Ecological Informatics 52 (2019) 103–121

Contents lists available at ScienceDirect

Ecological Informatics

journal homepage: www.elsevier.com/locate/ecolinf

Cascaded deep network systems with linked ensemble components for underwater fish detection in the wild

Alfonso B. Labao, Prospero C. Naval Jr*

Computer Vision and Machine Intelligence Group, Department of Computer Science, College of Engineering, University of the Philippines, Philippines

ARTICLE INFO

Keywords:
Fish detection in the wild
Deep learning applications to the environment

ABSTRACT

We propose a fish detection system based on deep network architectures to robustly detect and count fish objects under a variety of benthic background and illumination conditions. The algorithm consists of an ensemble of Region-based Convolutional Neural Networks that are linked in a cascade structure by Long Short-Term Memory networks. The proposed network is efficiently trained as all components are jointly trained by backpropagation. We train and test our system for a dataset of 18 videos taken in the wild. In our dataset, there are around 20 to 100 fish objects per frame with many fish objects having small pixel areas (less than 900 square pixels). From a series of experiments and ablation tests, the proposed system preserves detection accuracy despite multi-scale distortions, cropping and varying background environments. We present analysis that shows how object localization accuracy is increased by an automatic correction mechanism in the deep network's cascaded ensemble structure. The correction mechanism rectifies any errors in the predictions as information progresses through the network cascade. Our findings in this experiment regarding ensemble system architectures can be generalized to other object detection applications.

1. Introduction

Fish detection and counting are crucial tasks in marine science for temporal tracking of species, understanding of fish behaviour (Spampinato et al., 2014), aquaculture (Zion, 2012), among others. For fisheries management and policy formulation, keeping track of fish stocks and population is crucial to effectively control fish harvesting, promote breeding and prevent stock depletion (Walsh et al., 2004). For these reasons, the size of fish populations has to be accurately determined through surveys (Costa et al., 2006).

Traditionally, fish surveys are carried out by recording information of fish captured in traps, in nets by trawling, with lines, or through the use of piscicides. Capture-tag-recapture are also used for determining age, growth, movement and behaviour in reef fish populations. Non-capture techniques include underwater visual census by divers and hydroacoutic methods which are more accurate and non-destructive (Spampinato et al., 2014). However, diver observation of fishes may suffer from observational bias as many fish species instinctively evade human divers, swimming away from the survey area (Spampinato et al., 2010).

To address these drawbacks, the use of cameras which are non-invasive and are less conspicuous to fishes has been suggested (Katsanevakis et al., 2012). Camera-based monitoring also offers rapid and continual observation of fish species to keep up with rapid shifts in population distribution (Hollowed et al., 2013) (Mieszkowska et al., 2014). Some works proposed in situ monitoring programs using ROV vessels that are cost-effective (Siddiqui et al., 2017). However, these methods require manual offline annotation of collected video frames by fish experts. Manual annotation is very inefficient since classifying and annotating a minute of footage may take up to 15 min of a marine biologist's time (Spampinato et al., 2008). Given the number of frames that have to be processed, manual annotation require statistical sampling techniques to gather confident estimates of the fish population which could lead to possible sampling errors by novice annotators.

An attractive alternative is to use computer vision techniques to detect fish from videos or image stills and automate the counting process. This allows the use of camera set-ups for monitoring, as well as automated and efficient fish counting. However, this approach presents non-trivial difficulties. Automatic detection of fish objects in underwater videos need to deal with several challenges (Garcia et al., 2002; Labao and Naval, 2017; Negahdaripour and Yu, 1995). Underwater media produce light scattering effects, wavelength-dependent absorption, and lens/air/water interface image distortions. Suspended particles in water deflect photons from their straight line trajectories and introduce backscatter, termed “marine snow” (Horgan and Toal, 2009). Longer wavelengths of visible light are strongly absorbed by water

* Corresponding author.
E-mail address: pcnaval@dcs.upd.edu.ph (P.C. Naval).

https://doi.org/10.1016/j.ecoinf.2019.05.004
Received 2 December 2018; Received in revised form 4 May 2019; Accepted 6 May 2019
Available online 09 May 2019
1574-9541/

resulting in varying fish colors relative to camera distance and depth. These factors confuse classical detection algorithms that are not designed to handle such difficulties. This is compounded by the fact that large numbers of fish, from 20 to 100 individuals per frame, have to be detected.

Some early methods that attempted to perform automatic fish detection often relied on background subtraction methods (Garcia et al., 2002). This approach gathers motion information using pixel-wise subtraction of consecutive image frames to segment and localize fish objects from static backgrounds. However, these approaches are limited by their dependence on fixed camera setups, static backgrounds, non-varying illumination conditions and on the assumption that the fishes are in motion. The latter condition may not be true for some fish species. Furthermore, the presence of underwater media problems mentioned above pose serious difficulties to background subtraction algorithms.

Recent advances in computer vision and machine learning provide methods that can potentially address the challenges presented by underwater media. Most of these techniques are based on deep learning algorithms that address the limitations of previous algorithms which rely on motion information or manually crafted features. Deep learning methods automatically generate features using convolution and other operations (Krizhevsky et al., 2012; LeCun et al., 1988). The most popular deep learning method for computer vision tasks is the Convolutional Neural Network (CNN) whose variants have been successfully applied to numerous image classification tasks (Karpathy et al., 2014; Krizhevsky et al., 2012; Simonyan and Zisserman, 2014).

Deep learning methods in computer vision have also progressed to deal with localization tasks. Several recent localization techniques utilize variations of the base CNN to predict bounding box coordinates of objects. One of the first deep learning localization network is the Region-based Convolutional Neural Network (R-CNN) which uses a selective search procedure generate object proposals (Girshick et al., 2016). A further improvement in localization networks is the Faster R-CNN which automates the proposal generation process itself using a Region Proposal Network (RPN) (Ren et al., 2017). Faster R-CNN served as a base network architecture for several other localization models (Dai et al., 2015; Li et al., 2016), and the G-RMI network (Fathi et al., 2019). The G-RMI network is notable since it uses an ensemble architecture whereby predictions of several networks are combined to increase accuracy performance. However, we note that several of these detection works use standard datasets (Everingham et al., 2010), and have not been applied to image data taken in the wild, which is the objective of this paper.

For the fish detection task, some prior works have implemented deep learning based systems (Labao and Naval, 2017; Li et al., 2015; Villon et al., 2016; Zhuang et al., 2017). In particular, Villon et al. (Villon et al., 2016) found that deep learning models outperform classical machine learning techniques that rely on manually crafted features. Their experiments were performed on the SEACLEF database which consists of 20 to 30 fish objects per frame. To differentiate these experiments from our paper, we note that the dataset which we use to train our models is more challenging and reflects more closely the actual number of fish objects at benthic depths. The dataset consists of 18 underwater videos, separated into 10 training videos and 8 test videos. In addition, fish objects are more dense, numbering 20 to 100 fish objects per frame, with small sizes of less than 900 square pixel areas. This adds up to a total of close to 10,000 fish objects that have to be detected by the algorithm, the majority of which are small. We also explicity set the number of fish objects in the training data set to be less than the number of fish objects in the test data set. This is to test the capacity of algorithms to generalize well over harder environments.

Given the challenges presented by the dataset, standard deep learning localization models may not be able to perform well and some enhancements to the base network architecture are needed. Hence, this paper proposes a deep learning architecture that adopts an ensemble structure whose components are detector networks (Ren et al., 2017). G-RMI implements the traditional type of ensemble by combining the outputs of several independent networks. Our proposed ensemble uses a special structure where the ensemble components are arranged in a cascade. The cascade components are not independent since they have connections in the form of Long Short-Term Memory (LSTM) links (Hochreiter and Schmidhuber, 1997). Moreover, the flow of information from one cascade to the next provides an automatic correction mechanism that increases accuracy. The proposed model has other benefits, such as (1) cascade components that are jointly trained in a single backpropagation pass and (2) LSTM links that process information with an attention mechanism which confers robustness against image distortions.

To assess the performance of our model, we compare our approach to a second ensemble system similar to traditional ensembles (Fathi et al., 2019) and to a strong baseline model composed of a single Faster R-CNN network as applied to the SEACLEF database (Zhuang et al., 2017). Experiments consist of a series of tests: (test 1) prediction on unseen fish objects found in the training frames, (test 2) prediction on new frames, (test 3) prediction on fish objects with multi-scale distortions and cropping, and (test 4) ablation tests. Experimental results show that for test 1, all networks performed similarly. For tests 2 and 3, the proposed cascaded ensemble outperformed other systems. We conjecture that the cascaded structure of our proposed system benefits from the automatic correction mechanism where cascades repeatedly refine initial proposals. In addition, the attention mechanism in System 1's LSTM links makes it more robust against scale distortions. This is verified in test 4, where LSTM links with attention mechanism significantly improve multi-scale inference.

For future work, our proposed cascade ensemble structure could be generalized to include other components aside from Faster R-CNN, as well as detect objects other than fish. In summary, our paper has these contributions:

Summary of Contributions:

- a localization network that adopts a cascaded ensemble structure, where components are linked by an LSTM network. For efficiency, all network components are trained in a single backpropagation pass
- an automatic correction mechanism under the cascade structure to lower prediction errors, along with an attention mechanism in recurrent network links for more robust predictions against image distortions
- a new dataset of 18 underwater video sequences of varying illumination conditions and backgrounds. Close to 88% of the fish objects in the test set have small object sizes of less than 900 square pixels, and where training videos have less fish objects than test videos to test generalization capacities of models
- performance comparisons of cascade ensemble with traditional ensemble systems and a strong baseline single object detector, under 4 tests with multi-crop distortions, cropping, and ablation.
- experiments that show better performance for the cascade ensemble. The benefits of the automatic correction mechanism and attention is demonstrated along with analysis.

2. Deep learning neural networks

This section presents some concepts on deep learning-based detection networks using Faster R-CNN for its base architecture. Briefly, a deep network is simply a neural network with several layers, thereby providing it with depth (Goodfellow et al., 2016). Such a neural network can automatically extract informative features that are appropriate for its given task (Goodfellow et al., 2016). Early neural networks are unable to increase their depth significantly due to vanishing gradients which are circumvented by deep networks through the use of an non-squashing activation function such as the Rectified Linear Unit (ReLU) activation function (LeCun et al., 2015). Furthermore, progress

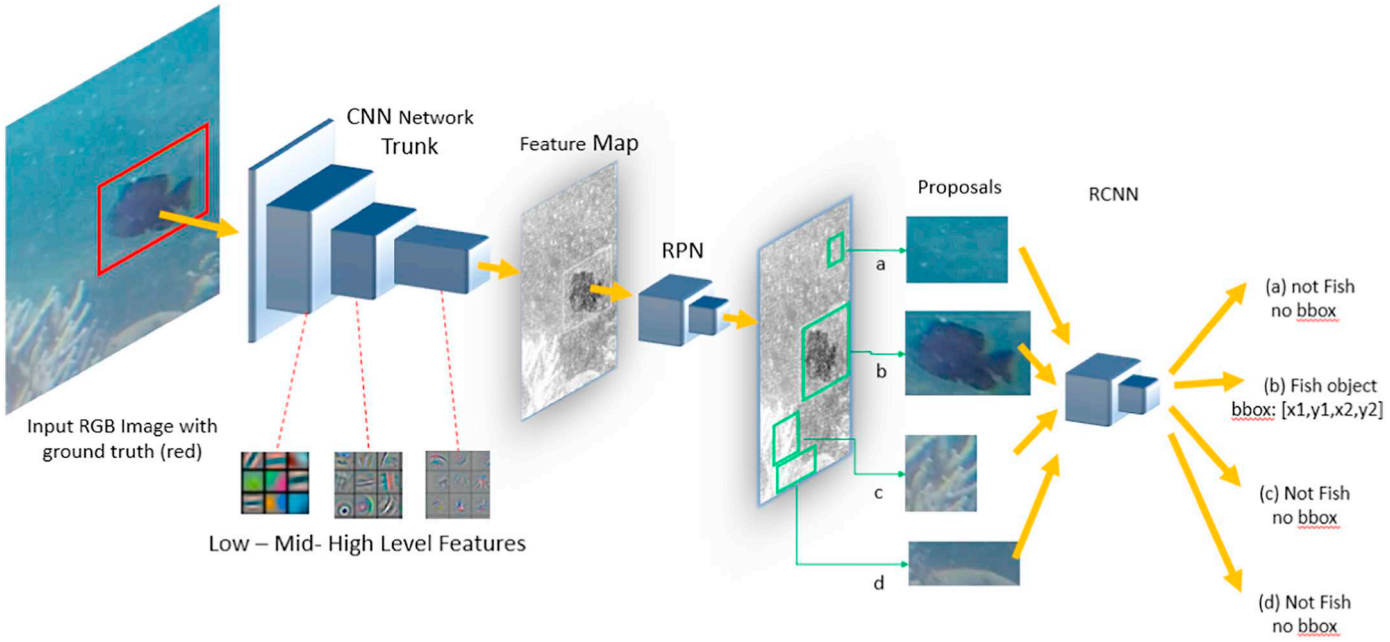

Fig. 1. Flow of the basic Faster R-CNN detection model.

in loss functions (Goodfellow et al., 2016) (i.e, the function that measures the amount of error in a network's prediction from ground truth) enabled more well behaved supervised training of deep networks.

