Evaluation of transfer learning techniques for classification and localization of marine animals

Parmeet Singh[1], Dr. Mae Seto[2]
[1]*Faculty of Computer Science, Dalhousie University, Halifax, Nova Scotia, Canada*
[2]*Department of Mechanical Engineering, Dalhousie University, Halifax, Nova Scotia, Canada*
pr819318@dal.ca, mae.seto@dal.ca

Keywords: Convolution Neural Networks, Ensemble Learning, Marine Animals, VGG, Inception , Computer Vision , Bounding Boxes

Abstract: The objective is to evaluate methods for simultaneous classification and localization towards a better size estimate of marine animals in still images. Marine animals in such images vary in orientations and size. It is challenging to create a bounding box that predicts the shape of the object. We compare axis-aligned and rotatable bounding box techniques for size estimation.

1 INTRODUCTION

1.1 Motivation

This paper reports on research and development to improve sustainability and traceability for the marine stewardship that keeps the oceans healthy and ensures the fish or lobster consumed is traceable to a sustainable source. There are ecological and economic sustainability reasons to assess and monitor stocks and stock sizes. Traceability aims to uniquely identify each animal for consumer quality purposes. The methods developed here target the Canadian fishery due to the small number of species involved. The aim of the project is to determine the count and species of a catch as well as each species' length and color (measure of quality) for economic sustainability reasons. Another objective is to determine the gender and characteristic dimensions, and their ratios, for ecological sustainability. For traceability, imagery is collected to assess whether say the carapace on a lobster can be uniquely identified.

The Canadian Department of Fisheries and Oceans is interested in such technology for in-shore waters. The present use of trained human Observers can only monitor 1.5-2.0% of the lobster fleet activity not the 20% to manage non-endangered species (lobster) or the 50% for endangered species (cod, cusk, jonah crab). Lastly, value chain stakeholders have articulated their requirement for primary grading (size and quality) while at-sea to guarantee lobster quality and value.

A catch from a fishing trawler can contain multiple marine animal species. They are usually manually sorted, by species, then shipped to different factories for further classification and processing. As part of this, there is a requirement to sort these animals based on their physical maturity and size.

Sorting and classifying marine catch (fish, in this case) on the basis of their dimensions and species using pattern recognition algorithms is proposed. Consequently, it is possible to automate the sort and classification steps of the fish processing with computer vision to improve traceability, profit margins and product quality. The collateral benefit is that the size distribution by species, of a catch, also has ecological significance towards characterizing the fish population and its evolution. There are earlier efforts in fish classification (Rathi et al., 2018)(Larsen et al., 2009)(Ogunlana et al., 2015) however, there is much less on estimation of the fish dimensions. This paper explores methods to simultaneously localize the fish in a static image, from amongst other objects, through a bounding box that contains the fish. Then, it determines its dimensions and classifies the fish by species.

The data was gathered from static images that were culled from publicly available on-line, and other, sources. The training dataset was created by manually cropping these images around the object (fish) with two-dimensional bounding boxes using software tools. Therefore, the features in this learning problem are the marine animal dimensions and species where the species itself implicitly contains many subfeatures which are learned.

When the model is applied to prediction, this object localization using bounding boxes also automates determining the dimensions of the fish. The CNN architectures used in the prediction are discussed next.

1.2 Convolution Neural networks

The neurons in a neural network provide an output value from applying a function to the input values from the receptive field in the prior layer. This function, also referred to as a filter, is in the form of a vector of weights and a bias. Learning is achieved through incremental changes to these weights and bias.

The weight parameters of CNN architectures like VGG16 and Resnet were trained on the ImageNet dataset. This is a database of 14 million images that contain 20,000 categories of objects which have been manually annotated. The ImageNet objects include crabs, lobsters and fish though not hallibut, cusk or cod - all objects of interest. Over one million of these images have bounding boxes drawn around the object. The computation effort to train the ImageNet data set was leveraged to initialize our learning models by using the weights / filters from the same CNN architectures. In this way, the knowledge learned from the pre-trained models was transferred to our learning models.

