as a means to leverage multi-view information in the proposed FCNN models. The images are first rectified using the camera calibration parameters. We create a disparity map using a well-known semi-global block matching-based disparity estimation algorithm [12] with a uniqueness threshold and block size of 15, a contrast threshold of 0.5 and a disparity range of 64. Upon creation of the disparity map, we interpolate the missing disparity data in the shadow regions of the disparity map using an efficient image inpainting algorithm [32]. The resulting disparity map is then regarded as an additional fourth channel that is concatenated with

the three RGB channels of the left-perspective image before input to the FCNN model. We compare the performance of two FCNN architectures, i.e., FCN8s [20] and Dilation8 [33], using the standard three-channel RGB input and the four-channel RGB plus disparity input, where the RGB inputs are derived from the left-perspective image in both cases.

To ensure experimental validity, we partition the image dataset into two subsets, a training set and a testing set using a 80:20 training set to testing set split ratio. We train the FCNN models using the training set and then report the model performance on the unseen testing set. The overall pixelwise classification accuracy is reported across all the coral reef classes. Since the pretrained *Imagenet* weights [25] do not include a disparity channel, we train each FCNN model from scratch. We train initially with a relatively high learning rate of $1 \times 10^{-3}$ to learn quickly some of the initial weights. We use the stochastic gradient descent algorithm with a Nesterov momentum term and a batch size of one as our optimization technique. We subsequently train the model using a learning rate of $1 \times 10^{-4}$ and weight decay rate of $1 \times 10^{-6}$ to refine the initially learned weights. Each FCNN model is trained for 15,000 iterations to ensure convergence, and the FCNN model with the highest validation accuracy is selected.

### 4.4. TwinNet: Stereoscopic FCNN

Instead of computing the disparity algorithmically using hand-crafted and/or pre-selected features derived from the stereo image pair (as is done in the case of the disparity FCNN), we seek to design a deep network architecture capable of learning the scene depth implicitly from the input stereo image pair for use in subsequent classification. We draw inspiration from the weight sharing schemes used in Siamese networks [16, 19] and the MVCNN [29]. Our base architecture is derived from the front-end of the *Dilation8* architecture [33]. The *Dilation8* architecture is, in turn, based on the *VGG-16* architecture [27], but employs dilated convolutions and fewer max pooling layers.

As depicted in Figure 2(a), the left- and right-perspective images are both input to the *Dilation8* base architecture, and the weights are shared at this point in the network. The left- and right-perspective outputs of the base architecture are then fed to a Siamese subnetwork comprising of two distinct submodules whose weights are learned independently. The outputs of the Siamese subnetwork are then fed to a stereo module consisting of three convolution layers. Each perspective's stereo module consists of three convolution layers and RELU activations with a two-dilated kernel size of three. The separated outputs of the stereo module are then concatenated on the channel axis and fed to a collapse module, which uses a convolution layer with a kernel size of one to reduce the number of channels to the total number of classes. At this point, the image is upsampled iteratively through transposed convolution and skip connections until it is returned to its original size as depicted in Figure 2(a).