Early deep networks were classification networks (Krizhevsky et al., 2012), but localization networks were proposed shortly after (Ren et al., 2015). Since this paper concentrates on localization networks, we show in Fig. 1 the general flow of Faster R-CNN (Ren et al., 2015) which is considered the standard deep learning localization network. The Faster R-CNN model has three main parts (1) a CNN network trunk, (2) a proposal generator RPN network, and (3) the R-CNN region classification network.

The CNN network trunk is the first component in Faster R-CNN. It receives an input RGB image of arbitrary size and generates a feature map containing highly informative features describing the input image. These features will be used to predict possible locations of objects (a detection task), as well as to predict their objectness probability (i.e. whether they correctly represent an object or not, which is a classification task). Given the feature map, detection is done using a proposal generation process carried out by the Region Proposal Network (RPN). The classification task is handled by the R-CNN which uses the same feature map to predict objectness probabilities of the proposals. The CNN, RPN and R-CNN are jointly trained during backpropagation thereby increasing training efficiency significantly.

2.1. Convolutional neural networks (CNN)

The Convolutional Neural Network (CNN) provides the feature generation component used by most deep networks designed for image analysis tasks (Krizhevsky et al., 2012). The trunk of a CNN is formed by a series of convolutional filters that are convolved over the input image to generate a stack of feature maps, with each feature map containing a set of special features characterizing the image. At the front end of the CNN trunk are usually found filters that detect low-level features which represent edges and color patterns. These are followed by increasingly sophisticated features, i.e. shapes and contours in later parts of the trunk. The final output of the CNN trunk is a highly informative feature map that will now serve as input for the Region Proposal Network and R-CNN components of our detection network. For our proposed network, the trunk adopts a residual network structure from (He et al., 2016).

In the case of underwater fish localization, the CNN's capacity to automatically generate features from arbitrary inputs is useful. It circumvents the need for features that rely solely on motion or brightness levels, i.e. by relying also on features that depend on edges, shapes, or contours that are brightness-independent. This renders CNN models robust to illumination changes as a result of changing depth and movement in surface waters. In addition, since the input consists of an RGB image, the CNN is also able to automatically generate features that use color information to differentiate fish objects from their background. This is shown in Fig. 1, where the features in the network trunk (from low level to high level) combine edge information along with color information.

2.2. Region proposal network for detection (RPN)

Early deep learning classification networks classify a single object found within the input image. This changes in the case of detection tasks, since an input image can contain several objects. To handle detection tasks, one of the methods used is proposal generation. Proposals are data structures that contain information on the locations of possible objects in the input image. For Faster R-CNN, a proposal is a 4-element tuple consisting of coordinates that represent corners of bounding boxes.

Faster R-CNN automates the production of proposals using a Region Proposal Network (RPN). We can view the RPN as another CNN with its own set of convolutional filters. The filters of the RPN operate over the input feature map from the trunk and predict a tuple consisting of objectness probabilities and proposal coordinates of boxes fixed at certain locations dispersed across the image. These locations, called “anchors”, form the points of a grid that span the entire input image. These anchors are spaced in intervals of 9 or 12 pixels and each anchor is assigned a set of ‘anchor boxes’ as shown in Fig. 2. Using its filters, the RPN predicts for each anchor box its objectness probability and regressed bounding box coordinates. Anchor boxes with high objectness probabilities are stored as proposal candidates and serve as inputs for the R-CNN since they are more likely to contain objects (Ren et al., 2015).

We note that the standard Faster R-CNN implemented a single RPN since one RPN is sufficient to detect relatively larger objects in the PASCAL VOC dataset. However, for our proposed network, we increase

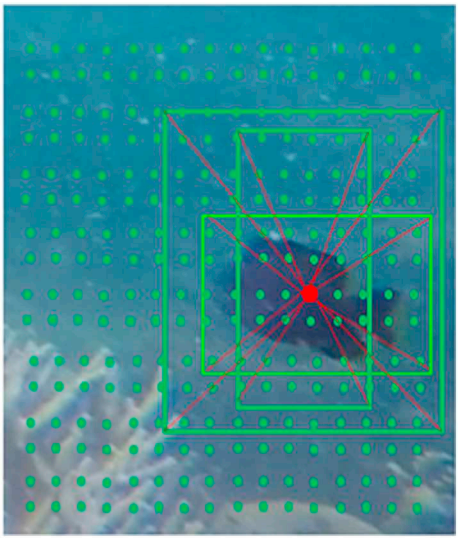

Fig. 2. A graphical illustration of the anchor grid over an input image used by the RPN. The anchors (green dots) are spaced evenly across the input image. Each anchor is assigned a set of 'anchor boxes' of varying aspect ratios and size, i.e. there are three anchor boxes for one anchor (red dot). For each anchor box (of each anchor), the RPN predicts its objectness probability and regresses the actual coordinates (bounding box corners) of a potential object - given the sub-image enclosed by the anchor box. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the number of RPNs to two, with each RPN having a specialized set of anchor boxes that can accommodate both small and large fish sizes.

2.3. Region classification network (R-CNN)

The R-CNN is a sub-network CNN that operates on 'smaller' feature map inputs, as shown in Fig. 1. It predicts the objectness probability and actual coordinates of each object captured by a proposal from the RPN. The R-CNN is actually a refinement network that corrects bounding box coordinates and objectness probabilities of each proposal. For this paper, our main contribution lies in modifying the R-CNN to a cascaded ensemble for improved localization accuracy.

2.4. Ensembles

In machine learning, a common practice to improve performance is to use ensembles that combines outputs of several models (Krogh and Vedelsby, 1995; Lee et al., 2019; Wang et al., 2019). The theory behind ensembles is that they resemble a form of 'bagging' to reduce the variance of predictors (Hastie et al., 2013). With lower variance, predictors generalize better on test sets that were not encountered during training. Mathematically, let σ_i^2 represent the variance of a ensemble component i given prediction $f(x)_i$ under input x. Suppose that the variances of component i and j, $i \neq j$ are not correlated, i.e. $\mathbb{E}[\sigma_i^2 \sigma_j^2] = 0$. The most basic form of ensemble involves averaging over the predictions of K components. The resulting variance can be expressed as:

$$Var[f(x)] = Var\left[\frac{1}{K}\sum_{i}^{K}\left(\sigma_i^2 + \sum_{j\neq i}\sigma_i^2\sigma_j^2\right)\right] = \frac{K\sigma^2}{K^2} = \frac{\sigma^2}{K}$$

Thus, having K components in an ensemble reduce $f(x)$'s variance by $1/K$ assuming that components are not correlated.

In this paper, we use cascade ensemble instead of simple averaging. This ensemble is implemented using a recurrent neural network (LSTM) to reduce both variance and bias, where bias is attenuated by an automatic correction mechanism. The Appendix provides a mathematical model to show bias reduction properties of cascades.

2.5. Long short-term memory recurrent neural network (LSTM)

Recurrent neural networks (RNN) are applied to sequence data where information has temporal dependencies. RNNs re-use prior predictions to capture context, where the simplest RNN re-uses its own prediction from prior sequence stesps. However, RNN's are prone to vanishing gradients or gradient explosions across long time periods. But one type of RNN, the Long Short-Term Memory (LSTM) Network (Gers et al., 2019) addresses the vanishing gradient problem through gates and bypass connections. In this paper, we use the LSTM to link the R-CNN cascade. The LSTM's capacity to retain information over long sequences, allows it to propagate gradients effectively across 7 cascade components. To verify the effect of LSTM links, we implement several ablation tests (test 4) in our experiments.

3. Methodology

In this section, we provide implementation details for our proposed algorithm. We term our proposed algorithm as System 1. For comparison, we implement 3 systems, termed Systems 1, 2, and 3. System 2 uses traditional ensembles while System 3 uses basic Faster R-CNN. For this section, we only provide details on System 1, and details on Systems 2 and 3 are in the Appendix.

- System 1: Multi-cascade object detection network with 2 RPNs and an ensemble of 7 CNN components linked by sequential LSTMs (jointly trained)
- System 2: Ensemble formed from 3 object detection networks trained separately: each with 1 RPN and 2 cascade components
- System 3 (Faster-R-CNN Baseline System): Single object detection network with 2 RPNs and 2 cascade components

The three systems receive a single RGB video frame as input. The input frame's dimensions ($H \times W \times 3$) can vary in height (H) and width (W) but are fixed at the 3 RGB channels. For output, the three systems provide a set of box coordinates for each detected fish object. For System 1, we enumerate its features as follows:

3.1. System 1 architecture features

- single 50-layer Residual Network trunk
- two RPN systems, to generate proposals that accommodate both small and large fish objects
- a R-CNN network composed of an ensemble of 7 CNN units arranged in a cascade
- seven CNN units are linked by an LSTM recurrent neural network
- LSTM link processes information using an attention mechanism
- all network components are trained jointly in a single back-propagation pass using compound loss functions

With the cascade ensemble, outputs of earlier cascades are re-used in future cascades - which allows predictions to be refined. In addition, the LSTM's attention mechanism processes information by sections, giving it robustness properties. We note that System 1 differs from existing detection models in the R-CNN part. Instead of a single CNN (Ren et al., 2015), we use a cascaded ensemble. The base model for System 1 is the multi-network 2-cascade network (MNC) of (Dai et al., 2015), but MNC does not implement a linking mechanism to propagate information from one cascade to the next.

3.2. 50-layer CNN residual network trunk

System 1 (and also Systems 2 and 3) adopts a 50-layer residual network structure from (He et al., 2016). Residual networks use skip layers to allow information to freely pass across the network. The units of residual networks are termed as "residual blocks" where each block

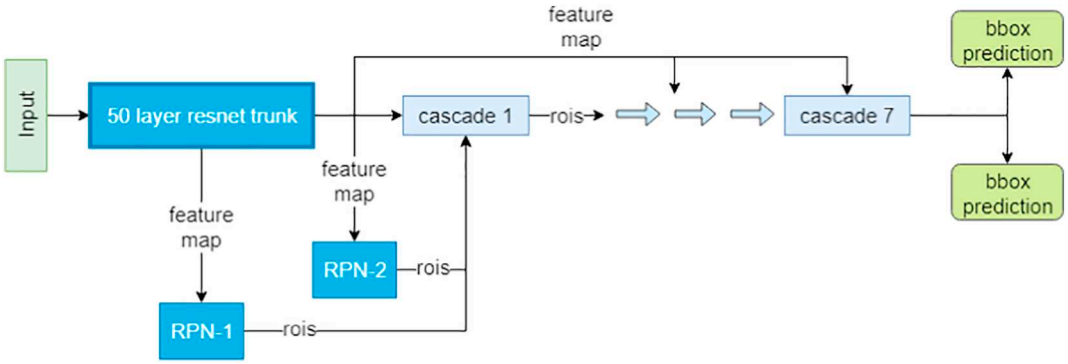

Fig. 3. System 1 Network Structure - a single 50-layer residual trunk provides shareable features for 2 Region Proposal Networks and an R-CNN composed of 7 cascade components. The feature map from the 50-layer residual network trunk is the input for RPN-1 and RPN-2 as well as for each of the cascades. The vertical connections over the cascades represent the sharing of the feature map to each of the cascade components. Hence, each cascade components re-uses the feature map from the main CNN trunk.

is composed of a series of 1 × 1 - 3 × 3 - 1 × 1 convolutions with skip addition and batch normalization layers (He et al., 2016). The trunk receives an RGB image of arbitrary size as input and outputs a shareable feature map of $H/16 \times W/16 \times 1024$ dimensions. For System 1, we use a single 50-layer residual network trunk, where its shareable feature map output is used for both RPN and R-CNN.

3.3. Region proposal network (RPN) for proposal generation

System 1 adopts a dual RPN structure to cater to both small and large fish objects (Fig. 3). RPN-1 is connected to the middle of the network trunk and receives a feature map of size $H/8 \times W/8$, while RPN-2 is directly connected to the shareable feature map at the end of the network trunk of size $H/16 \times W/16$. Each anchor in the RPN is assigned a set of twelve anchor boxes where pixel dimensions of the 12 anchor boxes are computed according to *base_size* × *scale* × *aspect_ratio*. For System 1, anchors in RPN-1 are separated by strides of 8 pixels with anchor boxes that have a *base_size* of 3. For RPN-2, anchors are separated by strides of 16 pixels, its anchor boxes have a *base_size* of 5. For all RPNs across systems, the vector *scale* is [8,16,32,64], while the vector *aspect_ratio* is [1 : 1, 1 : 2, 2 : 1]. Multiplying *base_size* with *scale* and *aspect_ratio* creates a total of 12 anchor boxes for each anchor location.