Our CNN architectures were initialized with the ImageNet weights. Next, more convolution and fully connected layers were added then it was re-trained using our fish data set. This transfer learning process to initialize the CNN architecture with pre-trained weights provides a good initial point for the learning model and reduces the overall computation time.

Then, the prediction phase uses the transferred learning to 'measure' the fish dimensions and classify its species. Therefore, there is value in creating a good training set.

Convolution neural networks (CNN) learn the features of an object in order to classify it. CNNs such as VGG16 (Simonyan and Zisserman, 2014), VGG19 (Simonyan and Zisserman, 2014), Resnet (He et al., 2016), etc. are trained with many images for that reason. The pre-trained weights of these networks can be used to classify a smaller set of images and consequently leverage the training effort from previous networks. This is transfer learning. This paper (1) evaluates an ensemble of the pre-trained CNNs to classify images of marine animals and (2) creates a bounding box area around the object to estimate its width and height.

1.3 Contributions

The contributions of this paper are as follows: (1) consider and prove viable the use of deep learning towards automating the classification and sizing of marine animals; (2) the implementation of bounding boxes to yield an automatic measure of an object's dimensions for sorting purposes, and (3) a classification that does a fair job of distinguishing between 5 marine animal species

The rest of this paper is organized as follows. Related work in the literature is reviewed to provide context for the authors' contributions. Then, the proposed methodology is described in detail with sub-sections on the training data, model, training and then evaluation.

2 RELATED WORK

Rathi et. al. (Rathi et al., 2018) performed classification of fish species. They perform pre-processing steps of Otsu's binarization, dilation and erosion to improve the quality of the image. They add the preprocessed image as the fourth channel to an already existing RGB image and use CNN for classification of fish images. Rathi et al. have used a custom CNN architecture which they trained from scratch. We have used pretrained CNN architectures which uses transfer learning to leverage the large amount of training time.

White et. al. (White et al., 2006) determines the orientation of the fish based on the moments of the polygon spanned by the fish silhouette. They determine the fish species based on colour and shape. White et al determine the length of the fish by mapping eight points on the detected outline of the fish. Contrary to White et al,we plan to determine the width and height of the fish by creating a bounding box around the fish. The bounding box technique is robust to various types of fishes.

Larsen et. al. (Larsen et al., 2009) perform classification of fish species based on shape and texture features. They estimate the parameters of an active appearance model using geometric and texture-based features. Subsequently, they apply principal component analysis (PCA) to these model parameters which yield features that they apply Linear Discriminant Analysis to for an accuracy of 76 percent. Their method is dependant of building a separate model apart from the learned one for each species of a fish. Our CNN architecture creates an intrinsic model of the fish species for classification and does not need a separate model.

Hsieh et al. (Hsieh et al., 2011) proposed a technique to measure a tuna fish's snork to fork length using Hough transform. The longest line measured by Hough transform in the image can indicate the length of the fish. They transform every point in image space to Hough space. The collinear points in the image space are presented in the Hough space. Therefore, the weight of the largest peak is the fish length in the image. In other words, this technique measures the longest length of an object in the image. However, the technique might not work if there are other objects longer than the fish in the image whereas our technique does not have that limitation.

Costa et. al. (Costa et al., 2013) were able to sort fish based upon size, gender and skeletal anomalies using external shape analysis. First,they used the Canny Operator in MATLAB (Canny, 1986)to create a binary image. The Canny operator smooths the image through Gaussian convolution and applies a 2D first derivative operator to highlight regions with edges. The next step was to create 200 equally spaced points along the outline of the marine animal. The shape of each fish was then analyzed by elliptic Fourier analysis (EFA) on the coordinates of each outline point. EFA is based on Fourier decompositions of the incremental changes in each of the x and y coordinates (Costa et al., 2011). The limitation of this is that this binary image is specific to a species and sensitive to variations within a species.