Table 1: Results of FCNN stereo and disparity architectures

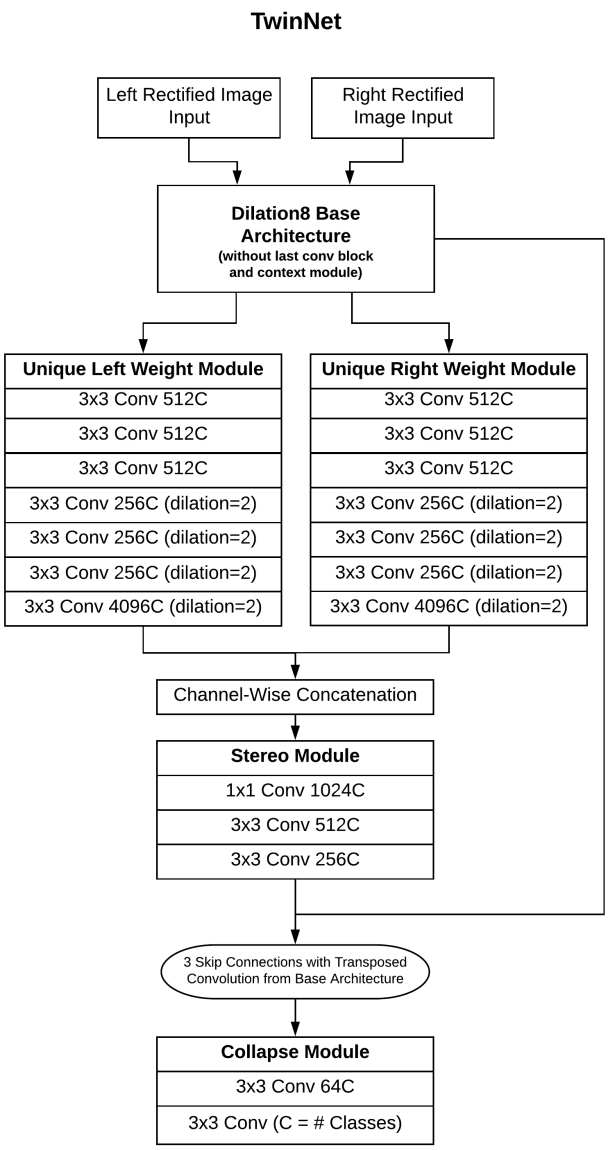
| Architecture | Accuracy (%) | Input |
|---|---|---|
| *FCN8s* | 50.45 | RGB |
| *Dilation8* | 62.84 | RGB |
| *FCN8s* | 53.82 | RGB + Disparity |
| *Dilation8* | 64.02 | RGB + Disparity |
| *TwinNet* (left image) | 61.93 | RGB |
| *TwinNet* | **66.44** | RGB Image Pair |

We compare performance of the proposed *TwinNet* architecture to that of its base *Dilation8* architecture. Furthermore we compare the performance of the *TwinNet* architecture under two scenarios, i.e., when provided with a stereo image pair and when provided only the left-perspective image as input. As before, we train our models with the training set and then report the model performance on the unseen testing set. The overall pixelwise classification accuracy across all classes is reported.

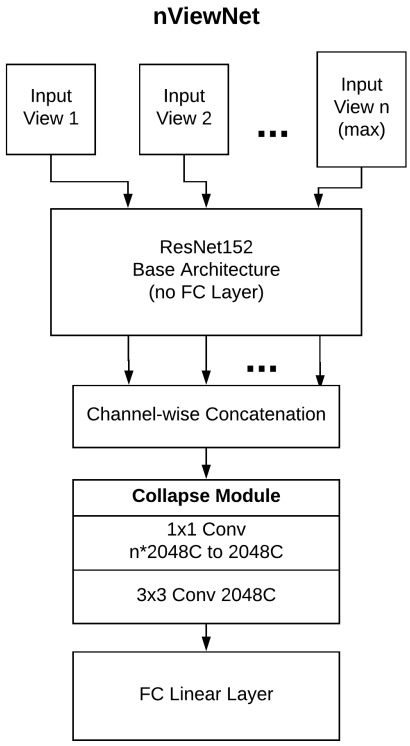
The base architecture weights are initialized using the *Imagenet* pretrained weights [25]. To retain the benefits of the pretraining, we freeze the base architecture weights and train the additional modules initially with a learning rate of $1 \times 10^{-3}$. We use a batch size of one and stochastic gradient descent with a Nesterov momentum term as our optimization technique. We then train the entire model using a learning rate of $1 \times 10^{-4}$ and weight decay rate of $1 \times 10^{-6}$. Each stereo FCNN model is trained for 7,000 iterations to ensure convergence, and the FCNN model with the highest validation accuracy is selected.