During training, for each anchor in the RPN, the proposal with largest Intersection over Union (IoU) over a ground truth box, or with an IoU of above 70% is assigned as foreground (class 1). The rest are assigned as background (class 0). These labels train the RPN to predict objectness probabilities of each anchor. The RPN also performs bounding-box regression of coordinate adjustments $[d_x, d_y, d_w, d_h]$, which adjust the reference anchor box to approximate the actual box coordinates $[x_1, y_1, x_2, y_2]$ of a fish object. RPNs are trained using a compound loss function. Formally, let $F(\Theta)$ denote the shareable feature map of size of $H/16 \times W/16$, where Θ denotes all network parameters. For System 1, let $F^8(\Theta)$ denote the feature map branching from the middle of the trunk with a size of $H/8 \times W/8$. The RPN passes a 3 × 3 convolution over $F(\Theta)$, followed by two 1 × 1 convolutions to produce $p_i{}^a(\Theta)$ and $t_i{}^a(\Theta)$, where i denotes an anchor location. The RPN's compound loss function L^{rpn} is

$$\begin{aligned} L^{rpn} &= l_{cls}(p_i^{anchor}(\Theta)) + l_{reg}(t_i^{anchor}(\Theta)) \\ &= l_{cls}^8(p_i^{anchor}(\Theta)) + l_{reg}^8(t_i^{anchor}(\Theta)) \end{aligned} \quad (1)$$

where $p_i^{anchor}(\Theta)$ is an 24 dimensional vector of object probabilities for anchor i's 12 scale and aspect ratios. The quantity $t_i^{anchor}(\Theta)$ is a 48-d vector of bounding box parameters, l_{cls} is softmax, while l_{reg} is the smoothL1 loss function. Vectors $l_{cls}{}^8$ and $l_{reg}{}^8$ refer to anchors taken at $F^8(\Theta)$, corresponding to RPN-1 in systems 1 and 2. From bounding-box regression, RPN feeds to R-CNN a set of proposals $B_{0,i}$ representing box coordinates $[x_1, y_1, x_2, y_2]$. During training, 256 proposals are processed, where 50% have foreground labels and the other 50% have background labels. During inference, the top 4000 proposals with the highest predicted objectness probabilities are fed to the R-CNN.

3.4. Multi-cascade R-CNN with an ensemble of 7 components and LSTM links

System 1's R-CNN is a 7-cascade ensemble as shown in Fig. 4. Each component j extracts sub-feature maps from the set of proposals generated by the previous cascade $j-1$. Sub-feature maps are extracted from $F(\Theta)$ through RoI Cropping which uses bilinear interpolation to fix all maps to size 6 × 6. Bilinear interpolation has nice differentiable properties that allows backpropagation of gradient corrections to be passed across cascades during training. Extracted sub-feature maps are passed on to the CNN unit assigned to cascade j. Each CNN unit in a cascade j is composed of three residual blocks. Residual block weights for each of the 7 cascade components are independent, resulting in an ensemble structure. (See Fig. 5.)

The three residual blocks in the CNN unit of a cascade outputs a 3 × 3 feature map with a depth of 2048 dimensions. This 3 × 3 × 2048 feature map is averaged across the dimensions resulting in a 3 × 3 map. The averaged map is fed to a sequential LSTM unit which reshapes the sub-feature map to a flattened vector of dimensions 1 × 9. This way, each element in the flattened vector corresponds to an element in the LSTM sequence. By processing each element in the flattened vector separately, the LSTM operates according to attention mechanism. Here, each block in the sequence corresponds to a coordinate in the 3 × 3 feature map, from the top-left coordinate down to the bottom-right coordinate. Attention mechanism schemes improve network accuracy since it can extract key features from portions of the object (Hara et al., 2017). The LSTM unit receives its prior 2048 dimensional state vector from the previous cascade and, after processing the sequence, outputs a 2048 dimensional hidden unit vector that serves as input to a fully connected layer. The current 2048 state vector in cascade j is passed to the next cascade $j+1$. The fully connected layer form the final stage of each cascade component j. It predicts proposal coordinate adjustments and objectness probabilities for each RoI proposal i. A state-bridge layer (Dai et al., 2015) transforms the reference proposal i to a new proposal given the predicted coordinate adjustments. In summary, the detailed steps in the R-CNN for System 1 are as follows:

System 1: procedures for cascades $j = 1$ to 7

Step 1: Receive RoI i proposal coordinates B_{j-1} from the previous cascade $j-1$. In the case of the first cascade $j = 1$, RoI proposal coordinates are from the RPN;

Step 2: Use RoI i proposal coordinates in the form $[x_1, y_1, x_2, y_2]$ to extract sub-feature maps from the shareable feature map $F(\Theta)$ at the

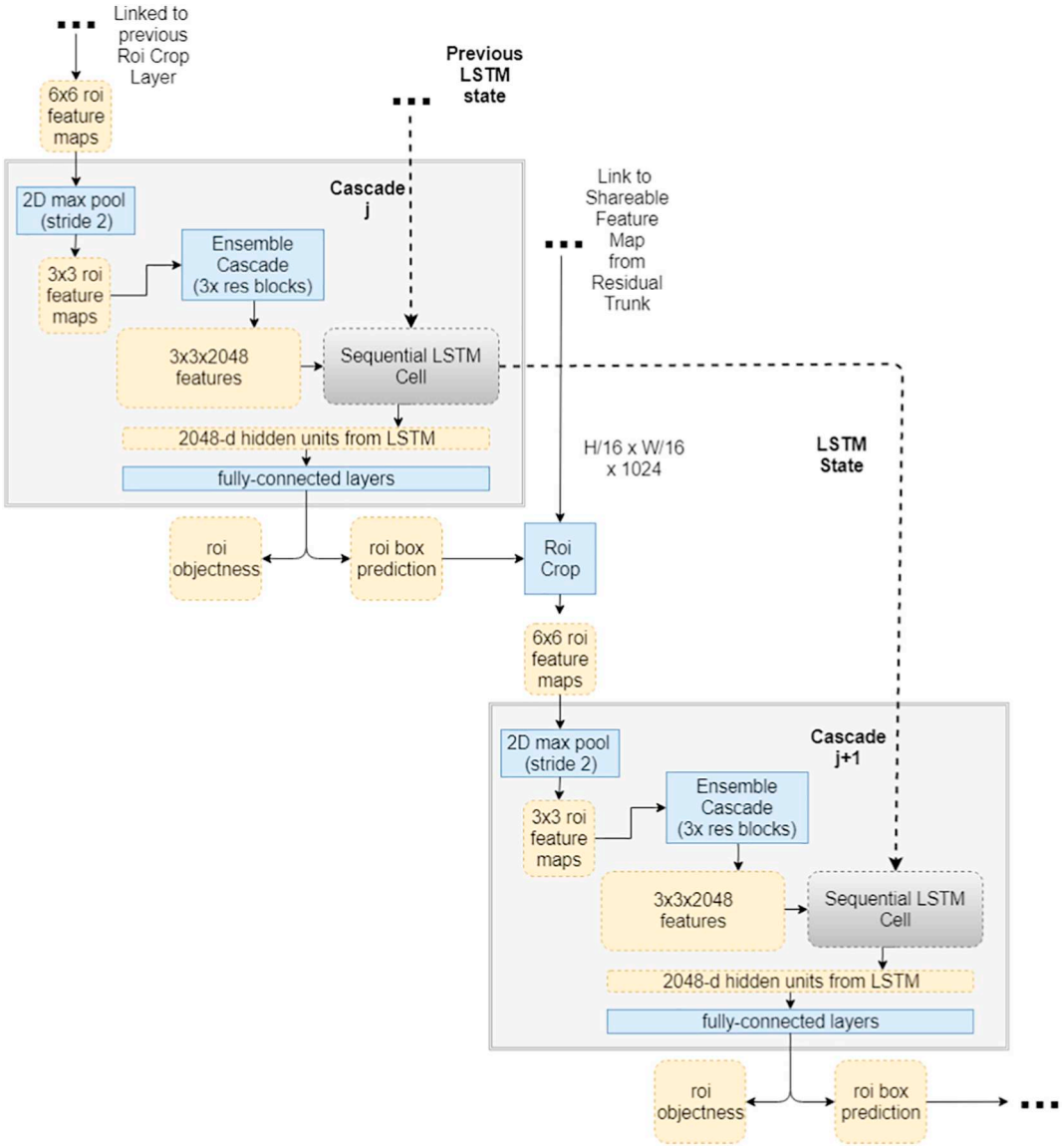

Fig. 4. The structure of two interconnected cascade components in the R-CNN. In System 1, cascade components are extended to 7 cascades, while for systems 2 and 3, the R-CNN has only two cascades.

end of the network trunk using RoI-cropping. Sub-feature maps are resized to a uniform $6 \times 6 \times 1024$;

Step 3: Pass the $6 \times 6 \times 1024$ sub-feature map to a series of convolutional layers (three residual blocks). The sub-feature map is downsized to 3×3 at a depth dimension $D = 2048$;

Step 4: Pass the 3×3 sub-feature map to an LSTM cell, where the sub-feature map is resized to sequential-form: $1 \times 9 \times 2048$) and the LSTM sequentially processes each block. In this model, the LSTM sequence has 9 blocks as shown in 4. In each cascade, the LSTM re-uses the hidden states S_{j-1} computed from the previous cascade (except for the first cascade $j = 1$). The 2048 dimensional hidden units are passed to the fully connected layer;

Step 5: Using the previous step's inputs, the fully connected layer predicts bounding box coordinate adjustments $p(i,j)$ and objectness probabilities $t(i,j)$. Coordinate adjustments are in the form $[d_x, d_y, d_w, d_h]$ which refer to adjustments with respect to the reference box center x and y coordinates and the box height h and width w;

Step 6: Predicted bounding box coordinate adjustments are processed according to a state-bridge layer (following (Dai et al., 2015)) which transforms the reference RoI B_{j-1} to a new set of RoIs B_j of the form $[x_1, y_1, x_2, y_2]$ using the predicted $[d_x, d_y, d_w, d_h]$;

Step 7: The current hidden state S_j and the new set of RoIs B_j are passed to the next cascade $j + 1$ which begins again at step 1.

The R-CNN in System 1 is trained using a compound loss function for each proposal RoI i, and for each cascade $j = 1, \ldots, 7$ following Eq. (2). Loss is averaged over RoIs i.

$$L_{j,i} = l_{cls}(p^{cls}_{j,i}(\Theta)) + l_{reg}(t_{j,i}(\Theta)) \tag{2}$$

where l_{cls} is a softmax loss function while l_{reg} is the smoothL1 loss function. The quantity $p_{j,\,i}$ refers to cascade j's prediction of objectness probability for proposal RoI i, while $t_{j,\,i}$ refers to cascade j's prediction of coordinate adjustments for RoI i. The end-output of the R-CNN is $p_{j=7,\,i}$ and $B_{j=7}$ referring to the final predicted probabilities of an RoI i and the final box coordinates.

3.5. Total loss and training details

The total loss of the network combines the compound losses of the RPN and of each cascade in the ensemble R-CNN:

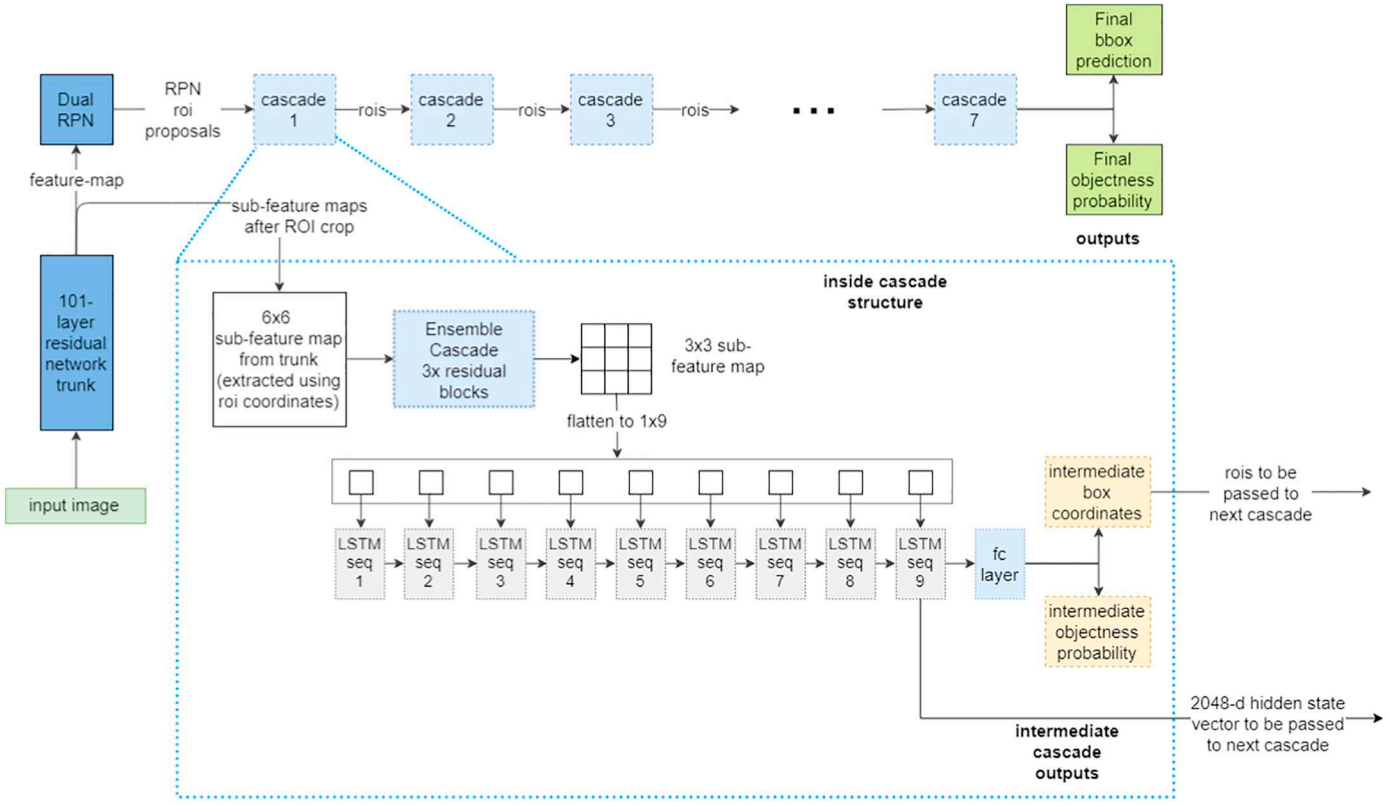

Fig. 5. System 1 R-CNN with Sequential LSTM Structure. The cascaded network in System 1 has 7 CNN ensembled components that are linked together through a sequential LSTM unit. The LSTM unit performs an attention mechanism over each CNN cascade output, by reshaping the 3 × 3 CNN output tensor to 1 × 9 and treating each element in the 9-dimensional flattened tensor as a part of a sequence.

$$L^{total} = L^{rpn} + \sum_{j=1}^{7} \left(\frac{1}{R} L_{j,i} \right) \tag{3}$$

For all three systems, network training is performed end-to-end for 200 epochs (with 300 training frames for each epoch). We use a weight decay parameter of 0.0001 and a Nesterov momentum value of 0.9. Learning rate is set at 0.001 which is divided by 10 after 150 epochs. No image pre-processing is performed other than subtraction from mean image. Network is initialized with ImageNet weights from (He et al., 2016) as done elsewhere (Dai et al., 2015; Ren et al., 2017). In a forward pass, 4 K proposals per RPN are processed, and the top proposals for objectness are retained after Non-Maximum Suppression (NMS) with 0.3 threshold (pre R-CNN). Final predicted proposals (post R-CNN) are NMS-suppressed with 0.1 threshold.