Ogunlana et. al. (Ogunlana et al., 2015) extracted fish sizes like the body length and width and the five fin lengths; namely anal, caudal, dorsal, pelvic and pectoral. Then, used support vector machines for species classification with a 78.59% accuracy, which was significantly higher than what was obtained for artificial neural networks, k-nearest neighbour and k-means clustering-based algorithms for the same dataset. This approach does not take into account the color and texture features of the marine animal.

Hasija et. al. (Hasija et al., 2017) use image sets to classify fish species using graph embedding discriminant analysis unlike state-of-the-art methods which operate on single images. Multiple views of the fish, as in our approach, might help in a better classification of the fish's species. However, their algorithm is not immune to distortion caused by noisy images which have a classification accuracy of 76 %.

Table 1: Distribution of collected images

label	cod	crab	halibut	cusk	lobster
count	724	503	913	459	631

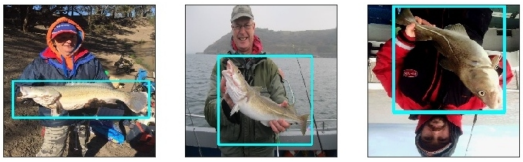

Figure 1: Axis-aligned bounding boxes

3 PROPOSED METHODOLOGY

3.1 Training Data Collection

Static images of marine animals for five species namely Jonah crab, lobster, halibut, cod and cusk were culled from the internet (Table 1). The first stage was to manually draw axis-aligned bounding boxes around the target object (its region of interest or ROI) with the *labelImg* software annotation tool(Tzutalin, 2015) in the images (Figure 1). Then, the second stage was to manually draw rotatable-bounding boxes(Liu et al., 2017) around the object's ROI with the *roLabelImg* software annotation tool.(Cgvict, 2017) (Figure 2).

3.1.1 Image Augmentation

This culled dataset was augmented using the *imgaug* (Aleju, 2015) software image augmentation library. The following image augmentations were performed: randomly rotated horizontally and vertically; affine transformations like image translation from -10% to 10%; rotations from -45° to 45 °; images sharpened with pixel intensity multiplicative ratios from 0.75 to 1.5; image brightness changed for each RGB channel by adding pixel intensity from -10 to 10 and contrast normalization ratios ranging from 0.9 to 1.10.

As part of the data preparation for the training, the images were re-sized to 224×224 pixels for input into the CNN. (Figure 3).

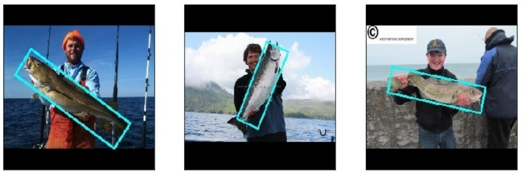

Figure 2: Rotated bounding boxes

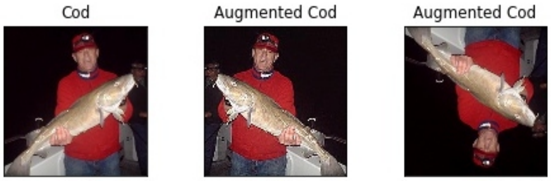

Figure 3: Example images generated for each Cod from image augmentation. Augmentation such as random image flipping,rotation and translation were applied to the original image.

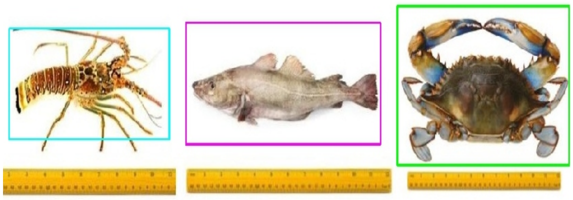

Figure 4: Automatic Size estimation using localization and ruler detection. Ruler detection gives the unit length in pixels.