### 4.5. Performance of the FCNN Architectures

Table 1 summarizes the results of our experiments on extending the FCNN architecture for stereoscopic image input. The FCNNs with inclusion of the disparity channel were observed to yield a small increase in classification accuracy over their counterparts that utilize only the three RGB color channels. In the case of *FCN8s*, we observe a 3.37% increase in classification accuracy and in the case of *Dilation8* [33] we see a corresponding 1.18% improvement. These results show that inclusion of disparity information provides some benefit in terms of classification accuracy. The performance of the *TwinNet* architecture compares well with that of *Dilation8* when only the left-perspective image is used as input. However, the *TwinNet* architecture exhibits marked improvement over *Dilation8* when both, the left-perspective and right-perspective images are utilized. Therefore, it is reasonable to conclude that the intermediate layers of the *TwinNet* architecture learn more discriminative features from the stereo image pair than from a single image. Furthermore, the increased classification accuracy of




(a) TwinNet Architecture




(b) nViewNet Architecture

Figure 2: The proposed *TwinNet* and *nViewNet* architectures: *Conv* denotes the convolution layer, *C* denotes the channels and *FC* denotes fully connected.

the *TwinNet* architecture over *Dilation8* with the incorporation of the disparity channel demonstrates the superiority of the implicitly learned depth features over the hand-crafted RGB and disparity features.

## 5. Multi-View Patch-Based CNNs

### 5.1. Image Data Collection

A subset of the collected images from our coral reef image dataset (USCSF) was annotated by experts to provide ground truth pixel classifications. During the annotation process, an individual pixel in an image is selected in a pseudorandom fashion. The pixel is shown along with its spatial context to an expert who then assigns it to one of the following 10 classes: (1) *Acropora palmata*, (2) *Orbicella spp.*, (3) *Siderastrea siderea*, (4) *Porites astreoides*, (5) *Gorgonia ventalina*, (6) sea plumes, (7) sea rods, (8) algae, (9) rubble, and (10) sand.

We use a photogrammetric processing tool (Agisoft Photoscan) to generate a 3D mesh reconstruction of the underlying coral reef surface from our image dataset (USCSF) and determine the camera locations from which the images were taken. We assign a unique ID to each face of the 3D mesh. We match each pseudorandomly annotated pixel in an image with its corresponding mesh face ID via backprojection.

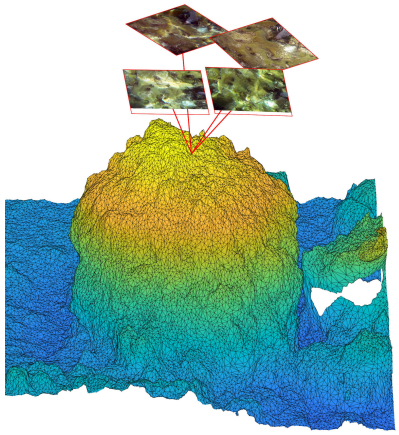


Figure 3: Projection from a 3D mesh to multiple images

As depicted in Figure 3, other views of the annotated mesh face are obtained by projecting the center of the mesh face into images using a standard projective camera model with extrinsic parameters (camera location and orientation) and intrinsic parameters (focal length, camera center, and radial distortion) obtained via an optimization procedure. In short, each mesh face ID is assigned a single class and associated with its corresponding location (patch) in one or more images. Our final dataset is comprised of 6,525 labeled meshes with 138,405 corresponding image patches.

### 5.2. Voting Networks

We propose a CNN architecture to handle a variable number of views of the underlying coral reef. Since our image data (USCSF) was extracted from survey video acquired in a serpentine fashion, any arbitrary point on the coral reef surface is likely to have been captured in multiple images from multiple points of view. The number of views for each coral reef surface point will vary and so, too, will the camera locations with respect to the points on the coral reef surface.

Our first approach to exploiting multiview information is incorporating a simple voting scheme within the CNN architecture. We train the *ResNet152* [10] CNN using a train/test stratification scheme where 80% of the data is used to train the network and 20% is used to test it. Each of the images from the training set (accompanied by the corresponding patch class labels) is used to train the *ResNet152* CNN [10]. The base architecture weights are initialized using the *Imagenet* pretrained weights [25]. We replace the last layer (i.e., the fully connected layer) in the *ResNet152* CNN with a different fully connected layer wherein the number of outputs equals the number of coral reef classes in our dataset. To retain the benefits of the pretraining, we freeze the base CNN weights and train the fully connected layer with a learning rate of $1 \times 10^{-3}$. We use a batch size of 64 and stochastic gradient descent with a Nesterov momentum term as our optimization technique. We then train the entire model using a learning rate of $1 \times 10^{-4}$ and weight decay rate of $1 \times 10^{-6}$.