4. Training and test data description

In this section, we present (1) statistics of our data, (2) performance metrics and four test schemes, and (3) experimental results for each of the 4 test schemes. In (3), we insert some qualitative analysis on the experimental results, and we provide a mathematical treatment of the analysis in the Appendix for reference.

4.1. Statistics on training and test set

Our training data consists of ten (10) underwater video sequences for a total of 300 training frames, with more than 10,000 fish objects. The videos were obtained at depths ranging from 7 to 24 m, taken from a custom-made stereo rig composed of three (3) GoPro cameras. The video frames have a wide variety of backgrounds and most contain large numbers of fish objects different species. In general, the training and test data in this experiment is harder than the benchmark PASCAL VOC dataset (Everingham et al., 2010), for the following reasons:

- uneven and changing illumination conditions, water backscattering effects, presence of marine snow, fish motion, and similar appearance of fish objects with coral background, etc.
- around half of the objects are small objects having areas smaller than 900 square pixels and with deformable shapes
- much larger quantity of objects per frame to be detected in the test set compared to training set, ranging from 20 to more than 100

For the test data, we gathered eight (8) videos with some differences in background environments from the training data. For each video, we randomly sample 3 frames for manual labeling, amounting to 27 frames with more than 2000 fish objects to be detected in total. We describe the datasets in Table 1, where for index notation, we append each train and test video with a 'J' at the beginning. The fish objects in the training data were manually annotated by a marine science researcher for expert verification.

In general, the number of fish objects in the 8 test videos are larger than those in the training videos. This presents a unique challenge for localization systems since they have to generalize over a more difficult test set. However, this suits the purpose of this experiment, which is to test the capacity of models to generalize over new environments. Table 2 shows different fish object sizes that can be found among the 8 test videos. As can be seen, roughly 88% have object sizes that fall between 100 and 2500 square pixels. In the COCO dataset (Lin et al., 2014) these sizes fall under the 'small' object size, and are among the harder-to-localize objects.

In terms of background, we include in Table 1 the rough proportion of water column areas against seabed and coral areas. We also include additional information on the illumination conditions of the video and on background objects, i.e. rocks/corals/particles.

Table 1
Training Data Statistics for the 18 videos (10 for training, 8 for test).

Training Video	Average Number of Fish Objects	% Water Column	Specs	Illumination	Test Video	Average Number of Fish Objects	% Water Column	Specs	Illumination
J01	6	95%	Particles/sand	clear	J103	74	60%	Rocks/corals	Dark
J06	33	60%	Rocks/corals	clear	J105	146	85%	Seabed/corals	Dark
J07	68	85%	Particles/ bubbles	clear	J115	88	70%	Rocks/corals	Clear
J08	16	40%	Rocks/sand	clear	J119	31	55%	Rocks/seabed	Dark
J09	30	60%	Rocks/corals	clear	J121	97	65%	Rocks/corals	Dark
J49	15	50%	Rocks/corals	blurred	J239	31	70%	Particles/ corals	Clear
J58	18	60%	Debris/corals	blurred	J243	40	60%	Rocks/corals	Clear
J59	7	70%	Rocks/corals	blurred	J255	20	40%	Rocks/corals	A bit blurred
J70	11	50%	Seabed/corals	dark					
J75	14	45%	Rocks/corals	dark					

Table 2
Fish Size Statistics for the test set: summed across all 8 test videos.

Fish Size x (in square pixels)	Number of Objects
$x < 100$	5
$100 \leq x < 900$	929
$900 \leq x < 2{,}500$	478
$2{,}500 \leq x < 10{,}000$	168
$x \geq 10{,}000$	6

5. Performance metrics and test schemes

5.1. Performance metrics for the specific task of the experiment

The performance of each system is measured in terms of how well each predicts the bounding box coordinates of fish objects per frame, agnostic to which type of species they belong. This is similar to a foreground/background localization problem where objects of interest are classified as foreground, while all other pixels and objects are classified as background. We measure localization performance in terms of intersection over union or IoU. We use IoU precision/recall/F-Score performance metrics described in PASCAL VOC where true positive (TP) boxes have ≥ 0.50 IoU threshold. If a predicted RoI has ≥ 0.50 IoU with a ground truth box, it is counted as TP, otherwise it is a false positive (FP). In the case of multiple overlapping boxes over a single ground truth object, only one predicted box is counted as TP, while the rest are FP. This is a penalty scheme for multiple boxes over a single ground truth object since only one box should remain for each ground truth object during inference.

5.2. Test schemes

To assess the performance of each system given our new dataset, we implement four different test schemes as follows:

1. Test 1: Localize unseen fish objects in videos that were used for training.
2. Test 2: Localize fish objects in entirely new video frames from a test set (independent of training set) using single crop inference
3. Test 3: Localize fish objects in the same set as test 2 - using multi-scale inference with multi-crop
4. Test 4: Ablation tests for System 1 where LSTM components are removed

Test Set 1 is meant to check the learning capacities of each system and assess if all are able to learn given familiar training sets. This test consists of 60 frames. This test makes sure that all systems are able to learn equally well. Test Set 2 is meant to check the generalization capacities of each system and assess if they are able to perform well given 8 independent test videos with entirely new environments and illumination conditions. In a way, test 2 is more crucial for assessing a system's performance than test 1 since its frames are independent from the training set. Test Set 3 builds upon Test Set 2 and uses multi-crop and multi-scale inference to check if the system can still localize despite additional distortions and removal of global information. Test Set 4 is an ablation test to check if the LSTM sequential links effectively propagates information across ensemble components.

6. Overview of system performance relative to background

We show in Figs. 6 and 7 sample detections of System 1 algorithm. In these figures, green boxes denote bounding box predictions outputted by the algorithm, while red boxes denote sample false negatives (i.e. missed detections) of the algorithm. From the figures, System 1 is able to localize fish in both the water column and rock/coral background. However, it reported a lot of false negatives for very small fish objects. This is shown in Fig. 6, where several very small objects are missed by the detector. These objects however are very far from the camera location and are hardly discernable even by an untrained human observer. Comparing detection accuracy in the water column background against a rock/coral background, System 1 performs better with a water column background. As seen in Fig. 7, it is able to detect several fish objects on top of the coral/rock bed. However, several fish objects that lie near the bottom against a rock background are missed out.

7. Analysis and comparison of system performance

7.1. Test set 1 systems performance: Prediction on training environments

From Table 3, all three systems perform comparably well for frames taken from training video sequences. Both precision and recall for all three systems are at an average of 60+. This indicates that all three systems are able to learn. Hence, given familiar frames, ensemble structures do not provide much benefit. As seen in Table 3, System 3 with its non-ensemble single network structure has an F-Score that is very close to the ensemble based structures of systems 1 and 2.

7.2. Test set 2 systems performance: prediction on new environments

For test set 2, Table 4 shows that System 1 has the best Precision, Recall and average weighted F-Score values. We conjecture that this is due to the correction mechanism of a cascade ensemble. (to be explained in more detail below). Both ensemble-based Systems 1 and 2 outperform the Faster R-CNN baseline model. This is expected since ensemble models have better generalization than single models. Comparing the multi-cascade ensemble network in System 1 with the separated network ensemble in System 2, System 1 performs better when

Fig. 6. Sample fish object detections taken from the test set using System 1. Green boxes are localization outputs from the algorithm, while red boxes depict some of the missed fish objects. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

tested in new environments. This is despite having only a single network trunk for System 1 - compared to the network structure of System 2 with three separate trunks.

Precision and recall measures in Test 2 however are lower compared to single crop inference on the training set in Test 1. But these could be brought further up by increasing the number of proposals of the network in each forward pass during inference. This will be performed in subsection 7.3, where the three systems are subjected to multi-crop inference testing with multi-scale distortions.

7.2.1. System 1 cascade correction mechanism

A cascade ensemble architecture can potentially perform correction mechanisms. The correction mechanism occurs as the CNN sub-networks pass information from one component in the cascade to the next such that prior errors in early cascades are rectified in future cascades. To show the need for correction mechanisms, Fig. 8 shows a proposal instance received by the first component of the R-CNN. The proposal includes IoU (Intersection over Union) in excess of 50%. Hence, it is a valid candidate for bounding box regression. However, as seen in Fig. 8, there is inherent ambiguity in information provided by the initial proposal, i.e. the actual fish object can acquire several possible

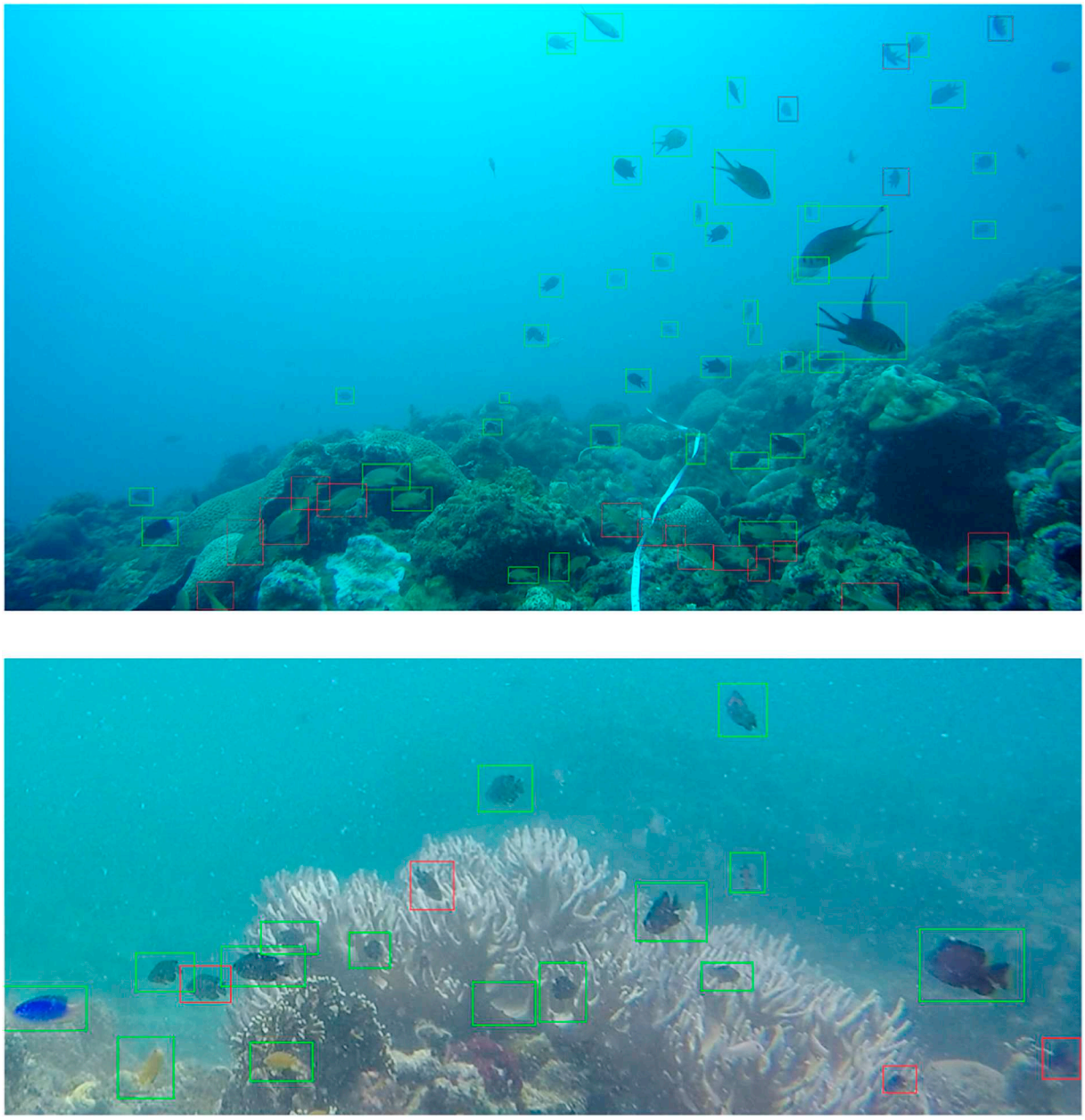

Fig. 7. Sample fish object detections taken from the test set using System 1. Green boxes are localization outputs from the algorithm, while red boxes depict some of the missed fish objects. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Test Set 1: Performance Results on 60 frames, where frames are taken from training video sequences.