3.2 MODEL

3.2.1 Architectures

Acquiring enough labelled data for image classification using supervised CNN can be a challenge. This can be mitigated by re-using models trained on different image sets (Yosinski et al., 2014). The CNN model architectures used are described next.

VGG16 and VGG19: These are deep convolutional networks trained by the Visual Geometry Group proposed by Simonyan and Zisserman (Simonyan and Zisserman, 2014). Their network uses 3 × 3 convolutional layers stacked on top of each other. The first step is a convolution of the image. Then, the image size is reduced through down sampling (max pooling). This alternates until the two layers become fully connected. The 16 and 19 in VGG16 and VGG19 refer to the number of convolutional layers in each of the networks, respectively.

Residual Network: Resnet(He et al., 2016) allows the addition of hundreds of layers to a network and is still able to achieve good performance compared to VGG. Residual networks use residual mapping or skip connections to a deeper version of the network. At each layer, Resnet is implemented as shown in eq.1

$$y = f(x) + x \tag{1}$$

such that *f(x)* is the convolution or batch normalization layers and x is the skip connection that allows the gradient to pass backwards, directly. Theoretically, the gradient could skip over all the intermediate layers and reach the bottom one without being diminished. Residual mappings therefore assist in avoiding the vanishing gradient problem that occurs in deep CNNs. Residual networks also use batch normalization layers which are intermediate normalization layers. Theses layers address the problem of vanishing and exploding gradients.(Bengio et al., 1994)(Glorot and Bengio, 2010).

MobileNet: MobileNet (Howard et al., 2017) uses a 3 × 3 depth-wise separable convolution which uses less computations than standard convolutions with only a small reduction in accuracy. Depth-wise separable convolutions are made up of two layers: depth-wise convolutions and point-wise convolutions. In depth-wise convolutions, filters are applied to each input channel. Point-wise convolution is a 1 × 1 convolution used to create linear combinations of the output of the depth-wise layer. This two-step method reduces the computation effort and learning model size. The depth-wise convolutions filter the input channels but do not combine them to create new features whereas the point-wise convolutions generate new features.

3.2.2 Feature Extraction

For feature extraction pre-trained networks of VGG16 (Simonyan and Zisserman, 2014), VGG19 (Simonyan and Zisserman, 2014), Resnet (He et al., 2016) and MobileNet (Howard et al., 2017) were used. The weights used for the pre-trained networks are those from the ImageNet dataset. The deeper layers of the pre-trained networks were trainable to improve the accuracy. The pre-trained networks branch out into a regression head and a classification head.

3.2.3 Size Estimation

The width and height of the bounding box is a measure of the marine animal size but it still needs to be scaled to its actual size. To achieve this, a ruler is inserted in the image field of view. We detect the ruler (Figure 4)(Konovalov et al., 2017) in the image background and determine the length of a pixel and scale the width and height of the bounding box to this.

3.2.4 Model Implementation

Figure 5 shows the architecture of our learning model. The classification head contains a fully connected layer of 2048 neural network units followed by another fully connected layer of 5 units. The output of the classification head are neural network units equal in number to the number of distinct marine animal species considered where each neural network unit

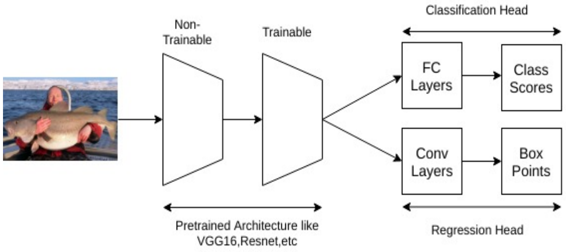

Figure 5: Convolutional layered architecture with the regression and classification heads for simultaneous localization and classification of images.

gives the probability that the image belongs to that species.