Table 2: Results of multiview patch-based CNNs

| Network Architecture | Accuracy (%) | Batch Size |
|---|---|---|
| *ResNet152* | 85.54 | 32 |
| *ResNet152* + Simple Voting | 90.70 | 32 |
| *ResNet152* + Logit Pooling | 91.00 | 32 |
| *nViewNet-4* | 93.52 | 16 |
| *nViewNet-8* | 94.26 | 16 |

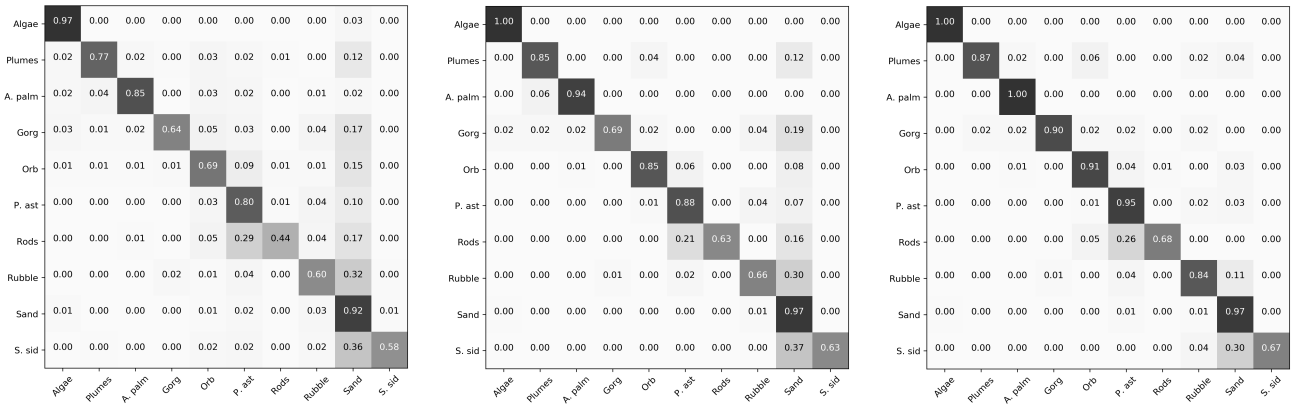
Each image corresponding to a mesh face in the validation set is then input to the trained network. Each image votes on a classification for that face, and the class label with a plurality of votes is output. As an alternative, we implement a pooling method which sums the logit values, essentially weighting each vote by its confidence value.

### 5.3. The nViewNet Architecture

We propose the *nViewNet* architecture to handle a variable number of multiple viewpoint images. The *nViewNet* uses the *ResNet152* [10] (without the fully connected layer) as a base architecture and, in similar fashion to the *TwinNet*, the base architecture weights are shared across all inputs. To ensure constant memory requirements and ease of training, we set a cap on the number of viewpoints to be included for classifying each mesh face. If the number of available viewpoints exceeds this preset cap or maximum, the additional viewpoint images are ignored and the retained viewpoint images selected at random.

Each viewpoint image, up to the preset maximum, is input to the base architecture, and the outputs fed to a collapse module. The collapse module takes two input images, each with $C$ channels, and concatenates them channel-wise. It then reduces the concatenated data, which has $2C$ channels, back to $C$ channels using a 2D convolution with a kernel size of one followed by another 2D convolution layer with a kernel size of three. The collapse module is invoked recursively to combine pairs of images in a tree-like fashion until only a single output remains as depicted in Figure 2(b). We use a linear transform to reduce the output of the collapse module to a vector of logits with a length equal to the number of classes. In the case where a mesh face is visible in fewer than the maximum allowable number of viewpoints, we invoke the collapse module with the existing viewpoints until the maximum is reached. We compare the results using a maximum of four viewpoints (*nViewNet-4*) and a maximum of eight viewpoints (*nViewNet-8*) to the previously proposed approaches.