System	Precision	Recall	F-Score
System 1	67.21	64.56	65.86
System 2	67.28	68.25	67.76
R-CNN Baseline	69.81	62.72	66.07

Table 4
Test Set 2: Performance Results on the 8-Video Test Set with New Backgrounds.

System	Precision	Recall	F-Score
System 1	**53.29**	**37.77**	**44.21**
System 2	47.37	33.54	39.28
R-CNN Baseline	43.99	21.00	28.43

The bold figures indicate the highest score for each performance measurement.

orientations for its tail. These orientations cannot be inferred from the initial proposal alone. If the R-CNN is limited to a single cascade, it is forced to select one of the many likely orientations of the fish object. If ever it incurs an error in bounding-box regression under a single-component R-CNN (i.e. choose a wrong orientation of the tail), it does not have a chance to rectify its error.

Fig. 9 shows the correction process that occurs in the ensemble cascade. The arrows in Fig. 9 indicate the diverse range of possible paths that the ensemble cascade can take from the initial proposal of the RPN (leftmost) to the final cascade (rightmost). The initial proposal of the RPN (denoted as x) is usually not very precise and has bounding box coordinates that do not properly enclose the fish object (where errors are represented by the term ε). The ensemble cascade relies on the assumption that as cascades progress from 1. K, errors gradually decrease by a factor of β where $\beta \in (0, 1)$. This assumption is reasonable given that each cascade minimizes a convex loss function under SGD

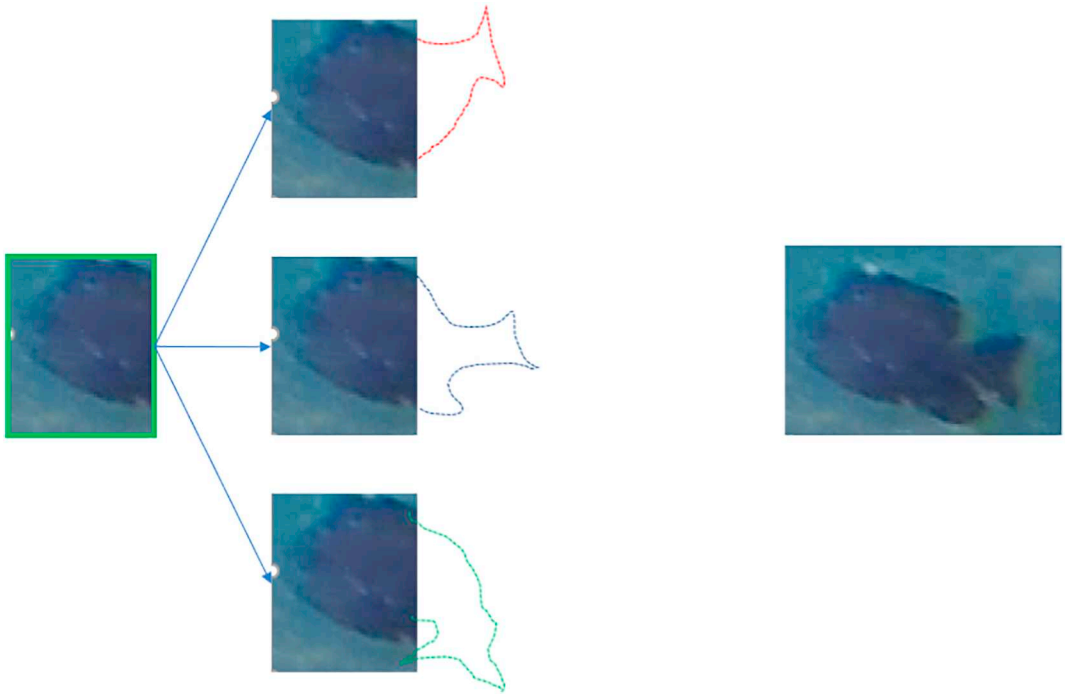

Fig. 8. Inherent ambiguity of information in initial proposals relative to ground truth (rightmost figure). The initial proposal (leftmost figure) shows a portion of a fish object. However, the actual fish object is larger than the initial proposal and can have several orientations of the tail. The correct orientation cannot be inferred from the initial proposal alone.

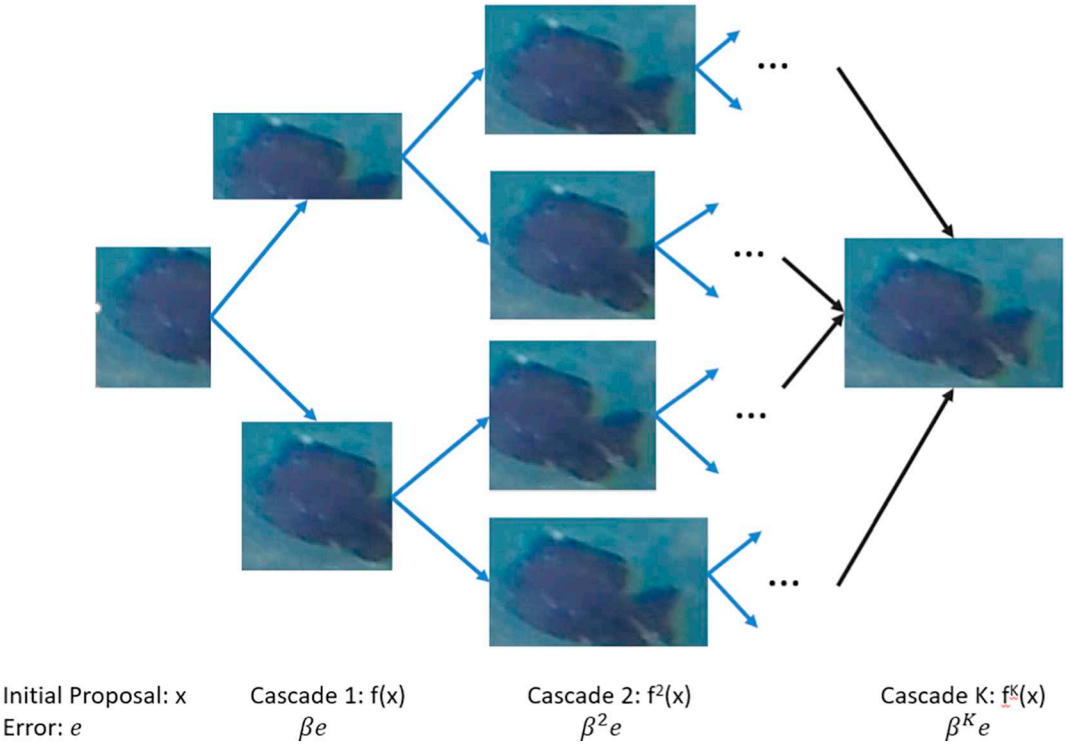

Fig. 9. Illustration of correction mechanisms. The initial proposal (leftmost figure) has an inaccurate bounding box with error ε relative to ground truth (rightmost figure). The 1st cascade uses bounding box regression (i.e. function f) to reduce error to $\beta\varepsilon$, where $\beta \in (0,1)$ is a correction parameter. Each RoI beginning at the proposal has two arrows from left to right (for illustration purposes), where these arrows express the diverse possibilities that the R-CNN in a cascade can predict. It follows that there are several possible paths from the initial proposal to the final prediction due to information ambiguity. But as long as $\beta \in (0,1)$, each prediction of a cascade serves to reduce the initial error ε. Eventually after several cascades, the predictions converge closely to the ground truth (rightmost figure), with β^K being a very small number.

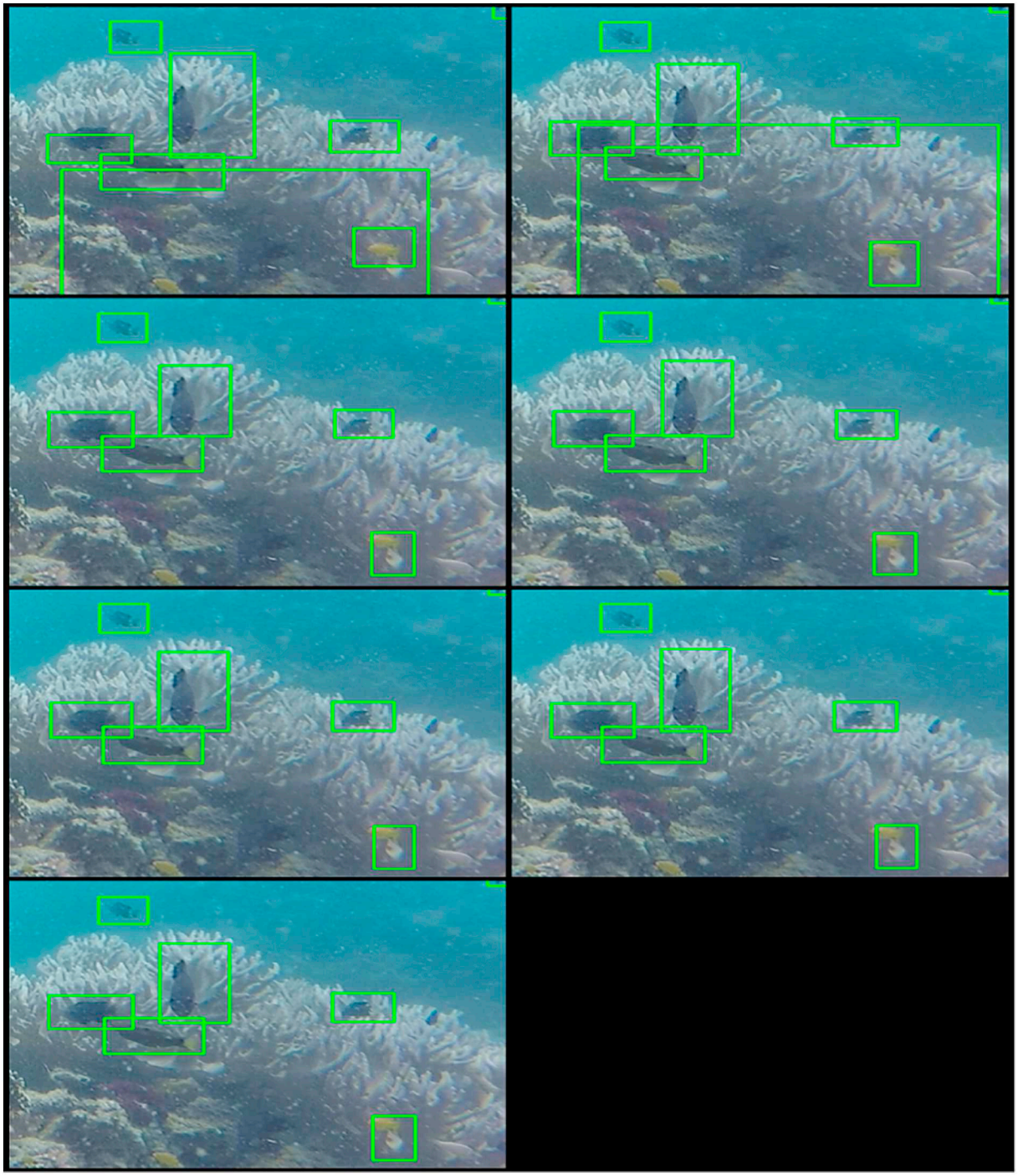

Fig. 10. Cascaded information gathering across steps. Neighboring contextual information are extracted as cascades progress: cascade 1 (1st row, 1st col), cascade 2 (1st row, 2nd col), cascade 3 (1st row, 3rd col), cascade 4 (1st row, 4th col), cascade 5 (2nd row, 1st col), cascade 6 (2nd row, 2nd col), step 7 (2nd row, 3rd col).

optimization, i.e. it is optimized to reduce bounding box errors from malformed proposals. In addition, each bounding box regression f results in a rectified proposal $f(x)$ that lessens the initial ambiguity (i.e. as shown in the rectified proposals of Fig. 9 from cascade 1 to cascade 2). Hence, $f(x)$ contains more information than x, and the next cascade $f(f(x))$ contains even more information. This process can be modeled as a recursion. Eventually, repeated applications of f results in an error of $\beta^K\varepsilon$ in the final cascade K which is a small number. The bounding boxes in this case $f^K(x)$ have close convergence to ground truth.

The mathematical expressions for the recursion are shown in Appendix. Our mathematical analysis relies on the assumption that later cascades refine prior cascades under SGD minimization of a convex loss function, i.e. with a correction parameter $\beta \in (0, 1)$. Under this assumption, the cascade ensemble is able to reduce both bias and variance compared to traditional ensemble averaging which is limited to reduction of variance only.