The regression head contains a convolution layer with 1024 3×3 filters followed by a 4×4 max pooling layer. The final layer of the regression head is a 1×1 convolution layer. The output of the regression head are localization coordinates for the object. In the case of axis-aligned bounding boxes, the coordinates are (x_1, y_1, x_2, y_2) where (x_1, y_1) is the lower left corner and (x_2, y_2) the upper right corner of the bounding box.

The size of the marine animal can also be estimated by creating a rotatable bounding box around it. A rotatable bounding box fits an object oriented at an arbitrary angle to the horizontal, better, than an axis-aligned one. The regression coordinates for a rotatable bounding box are (x_c, y_c, h, w, θ) where x_c, y_c are the coordinates for the center of the bounding box, h, w are its height and width, respectively, and θ is its orientation relative to the horizontal.

3.2.5 Ensemble Architecture

Ensemble learning uses multiple models to attain better predictive performance than that obtained by any one model.(Zhou, 2009). The classification performance can be increased by combining the predictions of multiple weak models instead of training a single strong one.

Different ensemble architectures were used to process the outputs from the regression and classification heads.

The results from the classification heads of each of the CNN architectures (VGG16, VGG19, Mobilenet and Resnet) were concatenatated. Then, the resulting 20-dimensional vector was sent to the ensemble CNN for classification. The ensemble CNN has a fully connected layer of 50 neural network units followed by another fully connected layer of 5 units (equal to the number of species).

Similarly, the output from the image localization heads of each of the CNN architectures (VGG16, VGG19, Mobilenet and Resnet) were concatenated. The resulting vector (20-dimensional for the axis-aligned and 25-dimensional for the rotatable boxes) was sent to the ensemble CNN for object localization. The ensemble CNN has a fully connected layer of 50 neural network units followed by another fully connected layer containing neural network units equal to the number of localization parameters (4 for axis-aligned and 5 for rotatable boxes).

3.3 TRAINING

3.3.1 Losses

Loss is used in the training to obtain the best weights for a model. Loss is optimized (minimized) in the training by adjusting the CNN weights. The cross-entropy loss (eq.2) was used for the classification head and the mean squared error loss (eq.3, 4) for the regression head. With the cross-entropy loss:

$$ce(y, \hat{y}) = -\sum_{i=1}^{\mathrm{n}} y_i \log(\hat{y}_i) \tag{2}$$

y_i is the ground truth label of the image and $\hat{y}_i$ is the predicted species score.

For the mean square error loss for the axis-aligned bounding boxes:

$$\begin{aligned} mse_{ax} = -\sum_{i=1}^{\mathrm{n}} ((\hat{x}_1^i - x_1^i)^2 + (\hat{x}_2^i - x_2^i)^2 + \\ (\hat{y}_1^i - y_1^i)^2 + (\hat{y}_2^i - y_2^i)^2) \end{aligned} \tag{3}$$

(x_1^i, y_1^i) and (x_2^i, y_2^i) are the ground truth coordinates of the lower left and upper right corners of the ith axis-aligned box and $(\hat{x}_1^i, \hat{y}_1^i)$ and $(\hat{x}_2^i, \hat{y}_2^i)$ are the predicted coordinates of the lower left and upper right corners of ith bounding box.

Eq. 4 is the mean square error loss for the rotatable bounding boxes.

$$\begin{aligned} mse_r = -\sum_{i=1}^{\mathrm{n}} ((\hat{x}_c^i - x^i{}_c)^2 + (\hat{y}_c^i - x^i{}_c)^2 + \\ (\hat{h}^i - h^i)^2 + (\hat{w}^i - w^i)^2 + (\hat{\theta}^i - \hat{\theta}^i)^2) \end{aligned} \tag{4}$$

(x_c, y_c) are the coordinates of the center of the *ith* box. (h,w,θ) are the height,width and angle of the box relative to the horizontal. $(\hat{x}_c, \hat{y}_c)$ are the predicted coordinates of the center of the *ith* box.($\hat{h}$,$\hat{w}$,$\hat{\theta}$) are the predicted height,width and angle of the box relative to the horizontal.