The base architecture weights for *nViewNet* are initialized using the weights determined via training on the in-




(a) ResNet152 (b) ResNet152 with Logit Pooling (c) nViewNet-8

Figure 4: Confusion matrices for three patch-based multi-view architectures. We abbreviate *Acropora palmata* as *A. palm*, *Gorgonia ventalina* as *Gorg*, *Orbicella spp.* as *Orb*, *Porites astreoides* as *P. ast*, and *Siderastrea siderea* as *S. sid*.

dividual viewpoints. This initializes the base architecture with feature outputs that are already useful for classification. We ensure that the training set and testing set contain the same images for *nViewNet* and that the voting network model is trained on individual viewpoints to maintain experimental validity. To retain the benefit of the pretraining, we freeze the base architecture weights and train the additional modules initially with a learning rate of $1 \times 10^{-4}$. We use stochastic gradient descent with a Nesterov momentum term as our optimization technique.

### 5.4. Performance of Multi-view Patch-Based CNNs

Table 2 summarizes the experimental results for the voting, logit pooling and *nViewNet* architectures. All architectures are observed to outperform the underlying base architecture, i.e., *ResNet152* [10], when used by itself. Although logit pooling is essentially tied to simple voting, it does improve upon the latter. The best *nViewNet* architecture i.e., *nViewNet-8*, is seen to outperform its logit pooling and voting counterparts by 3.26% in terms of classification accuracy thus demonstrating the advantages of learning combined features from multiple viewpoints. Although the *nViewNet* architecture has an upper bound on the number of viewpoints it can handle, both, the *nViewNet-4* and *nViewNet-8* architectures were seen to outperform their voting and logit pooling counterparts that exploit every available viewpoint (with no imposed limit). The voting and logit pooling architectures however, are easier to implement and outperform the underlying *ResNet152* [10] base architecture by a significant margin.

## 6. Conclusions and Future Work

In this paper, we showed how integrating underwater coral reef images from multiple viewpoints could improve the classification accuracy of CNNs in the context of semantic segmentation and 3D reconstruction of coral reefs. We explored two FCNN architectures, one which uses the disparity map computed from a stereoscopic image pair as an additional input channel, complementing the three RGB color channels, and the other, termed as the *TwinNet* which is based on a Siamese network architecture and accepts both the left- and right-perspective images of the stereoscopic image pair as input. We also explored a patch-based CNN, termed as the *nViewNet* that accepts a variable number of multiple-viewpoint images as input while generating a 3D semantic segmentation of the coral reef. We also explored the incorporation of voting and logit pooling-based schemes within the conventional patch-based CNN. Our experimental results showed that the aforementioned enhancements to the FCNN and patch-based CNN architectures resulted in superior performance in terms of classification accuracy compared to the corresponding base CNN architectures, i.e., *Dilation8* [33] in the case of the FCNN and *ResNet152* [10] in the case of the patch-based CNN.

In our future work, we plan to explore methods for exploiting a variable number of viewpoint images with no upper bound and without image repetition. We also plan to explore extensions of the logit pooling scheme that would discard low confidence predictions and defer to human judgment in such cases. Our long term goal is to develop an end-to-end trainable deep learning-based simultaneous localization and mapping (SLAM) scheme for semantic 3D reconstruction of coral reefs from image and video input. To that end, we also intend to explore ways of combining deep learning-based SLAM with Bayesian nonparametric hierarchical topic modeling schemes [9].

**Acknowledgment:** This research was funded in part by a Robotics Research Equipment Grant by the Faculty of Robotics and the Office of Vice President for Research, The University of Georgia, Athens, Georgia, to Dr. Bhandarkar and Dr. Hopkinson.

## References


[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.

[2] K. R. Anthony. Coral reefs under climate change and ocean acidification: Challenges and opportunities for management and policy. *Annual Review of Environment and Resources*, 41, 2016.

[3] O. Beijbom, P. J. Edmunds, D. I. Kline, B. G. Mitchell, and D. Kriegman. Automated annotation of coral reef survey images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, Rhode Island, 2012.

[4] O. Beijbom, P. J. Edmunds, C. Roelfsema, J. Smith, D. I. Kline, B. P. Neal, M. J. Dunlap, V. Moriarty, T.-Y. Fan, C.-J. Tan, S. Chan, T. Treibitz, A. Gamst, B. G. Mitchell, and D. Kriegman. Towards automated annotation of benthic survey images: Variability of human experts and operational modes of automation. *PloS One*, 10(7):e0130312, 2015.