We can see the correction mechanism from the behaviour of bounding box predictions shown in Fig. 10. The first cascade has

predicted boxes that are poorly aligned, and a coral object is wrongly identified as fish. As the network progresses to cascade 2, several of the initial predicted boxes for fish objects are re-aligned, and the false coral detection box is expanded to include a larger area of the initial proposal. In cascade 3, the network managed to detect that the coral detection is a non-fish object and rectified its prediction. From cascade 4 to 7, only correct fish objects are identified as valid detections. This pattern across cascades show a step-wise gathering of contextual information by the ensemble CNN units around the neighboring area of an initial Region of Interest. This information gathering process occurs whenever a cascade j performs prediction of bounding box coordinate adjustments to refine the initial reference box from cascade $j-1$. The network uses the newly predicted box coordinates from cascade j to re-extract features from the shareable feature map upon the start of cascade $j+1$. In this example, a total of 7 boxes are predicted within the neighboring area of an initial proposal from cascade 1 to cascade 7. With multiple box predictions and repeated RoI-crop feature re-extractions for each cascade, there is more likelihood that the ensemble R-CNN eventually gathers key contextual information such as the locations of a fish's snout and tail or the actual boundaries of a non-fish object (e.g., coral). This contextual information gathering allows the network to automatically correct predictions for both proposal coordinates and objectness probabilities thereby increasing precision.

Aside from increasing precision, a cascade structure also improves recall. With repeated feature re-extractions and bounding box predictions, poorly formed initial RoIs in prior cascades are eventually corrected in future cascades. As corrected RoIs tend to include more contextual information, missed objects in prior cascades due to false objectness probability predictions are stochastically rectified in future cascades, leading to more true positive detections. This mechanism is shown in Fig. 11 where Cascade 1 has several false detections with poor alignments, and Cascade 2 misclassified some proposals as non-objects, missing 2 fishes. Cascade 3 up to Cascade 7 re-detected the 2 missing fishes after re-prediction of object boundaries and re-extraction of better RoI feature information. This increased recall performance.

In Fig. 11, having only two cascades may not be optimal for increasing object recall as some detections may still be missed. This explains why the baseline system which is a single non-ensemble network with only two cascades produced a low recall of 21%.

7.3. Test set 3 systems performance: robustness testing through multi-crop inference with scale distortions

This section describes another test experiment which subjects the three systems to robustness tests using eight test videos. Here, the test images are cropped according to nine different sections, and predicted detections for all 9 sections are combined for final inference. (See Fig. 12.) Each cropped section is tested according to a scale multiplier of 0.75, 1.0, and 2.0. The rationale behind this method is to test the generalization capacity of the three systems according to different image scales while removing portions of the global context. If the system performs well despite the removal of global context and scale distortions, it means that the system can generalize and is robust to overfitting. Among the different scales, the system is expected to perform worse for a scale resize of 0.75, since information is lost upon down-sampling of the image by 25%.

We note that multi-crop inference could actually increase system performance in some instances (Fathi et al., 2019) since it allows networks to focus on a sub-region. Given nine sub-sections, the total number of proposals could increase up to nine times. However, improvements in localization is dependent on the network's capacity towards to predict well despite removal of global contexts. In this test, each system is allowed 2000 proposals per section along with an objectness probability threshold per detected fish object of 70%.

From Tables 5, 6 and 7, all three systems performed worse for the scale resize of 0.75. This type of performance degradation is expected given that downsampling of the image removes key information defining fish objects. However, even with downsampling at this rate, System 1 performs best with the highest F-score of 33.64. The separate network ensemble of System 2 performed poorly with an F-score of 24.65, indicating that the system is not robust to distortions of smaller image scales.

Given a scale multiplier of 1.0, both System 2 and the baseline system displayed better performance compared to non multi-crop inference in accordance with the findings in (Fathi et al., 2019) for ensembles with separated networks. While System 2 has the highest recall given a scale multiplier of 2.0, its precision suffered, with a score of only 40.01. This indicates that System 2, inspite its ensemble mechanism, is not very consistent given different scales. The most consistent model for multi-crop inference across all scale resize ratios is System 1. In fact, System 1's F-score increased from 44.21% in non multi-crop inference to 48.84% given a scale multiplier of 1.0, and to a larger value of 56.15 given a scale multiplier of 2.0. The large increase indicates that System 1 can utilize the image's expanded resolution to improve detection. Among all the tests conducted in this experiment, System 1 with 2.0 scale increase and multi-crop inference reported the best performance.

7.4. Test set 4 systems performance: ablation tests for system 1

We implement Test Set 4 as an ablation test for System 1 where we determine the effect of the recurrent LSTM links. Instead of LSTM links, we implement vector links in the form of flattened feature maps. More specifically, for Step 5 in the algorithm shown in Sec. 3.3, instead of the 2048 dimensional hidden units S_{j-1} that are passed to the fully connected layer, we implement average pooling over the 3×3 sub-feature map produced by the CNN component resulting in a 2048-d vector. To serve as link, we concatenate the 2048-d averaged vector in cascaded component j with the respective 2048-d averaged vector in the previous cascade component $j-1$. The result is a 4096-d concatenated vector that serves as input to the fully connected layer for bounding box prediction.

We implement the same multicrop tests as in Test Set 3 to the modified System 1 with vector links. From Table 8 to Table 10, we show the results of the original System 1 with LSTM links, the modified System 1 with vector links, and the baseline system 3. We choose to include system 3 among the results in Test Set 4 since it is equivalent to a 2-component cascaded ensemble variant of System 1.

From Table 8, having LSTM links at a scale distortion of $0.75\times$ do not indicate any performance improvement, as performance from the modified System 1 is comparable with the original system. But from Table 9 to Table 10, it could be seen that System 1's performance with LSTM links improves, while the modified System 1 with vector links reports bad performance at a scale distortion of $2.0\times$. In fact, the baseline non-ensemble system performs even better than the modified System 1 at a scale distortion of $2.0\times$. This indicates that LSTM links with attention mechanisms provides more robustness since it is able to maintain good performance despite multiple scale distortions.

7.4.1. Insights on attention mechanisms in system 1 LSTM unit

The robustness of System 1's performance can be attributed to the attention mechanism in System 1's LSTM which focuses on sub-regions and links their features in a sequential fashion. It does not rely on features taken from the entire object RoI image, compared to Systems 2 and 3, which convolve on the entire 6×6 RoI feature map after RoI-cropping. This means that System 1 has to detect the key features of an

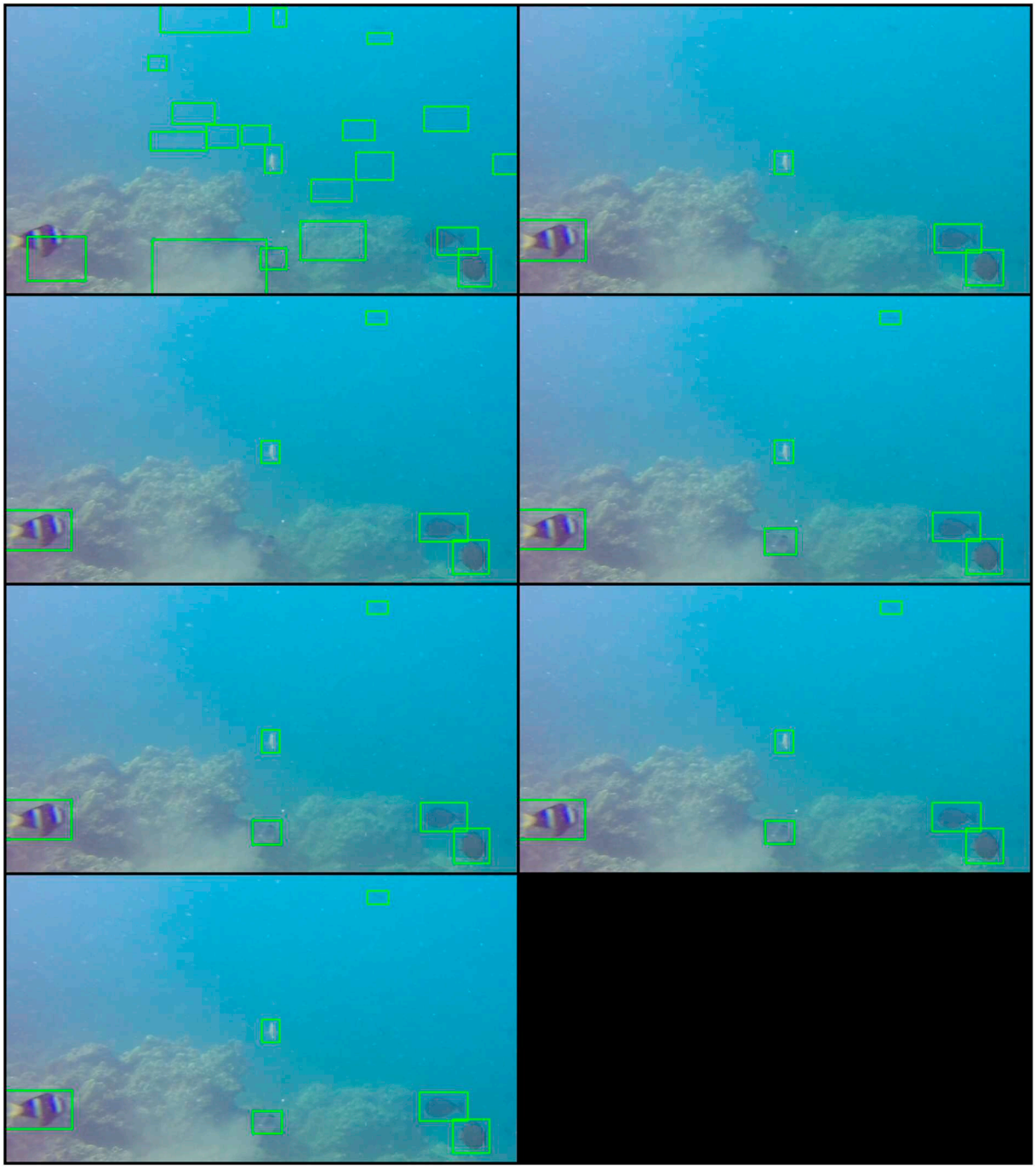

Fig. 11. Increased recall with 7-cascade R-CNN: Cascade 1 (1st row, left), cascade 2 (1st row, right), cascade 3 (2nd row, left), cascade 4 (2nd row, right), cascade 5 (3rd row, left), cascade 6 (3rd row, right), cascade 7 (4th row).

object and re-orientations due to scale distortions do not affect inference. A sample of the attention mechanism procedure is shown in Fig. 13, where 9 key features of the fish object are gathered and arranged in a sequential manner. The LSTM unit uses the sequence of fish object portions to construct the overall object. Since it does not depend on global information, it is rendered more robust to scale distortions, similar to a 'bag-of-words' scheme.

8. Future work

For future research, System 1 can be modified to include additional sub-networks for various tasks, i.e. species classification or semantic segmentation. The network can likewise be modified to incorporate temporal information captured in frame sequences, e.g. fish movements. This allows prediction not only on static fish locations but also on their behavioural (swimming) patterns. In terms of cascade

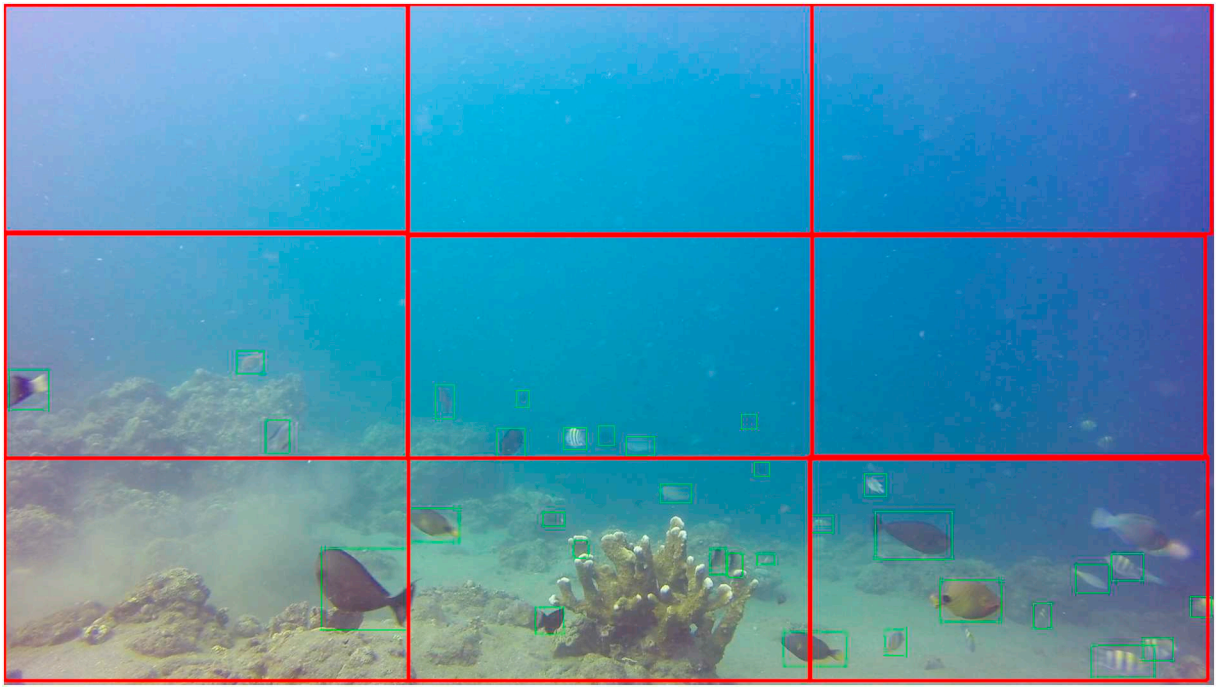

Fig. 12. Multi-Crop Inference with Nine (9) Subsections. During inference, the network systems processes each subsection independently. This leaves out global information, but allows the networks to focus more proposals on a single cropped section during inference - leading to more detections for increased recall.