The loss function used for training is the sum of the mean squared error from the regression head and the entropy loss error from the classification head. Forty epochs were trained. The evaluation of the resulting model is discussed in the next section.

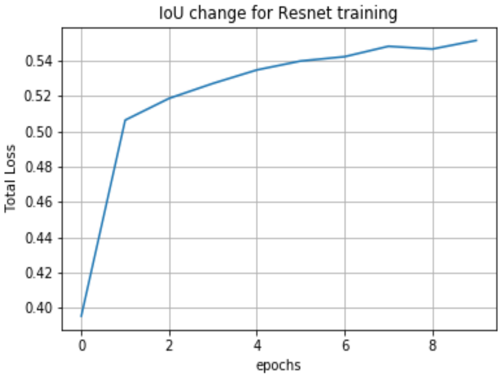

Figure 6: IoC metric change for Resnet architecture. IoC is intersection over union. The IoC increases during training time and converges to 0.7. A higher IoC score indicates better localization accuracy.

4 MODEL EVALUATION

4.1 Methodology

The intersection over union (IoU) was used to evaluate the accuracy of the axis-aligned bounding boxes. The IoU is defined as the ratio of the intersection (overlap) area between two bounding boxes and the area of union of the two bounding boxes.

Figure 6 shows the increase in IoU value while training. Both these trends are indicative of a good model.

Five-fold cross validation (Kohavi et al., 1995) was used for model evaluation. The original training dataset was divided into five folds. At each iteration, four folds were used for training and the fifth for testing/evaluation. The data augmentation described earlier was performed on the training set but not on the test/evaluation set.

4.2 Results

The height and width of the object in pixel lengths were annotated in the images. To compare the performance between axis-aligned (Table 2) and rotatable (Table 3) bounding boxes, the mean absolute error in pixels of the annotated height and width and the predicted height and width was calculated (Table 4)

Table 4 compares mean absolute error in predicted height and width (in pixels) between the two types of bounding boxes. The rotatable bounding boxes have notably lower mean absolute errors than the axis-aligned ones. This suggests rotatable bounding boxes are a better measure of the target height and width.

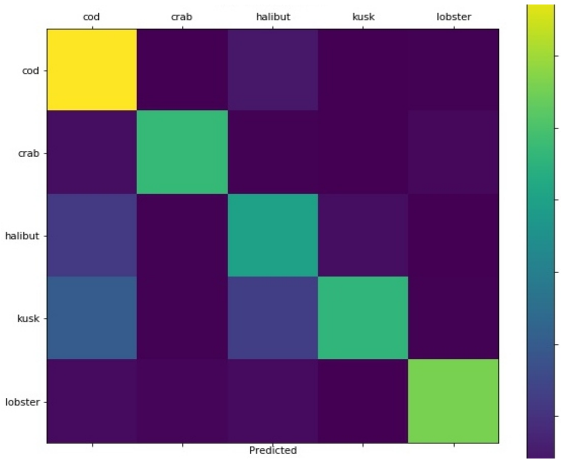

Figure 7: Confusion matrix for the Resnet classification. The diagonal cells show greater values indicating better classification accuracy for each label.

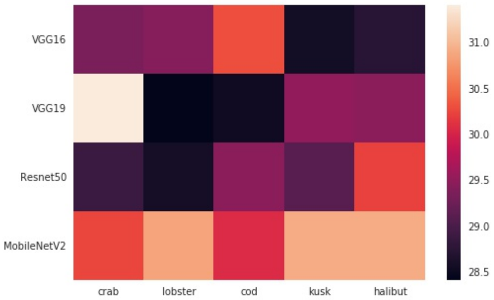

Figure 8: Weights assigned to the CNN architectures for prediction of each label in the species classification

Figure 7 illustrates the confusion matrix for species classification. The classifier misinterprets some cusk images as cod. However, that is not unexpected as cusk are a type of cod.