[5] O. Beijbom, T. Treibitz, D. I. Kline, G. Eyal, A. Khen, B. Neal, Y. Loya, B. G. Mitchell, and D. Kriegman. Improving automated annotation of benthic survey images using wide-band fluorescence. *Scientific Reports*, 6(23166):1–11, 2016.

[6] B. J. Boom, J. He, S. Palazzo, P. X. Huang, C. Beyan, H.-M. Chou, F.-P. Lin, C. Spampinato, and R. B. Fisher. A research tool for long-term and continuous analysis of fish assemblage in coral-reefs using underwater camera footage. *Ecological Informatics*, 23:83–97, September 2014.

[7] Y. Boykov and G. Funka-Lea. Graph cuts and efficient n-d image segmentation. *International Journal of Computer Vision (IJCV)*, 70(2):109–131, 2006.

[8] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision (IJCV)*, 59(2):167–181, September 2004.

[9] G. Flaspohler, N. Roy, and Y. Girdhar. Feature discovery and visualization of robot mission data using convolutional autoencoders and bayesian nonparametric topic modeling. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1–8, 2017.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[11] J. D. Hedley, C. M. Roelfsema, I. Chollett, A. R. Harborne, S. F. Heron, S. Weeks, W. J. Skirving, A. E. Strong, C. M. Eakin, T. R. Christensen, et al. Remote sensing of coral reefs for monitoring and management: A review. *Remote Sensing*, 8(2):118, 2016.

[12] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008.

[13] O. Hoegh-Guldberg, P. J. Mumby, A. J. Hooten, R. S. Steneck, P. Greenfield, E. Gomez, C. D. Harvell, P. F. Sale, A. J. Edwards, K. Caldeira, et al. Coral reefs under rapid climate change and ocean acidification. *Science*, 318(5857):1737–1742, 2007.

[14] M. Johnson-Roberson, M. Bryson, A. Friedman, O. Pizarro, G. Troni, P. Ozog, and J. C. Henderson. High-resolution underwater robotic vision-based mapping and three-dimensional reconstruction for archaeology. *Journal of Field Robotics*, 34(4):625–643, 2017.

[15] A. King, S. M. Bhandarkar, and B. M. Hopkinson. A comparison of deep learning methods for semantic segmentation of coral reef survey images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1394–1402, 2018.

[16] G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[18] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, 2015.

[19] X. Liu, Y. Zhou, J. Zhao, R. Yao, B. Liu, and Y. Zheng. Siamese convolutional neural networks for remote sensing scene classification. *IEEE Geoscience and Remote Sensing Letters*, pages 1–5, February 2019.

[20] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[21] A. Mahmood, M. Bennamoun, S. An, F. A. Sohel, F. Boussaid, R. Hovey, G. A. Kendrick, and R. B. Fisher. Coral classification with hybrid feature representations. In *Proceedings of the IEEE International Conference on Image Processing*, pages 519–523, 2016.

[22] A. Mahmood, M. Bennamoun, S. An, F. A. Sohel, F. Boussaid, R. Hovey, G. A. Kendrick, and R. B. Fisher. Deep image representations for coral image classifications. *IEEE Journal of Oceanic Engineering*, 44(1):121–131, 2019.

[23] M. S. A. Marcos, M. Soriano, and C. Saloma. Classification of coral reef images from underwater video using neural networks. *Optics Express*, 13(22):8766–8771, 2005.

[24] O. Pizarro, P. Rigby, M. Johnson-Roberson, S. B. Williams, and J. Colquhoun. Towards image-based marine habitat classification. In *Proceedings of OCEANS*, pages 1–7, 2008.

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[26] A. Shihavuddin, N. Gracias, R. Garcia, J. Escartin, and R. B. Pedersen. Automated classification and thematic mapping of bacterial mats in the north sea. In *Proceedings of MTS/IEEE OCEANS-Bergen*, pages 1–8, 2013.

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[28] M. D. Stokes and G. B. Deane. Automated processing of coral reef benthic images. *Limnology and Oceanographic Methods*, 7(2):157–168, 2009.

[29] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.

[30] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, and the impact of residual connections on learning. In *Proceedings of AAAI Conference*, volume 4, page 12, 2017.

[31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[32] A. Telea. An image inpainting technique based on the fast marching method. *Journal of Graphics Tools*, 9(1):23–34, 2004.

[33] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.