Table 5
Test Set 3: Multi-Crop Inference Performance Statistics (Scale Multiplier: 0.75).

System	Precision	Recall	F-Score
System 1	**39.56**	**29.56**	**33.64**
System 2	28.89	21.19	24.45
Baseline	33.74	17.34	22.91

The bold figures indicate the highest score for each performance measurement.

Table 6
Test Set 3: Multi-Crop Inference Performance Statistics (Scale Multiplier: 1.0).

System	Precision	Recall	F-Score
System 1	48.51	**49.18**	**48.84**
System 2	48.25	27.81	35.28
Baseline	**55.00**	21.50	30.92

The bold figures indicate the highest score for each performance measurement.

Table 7
Test Set 3: Multi-Crop Inference Performance Statistics (Scale Multiplier: 2.0).

System	Precision	Recall	F-Score
System 1	**60.32**	52.52	**56.15**
System 2	40.01	**61.16**	48.42
Baseline	47.11	44.20	45.61

The bold figures indicate the highest score for each performance measurement.

Table 8
Test Set 4: Multi-Crop Inference Ablation Tests (Scale Multiplier: 1.0).

System	Precision	Recall	F-Score
System 1 (w/ LSTM)	39.56	29.56	33.64
System 1 (w/ vector links)	**39.04**	**32.37**	**35.38**
Baseline (System 1 w/ 2 cascades)	33.74	17.34	22.91

The bold figures indicate the highest score for each performance measurement.

Table 9
Test Set 4: Multi-Crop Inference Ablation Tests (Scale Multiplier: 1.0).

System	Precision	Recall	F-Score
System 1 (w/ LSTM)	48.51	**49.18**	**48.84**
System 1 (w/ vector links)	54.72	34.68	42.45
Baseline (System 1 w/ 2 cascades)	**55.00**	21.50	30.92

The bold figures indicate the highest score for each performance measurement.

Table 10
Test Set 4: Multi-Crop Inference Ablation Tests (Scale Multiplier: 2.0).

System	Precision	Recall	F-Score
System 1 (w/ LSTM)	**60.32**	**52.52**	**56.15**
System 1 (w/ vector links)	30.29	38.27	33.82
Baseline (System 1 w/ 2 cascades)	47.11	44.20	45.61

The bold figures indicate the highest score for each performance measurement.

structure, further research can be done to analyse the effects of cascade lengths on overall accuracy. In addition, we constructed System 1 to work well in offline server systems, i.e. computers with GPU's. For ROV platforms with lower computational resources, System 1 can be simplified to accomodate computational hardware with less memory without compromising much of its accuracy.

9. Conclusion

We constructed three deep network object detection systems to perform large-scale fish object detection over a new dataset consisting of 20 to 100 fish objects with majority having object sizes of 100 to 2500 square pixels. We report the following technical contributions:

1. We compare three types of ensemble systems, where the first system consists of an integrated network with LSTM-linked ensemble components arranged in a cascade. The second system is a traditional ensemble system patterned after G-RMI where predictions are combined from three separate residual networks at the last stage of

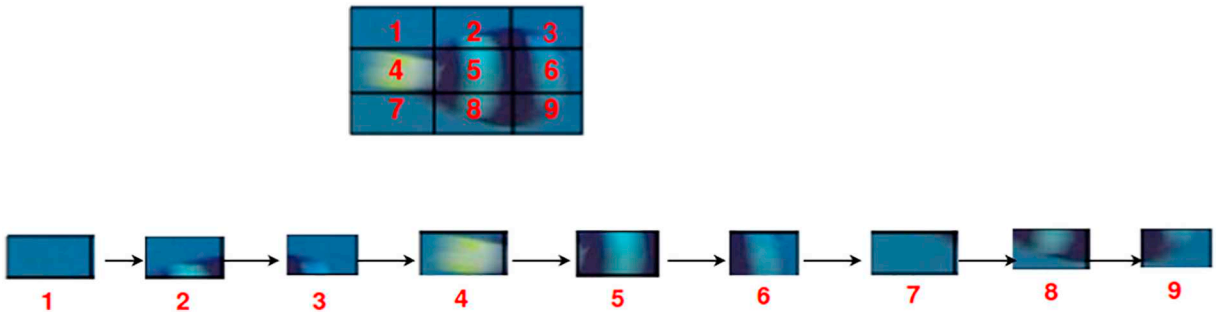

Fig. 13. Sample sub-image sequence in the LSTM unit of System 1. The LSTM unit focuses on key features from each sub-section in the image given a 3 × 3 RoI feature map produced by a CNN unit.

inference. The third system is a Faster R-CNN baseline. We implement four tests to assess the three system's performance.

2. Ensembled fish object detection networks outperform the baseline single model when tested on new frames. Comparing the two types of ensemble structures, an integrated system network with linked multi-cascade ensemble units performs better than an ensemble composed of three separately trained networks.
3. The cascade structure of the integrated network of System 1 relies on an automatic correction mechanism involving a step-wise gathering of contextual information around the initial proposal. As cascades progress, link connections between cascades propagate key feature information to correct bounding box and objectness predictions. This lessens initial information ambiguity from early proposals.
4. The sequential LSTM link allows the network to improve generalization by focusing on key-features of objects according to an attention mechanism. This enables the network structure in System 1 to report the best performance given multi-scale distortions with multi-cropping.
5. With multi-crop inference, scale distortions and removal of global cues, System 1 displays good generalization capacity. The best performance given System 1 is under a scale increase of 2× and 9-section multi-crop inference.
6. Ablation tests indicate that recurrent LSTM links are crucial for generalization performance.

Acknowledgment

This work was supported by the Philippine Council for Industry, Energy and Emerging Technology Research and Development of the Department of Science and Technology under the FishDrop Project. The authors also wish to thank Dr. Laura T. David and Mr. Mark Manalo of the Ocean Color and Coastal Oceanography Laboratory, Marine Science Institute, University of the Philippines Diliman. Mr. Mark Manalo was responsible for manually annotating the training frames for the fish objects.

Appendix A

A.1. Table of Acronyms

A.1.1. System 2, and baseline system: R-CNN with 2 cascade components

System 2 is an ensemble system based on three separate networks following (Fathi et al., 2019), where each network has its own RPN and R-CNN that are trained separately, and the outputs of the three networks are combined upon inference. This results in three network trunks and three separate RPNs in total. Similar to System 1, each of the 3 networks is fitted with a 50-layer residual network for feature generation along with its own RPN and 2-cascade R-CNN similar to (Dai et al., 2015). The first network has an RPN *base_size* of 3, the second network has an RPN *base_size* of 2, while the third has an RPN *base_size* of 5. Thus, System 2 trains three separate RPNs. The final predictions on bounding box coordinates and objectness probabilities of each network are combined during inference. One benefit of this system is that each network is able to specialize on a certain fish size given the multiplicity of anchor box aspect scales and sizes for each network. This system increases the network's generalization capacity since it is able to leverage on the different specializations of each network in the ensemble. Averaging of the ensemble components in System 2 is indirectly performed through the NMS process, where the NMS combines boxes that are located close to each other. Hence, if several boxes are estimated over a certain portion, it is likely that majority of the network components of System 2 predicted an object located over the portion, resulting in a form of majority vote. For a proposal to be classified as a detected object, the majority vote has to surpass a certain threshold of the objectness probability. To increase recall, we do not set the threshold limit very high.

DL	Deep Learning
CNN	Convolutional Neural Network
R-CNN	Region Convolutional Neural Network
Faster R-CNN	Faster Region Convolutional Neural Network (the basic localization network)
G-RMI	Google Research and Machine Intelligence (a type of ensemble localization network)
MNC	Multi Network Cascade (a type of cascade localization network)
LSTM	Long Short-Term Memory Unit (a type of recurrent neural network)
SEACLEF	dataset for fish object localization
PASCAL VOC	dataset for object localization (with larger and fewer objects than COCO)
COCO	dataset for object localization (harder dataset for localization than PASCAL VOC)

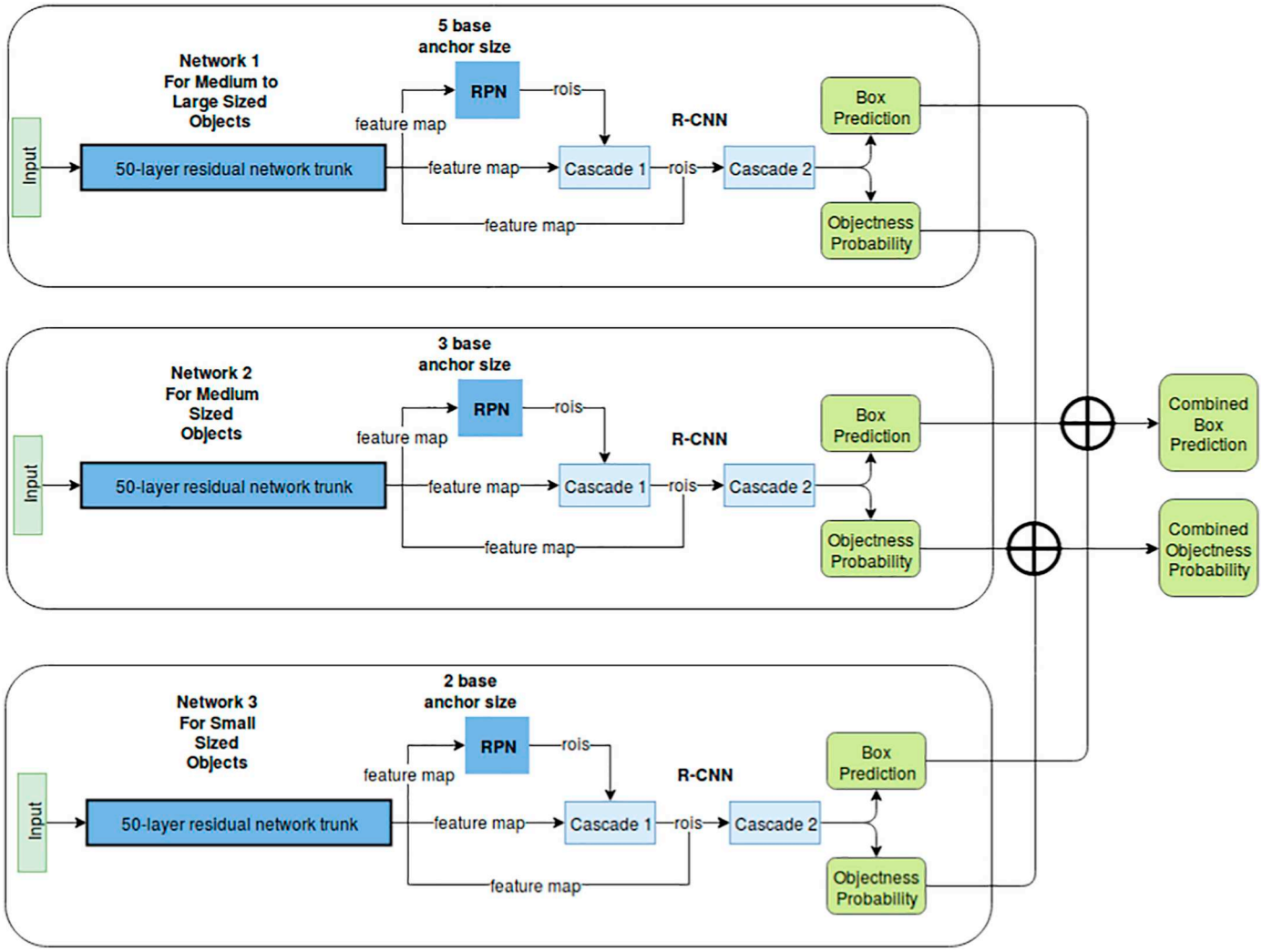

Fig. 14. System 2 Network Structure - 3 separate networks are trained separately, each with its own trunk and Region Proposal Network and R-CNN. The R-CNN has two cascade components for proposal refinement. Final bounding box and objectness probabilities come from a combination of the three separate networks.

The baseline system follows a Faster-R-CNN network structure fitted with two cascade components similar to MNC, and serves as the baseline for comparison (Zhuang et al., 2017). It has a single 50-layer residual network trunk that branches out to 1 PRN and a 2-cascade R-CNN. It mirrors closely the network in (Dai et al., 2015) and in (Zhuang et al., 2017) where Faster-R-CNN is applied to fish detection. Due to the lack of ensemble mechanisms in this system however, it is expected to be outperformed by Systems 1 and 2 as verified in (Fathi et al., 2019).