Figure 8 shows the weights learned by the ensembling architectures for classification of fish species. Darker colors depict higher weights for the output of a particular ensembling architecture. Note, the confidence in prediction of cusk and halibut is high for the VGG16 ensembling architecture prediction.

Figure 9 shows the weights learned by the ensembling architecture for localization species. Again, darker colors depict a higher weight for the output of a particular method. The ensemble architecture gives a higher weight to MobileNetV2 because it shows a better localization accuracy compared to VGG19 and Resnet.

The ensemble classification accuracies and localization metrics are better than individual CNN architectures for both axis-aligned and rotatable bounding

Table 2: Mean localization accuracy, intersection over union and classification accuracy of CNN architectures using axis-aligned bounding boxes. The classification accuracy of the ensembled architecture is 81% which is better than classification accuracy of the individual CNNs. The ensembled CNN gives a test IoU score of 0.57 which is marginally better than the individual CNN IoU scores.

Model	Localization		IoU		Classification	
	Train	Test	Train	Test	Train	Test
VGG16	0.92	0.78	0.79	0.56	0.98	0.76
VGG19	0.93	0.79	0.79	0.56	0.98	0.75
Resnet	0.80	0.77	0.58	0.54	0.73	0.76
MobileNet	0.85	0.80	0.65	0.57	0.59	0.59
Ensembled	0.93	0.81	0.78	0.57	0.99	0.81

Table 3: Mean localization accuracy and classification accuracy of CNN architectures using rotatable bounding boxes. The classification accuracy of the ensembled architecture is 83% which is better than classification accuracy of the individual CNNs. The ensembled CNN gives a localization accuracy of 0.80 which is marginally better than the individual CNN localization accuracy scores.

Model	Localization		Classification	
	Train	Test	Train	Test
VGG16	0.93	0.76	0.94	0.76
VGG19	0.94	0.79	0.90	0.73
Resnet	0.67	0.6	0.70	0.73
MobileNet	0.87	0.76	0.58	0.56
Ensembled	0.95	0.80	0.99	0.836

Table 4: The mean absolute error (in pixels) for width and height using axis-aligned and rotatable bounding boxes. The rotatable bounding boxes show lower mean absolute error in the predicted height and width.

Model	Axis-Aligned		Rotatable Boxes	
	height	width	height	width
VGG16	39.72	59.72	20.99	16.01
VGG19	39.98	61.04	18.94	16.69
Resnet	40.04	65.61	27.96	22.03
MobileNet	40.58	50.58	20.64	16.78
Ensembled	40.57	56.95	17.19	13.00

boxes. The ensemble classification had some value.

4.2.1 Visualization

Figure 10 shows heat map images that portray the activation of the convolutional layers. The final layer shows higher 'temperatures' around the fish outline indicating the regions that contribute more to the classification.

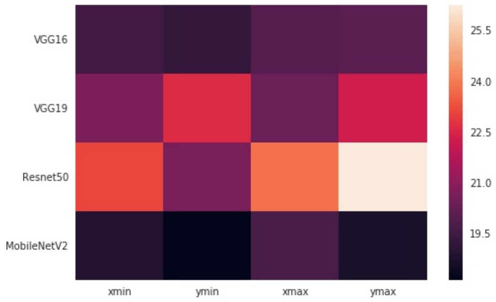

Figure 9: Weights assigned to CNN architectures for prediction of the corners of bounding boxes in localization. xmin and ymin are left bottom coordinates and xmax and ymax are top right coordinates

5 CONCLUDING REMARKS

This paper reports on work that considers and proves the viability of using pattern recognition towards automating the species classification and sizing of marine animals. The pre-trained weights used in the convolution neural network were based on those used in ImageNet which contained lobster, crab, and several types of fish though not the cod, cusk and halibut that were of interest.

The work evaluates axis-aligned and rotatable bounding boxes for marine animal classification and localization aimed at their size estimation in static images. Based on an analysis of the mean absolute error in bounding box heights and widths, it was observed that rotatable bounding boxes perform notably better. Therefore, rotatable bounding boxes yield a better estimate of the marine animal height and width.

Future work builds on the achievements to date to

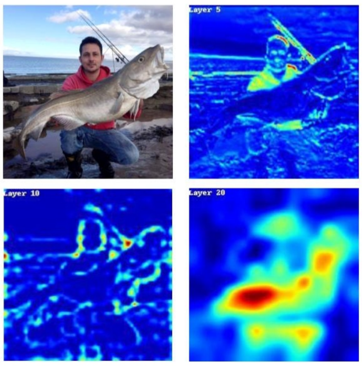

Figure 10: Activation of convolutional layers. Upper left: The original image, Upper right: activation from layer 5, Lower left: activations from layer 10, lower right: activations from layer 20. Activations from layer 2 show higher temperatures around the object area indicating the regions that contribute more to the prediction.

collect and prepare more training data for the specific species of interest to increase the model accuracy. As well, work is underway to build a marine animal presentation system for an image capture system that is integrated with the learning tools developed to date. This will be integrated into a marine animal processing plant for testing in an operationally relevant environment.

REFERENCES

Aleju (2015). aleju/imgaug.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698.

Cgvict (2017). cgvict/rolabelimg.

Costa, C., Antonucci, F., Boglione, C., Menesatti, P., Vandeputte, M., and Chatain, B. (2013). Automated sorting for size, sex and skeletal anomalies of cultured seabass using external shape analysis. *Aquacultural engineering*, 52:58–64.

Costa, C., Antonucci, F., Pallottino, F., Aguzzi, J., Sun, D.-W., and Menesatti, P. (2011). Shape analysis of agricultural products: a review of recent research advances and potential application to computer vision. *Food and Bioprocess Technology*, 4(5):673–692.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.

Hasija, S., Buragohain, M. J., and Indu, S. (2017). Fish species classification using graph embedding discriminant analysis. In *Machine Vision and Information Technology (CMVIT), International Conference on*, pages 81–86. IEEE.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Hsieh, C.-L., Chang, H.-Y., Chen, F.-H., Liou, J.-H., Chang, S.-K., and Lin, T.-T. (2011). A simple and effective digital imaging approach for tuna fish length measurement compatible with fishing operations. *Computers and Electronics in Agriculture*, 75(1):44–51.

Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada.

Konovalov, D., Domingos, J., Bajema, C., White, R., and Jerry, D. (2017). Ruler detection for automatic scaling of fish images. In *Proceedings of the International Conference on Advances in Image Processing*, pages 90–95. ACM.

Larsen, R., Olafsdottir, H., and Ersbøll, B. K. (2009). Shape and texture based classification of fish species. In *Scandinavian Conference on Image Analysis*, pages 745–749. Springer.

Liu, L., Pan, Z., and Lei, B. (2017). Learning a rotation invariant detector with rotatable bounding box. *arXiv preprint arXiv:1711.09405*.

Ogunlana, S., Olabode, O., Oluwadare, S., and Iwasokun, G. (2015). Fish classification using support vector machine. *African Journal of Computing & ICT*, 8(2):75–82.

Rathi, D., Jain, S., and Indu, D. S. (2018). Underwater fish species classification using convolutional neural network and deep learning. *arXiv preprint arXiv:1805.10106*.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Tzutalin (2015). tzutalin/labelimg.

White, D., Svellingen, C., and Strachan, N. (2006). Automated measurement of species and length of fish by computer vision. *Fisheries Research*, 80(2-3):203–210.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328.

Zhou, Z.-H. (2009). When semi-supervised learning meets ensemble learning. In *International Workshop on Multiple Classifier Systems*, pages 529–538. Springer.