A.2. Analysis on cascade ensemble

To express our ideas regarding the benefit of cascade ensembles, we present a simple mathematical model on the cascade correction mechanism. Let there be K components on the cascade. Without loss of generality, let $f_i(x)$ denote a computable function whose parameters are the optimal network weights of each component in the cascade. These components are indexed by $i \in K$, s.t. $f_i(x) = f_j(x) \quad \forall \quad i \neq j \in K$ given input x - i.e. all cascade components are equal and are at their optimal values (under SGD minimization with convex loss). Without loss of generality, let $f(x_i) = x_{i+1}$, i.e. $f(x_i) : \mathbb{R}^d \rightarrow \mathbb{R}^d$, where d is the dimension of the feature map, i.e. f maps sections of feature maps to sections feature maps through the Roi-Cropping process. From Section 3.3.1, the cascades result in a recursion using f, where:

$$
\begin{aligned}
x_0 &= \text{input for cascade 1} \\
x_1 &= f_1(x_0) \\
x_2 &= f_2(x_1) = f_2(f_1(x_0)) \\
&\cdots \\
x_i &= f_i(f_{i-1}(\cdot f_1(x_0))
\end{aligned} \tag{4}
$$

The recursion results in a sequence of inputs $[x_0, x_1, x_2 \ldots x_K]$, where each input x represents a section of the feature map (enclosed in a regressed bounding box). The cascade ensemble supposes that as cascades progress, the coordinates of the section's bounding box approaches closer its ground-truth coordinates. More formally, let $b(x)$ be a function that computes the coordinates of x, i.e. $b(x) \rightarrow [x1, y1, x2, y2]$. Let the ground-truth coordinates be b^*. The cascade ensemble assumes that $|b(x_i) - b^*| \geq |b(x_j) - b^*| \, \forall \, j > i \in K$, s.t. $|b(x_i) - b^*| \rightarrow 0$ and $b(x_i) \rightarrow b^*$ as $i \rightarrow \infty$. The result of this assumption is that inputs for higher indexed cascades have less error and possess more information on the actual object. This assumption is reasonable given Eq. 2, where each cascade tries to minimize a convex loss function using SGD for accurate bounding box regression. Hence, an application of $f(x)$ results in coordinates $b(x)$ that are closer to b^*. This process is actually the same as what is done by the R-CNN in the Faster-R-CNN

for proposals from the RPN, i.e. proposal refinement.

Now given more informative inputs x, it is reasonable to assume that under uniform optimal cascade parameters (contained in $f(x^*)$, with x^* as the ground-truth feature map) cascade errors decrease for higher-indexed cascades (in the case of fish objects, higher-indexed cascades are able to 'see' more of the fish such as its shape, fins, tail, etc.). For instance, let f^* denote the ground truth feature map, where $b^*(f^*) = [x^*1, y^*1, x^*2, y^*2]$ provides the ground truth coordinates of the object. Given f^*, we can use f s.t. $b(f^*) = b(f(x)) + \varepsilon_b$ for the coordinate regression and $p^*(f^*) = p(f(x)) + \varepsilon_p$ for the objectness probability. We conjecture that $|b(f(x_j)) - b^*(f^*)| < |b(f(x_i) - b^*(f^*)|\, \varepsilon_{b,j} < \varepsilon_{b,i}$ for $j > i, \quad \forall j, i \in K$. We express this relationship as follows for bounding box coordinate regression:

$$\begin{aligned} b^*(f^*) - b(f_{i+1}(x_i)) &= b(f_{i+1}[x_i]) + \varepsilon_{b,i+1} \\ \text{where } \varepsilon_{b,i+1} \text{ is:} & \\ \varepsilon_{b,i+1} &= \beta \varepsilon_{b,i} \\ &= \beta [b(f^*) - b(f_i(x_{i-1}))] \end{aligned} \tag{5}$$

$\beta \in (0,1)$ is a 'correction parameter' that adjusts errors of prior cascades. Using Eq. 4 and 5, we form a recursion of $b(f)$ relative to ground truth $b^*(f^*)$:

$$\begin{aligned} b^*(f^*) &= b[f_1(x_0)] + \varepsilon_{b,0} \text{ start of cascade with input } 0 \\ b^*(f^*) &= b[(f_2(x_1)] + \varepsilon_{b,2} = f[f(x_0)] + \beta \varepsilon_{b,0} \\ b^*(f^*) &= b[(f_i(x_{i-1})] + \varepsilon_{b,i} = f^i[f(x_0)] + \beta^i \varepsilon_{b,0} \\ &\cdots \\ b^*(f^*) &= b[f_K(x_{K-1})] + \varepsilon_K = f^K(x_0) + \beta \prod_{K-1} \beta \varepsilon_{b,0} \\ &= b[f^K(x_0)] + \beta^K \varepsilon_{b,0} \end{aligned}$$

where $[|b[f^*] - b[f^K(x_0)]|] \to 0$ as cascades progress $i \to K$ given the assumption on correction parameter $\beta \in (0,1)$. This implies $b[f^K(x_0)] \to b^*[f^*]$ assuming that f represents optimal weights as $k \to \infty$. This expresses the notion that as cascades progress from 1. K, they refine the original error e_0, and provide more accurate bounding boxes over a refined feature map $b(f_i(x))$. The same equations apply for predictions on objectness probabilities p.

$$\begin{aligned} p^*(f^*) &= p[f_1(x_0)] + \varepsilon_{p,0} \text{ start of cascade with input } 0 \\ p^*(f^*) &= p[(f_2(x_1)] + \varepsilon_{p,2} = f[f(x_0)] + \beta \varepsilon_{b,0} \\ p^*(f^*) &= p[(f_i(x_{i-1})] + \varepsilon_{p,i} = f^i[f(x_0)] + \beta^i \varepsilon_{p,0} \\ &\cdots \\ p^*(f^*) &= p[f_K(x_{K-1})] + \varepsilon_K = f^K(x_0) + \beta \prod_{K-1} \beta \varepsilon_{p,0} \\ &= p[f^K(x_0)] + \beta^K \varepsilon_{p,0} \end{aligned}$$

Hence, comparing a cascade ensemble f with a traditional ensemble g, suppose that all cascade and traditional ensemble components f and g have equal variance σ^2 with no correlation $\mathbb{E}[\sigma_i^2 \sigma_j^2] = 0$ for $i \neq j$. Then variances in both ensemble types are equal. However, for K components in both types we have:

$$|b^*(f^*) - f_K(x_{K-1})| \leq \left| b^*(f^*) - \frac{1}{K} \sum Kg(x_0) \right|$$

i.e. the bias in cascade estimates $f_K(x_{K-1})$ at the end of the Kth cascade is less than traditional ensembles g which merely average the outputs of the K components.

References

Spampinato, C., Giordano, D., Di Salvo, R., Chen-Burger, Y.-H.J., Fisher, R.B., Nadarajan, G., 2010. Automatic fish classification for underwater species behavior understanding. In: Proceedings of the First ACM International Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams. ACM, pp. 45–50.

LeCun, Y., Touresky, D., Hinton, G., Sejnowski, T., 1988. A theoretical framework for back-propagation. In: Proceedings of the 1988 Connectionist Models Summer School. vol. 1. CMU, Pittsburgh, Pa: Morgan Kaufmann, pp. 21–28.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105.

Simonyan, K., Zisserman, A., 2014. Very Deep Convolutional Networks for Large-scale Image Recognition. CoRR abs/1409.1556. arXiv:1409.1556. URL. http://arxiv.org/abs/1409.1556.

Li, J., Liang, X., Li, J., Xu, T., Feng, J., Yan, S., 2016. Multi-stage Object Detection With Group Recursive Learning. CoRR abs/1608.05159. URL. http://arxiv.org/abs/1608.05159.

Dai, J., He, K., Sun, J., 2015. Instance-aware Semantic Segmentation Via Multi-task Network Cascades. CoRR abs/1512.04412. arXiv:1512.04412. URL. http://arxiv.org/abs/1512.04412.

Li, X., Shang, M., Qin, H., Chen, L., 2015. Fast accurate fish detection and recognition of underwater images with fast r-cnn. In: OCEANS 2015 - MTS/IEEE Washington, pp. 1–5. https://doi.org/10.23919/OCEANS.2015.7404464.

Villon, S., Chaumont, M., Subsol, G., Villéger, S., Claverie, T., Mouillot, D., 2016. Coral reef fish detection and recognition in underwater videos by supervised machine learning: Comparison between deep learning and hog + svm methods. In: International Conference on Advanced Concepts for Intelligent Vision Systems. Springer, pp. 160–171.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.

Krogh, A., Vedelsby, J., 1995. Neural network ensembles, cross validation, and active learning. In: Advances in Neural Information Processing Systems, pp. 231–238.

Hastie, T., Tibshirani, R., Friedman, J., 2013. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer Series in Statistics. Springer New York URL. https://books.google.com.ph/books?id=yPfZBwAAQBAJ.

Hara, K., Liu, M., Tuzel, O., Farahmand, A., 2017. Attentional Network for Visual Object Detection. CoRR abs/1702.01478. arXiv:1702.01478. URL. http://arxiv.org/abs/1702.01478.

Lin, T., Maire, M., Belongie, S.J., Bourdev, L.D., Girshick, R.B., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft COCO: Common Objects in Context. CoRR abs/1405.0312. arXiv:1405.0312. URL. http://arxiv.org/abs/1405.0312.

Costa, C., Loy, A., Cataudella, S., Davis, D., Scardi, M., 2006. Extracting fish size using dual underwater cameras. Aquac. Eng. 35 (3), 218–227.

Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A., 2010. The pascal visual object classes (voc) challenge. Int. J. Comput. Vis. 88 (2), 303–338.

Fathi, A., Korattikara, A., Sun, C., Fischer, I., Huang, J., Murphy, K., Zhu, M., Guadarrama, S., Rathod, V., Song, Y., et al., 2019. G-rmi Object Detection. URL.

http://image-net.org/challenges/talks/2016/GRMI-COCO-slidedeck.pdf.
Garcia, R., Nicosevici, T., Cuf, X., 2002. On the way to solve lighting problems in underwater imaging. In: OCEANS'02 MTS/IEEE. vol. 2. IEEE, pp. 1018–1024.
Gers, F.A., Schmidhuber, J., Cummins, F., 2019. Learning to Forget: Continual Prediction With lstm.
Girshick, R., Donahue, J., Darrell, T., Malik, J., 2016. Region-based convolutional networks for accurate object detection and segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 38 (1), 142–158. https://doi.org/10.1109/TPAMI.2015.2437384.
Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y., 2016. Deep Learning. vol. 1 MIT Press Cambridge.
Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.
Hollowed, A.B., Barange, M., Beamish, R.J., Brander, K., Cochrane, K., Drinkwater, K., Foreman, M.G., Hare, J.A., Holt, J., Ito, S.-i., et al., 2013. Projected impacts of climate change on marine fish and fisheries. ICES J. Mar. Sci. 70 (5), 1023–1037.
Horgan, J., Toal, D., 2009. Computer Vision Applications in the Navigation of Unmanned Underwater Vehicles. pp. 194–214.
Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L., 2014. Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 1725–1732.
Katsanevakis, S., Weber, A., Pipitone, C., Leopold, M., Cronin, M., Scheidat, M., Doyle, T.K., Buhl-Mortensen, L., Buhl-Mortensen, P., Anna, G., et al., 2012. Monitoring marine populations and communities: methods dealing with imperfect detectability. Aquat. Biol. 16 (1), 31–52.
Labao, A., Naval, P., 2017. Non-motion-based segmentation of fish objects in underwater videos using resnet-fcn. In: Asian Conference on Intelligent Information and Database Systems.
LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436.
Lee, M., Lee, J., Chang, J.-H., 2019. Ensemble of jointly trained deep neural network-based acoustic models for reverberant speech recognition. Digital Signal Process. 85, 1–9.
Mieszkowska, N., Sugden, H., Firth, L., Hawkins, S., 2014. The role of sustained observations in tracking impacts of environmental change on marine biodiversity and ecosystems. Philos. Trans. R. Soc. A Math. Phys. Eng. Sci. 372 (2025), 20130339.
Negahdaripour, S., Yu, C.H., 1995. On shape and range recovery from image shading for underwater applications. In: Underwater Robotic Vehicles: Design and Control, pp. 221–250.
Ren, S., He, K., Girshick, R., Sun, J., 2017. Faster r-cnn: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 39 (6), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031.
Siddiqui, S.A., Salman, A., Malik, M.I., Shafait, F., Mian, A., Shortis, M.R., Harvey, E.S., 2017. Automatic fish species classification in underwater videos: exploiting pre-trained deep neural network models to compensate for limited labelled data. ICES J. Mar. Sci. 75 (1), 374–389 H. editor: Howard Browman.
Spampinato, C., Chen-Burger, Y.-H., Nadarajan, G., Fisher, R.B., 2008. Detecting, tracking and counting fish in low quality unconstrained underwater videos. VISAPP (2) 2008 (514–519), 1.
Spampinato, C., Palazzo, S., Boom, B., van Ossenbruggen, J., Kavasidis, I., Di Salvo, R., Lin, F.-P., Giordano, D., Hardman, L., Fisher, R.B., 2014. Understanding fish behavior during typhoon events in real-life underwater environments. Multimed. Tools Appl. 70 (1), 199–236.
Walsh, S.J., Godø, O.R., Michalsen, K., 2004. Fish behaviour relevant to fish catchability. ICES J. Mar. Sci. 61 (7), 1238–1239.
Wang, H., Shen, Y., Wang, S., Xiao, T., Deng, L., Wang, X., Zhao, X., 2019. Ensemble of 3d densely connected convolutional network for diagnosis of mild cognitive impairment and alzheimerâ€™s disease. Neurocomputing 333, 145–156.
Zhuang, P., Xing, L., Liu, Y., Guo, S., Qiao, Y., 2017. Marine animal detection and recognition with advanced deep learning models. In: Working Notes of CLEF.
Zion, B., 2012. The use of computer vision technologies in aquaculture–a review. Comput. Electron. Agric. 88, 125–132.