Convolutional Neural Networks for Counting Fish in Fisheries Surveillance Video

Mr. G. French[1]

Dr. M. H. Fisher[1]

Dr. M. Mackiewicz[1]

Dr. C. L. Needle[2]

[1] University of East Anglia
Norwich, UK

[2] Marine Laboratory, Marine Scotland
PO Box 101, 375 Victoria Road,
Aberdeen, UK

Abstract

We present a computer vision tool that analyses video from a CCTV system installed on fishing trawlers to monitor discarded fish catch. The system aims to support expert observers who review the footage and verify numbers, species and sizes of discarded fish. The operational environment presents a significant challenge for these tasks. Fish are processed below deck under fluorescent lights, they are randomly oriented and there are multiple occlusions. The scene is unstructured and complicated by the presence of fishermen processing the catch. We describe an approach to segmenting the scene and counting fish that exploits the N^4-Fields algorithm. We performed extensive tests of the algorithm on a data set comprising 443 frames from 6 belts. Results indicate the relative count error (for individual fish) ranges from 2% to 16%. We believe this is the first system that is able to handle footage from operational trawlers.

1 Introduction

This paper describes an ongoing development of a computer vision tool that augments an existing CCTV system that is installed on board fishing trawlers and whose purpose is the monitoring of the fish catch discard. We describe the regulatory environment and the motivation for this project in more detail in Section 2. A key element of the system is a video camera overlooking a conveyor belt, where the fish are processed and at the end of which only the fish to be discarded remain (see Fig. 1, left column). Currently, the footage is analysed by human experts who manually count the discarded fish, measure their size and identify their species. The objective of the project is to reduce the viewers' workload as much as possible by automating this tedious and expensive procedure.

An initial requirement leading to the above objective is to detect and count fish entering the discard shoot The operational environment presents significant challenges for this task and while in Section 3 we review a body of previous work, no current systems can deal with the multiple occlusions that arise during periods of high throughput. Our approach to fish segmentation and counting utilises convolutional neural networks (CNNs) [21, 28, 33]

and comprises the following steps. The first step involves foreground segmentation (fish vs. non-fish). The output of this step requires further refinement as many frames contain a large number of overlapping fish. Therefore, the output from the first step is further segmented by a CNN-based edge detector followed by the Watershed algorithm [7]. The CNNs in both steps are based on the N^4-Fields approach [21]. The details of the fish segmentation and counting algorithm are described in Section 5.

We present extensive tests of the algorithm in Section 6. The fish relative count error ranges from 2% to 16%, which is a promising result in this ongoing work as the required figure for this task is 10%.

2 Motivation

The EU Common Fisheries Policy aims to ensure the exploitation of living aquatic resources is constrained to provide sustainable economic, environmental and social conditions. The annual setting of total allowable catch (TAC) quota has been a key feature of fishing controls over the last decade [16]. Mismatches between the TAC and estimates of the actual catch taken have prompted regulatory bodies to incentivise fishermen to document [12, 27] and be accountable for total catches, including discards, with exemption from certain regulations and increased quotas. This is accomplished through remote electronic monitoring (REM). Vessels that adopt the Catch Quota Monitoring System (CQMS) are fitted with camera and telemetry systems that provide a record of fishing activity. Pilot studies in the UK concluded that use of REM technology for CQMS is feasible; and this view is supported by experiences in the US, Canada and New Zealand [9, 30, 32, 34].

A significant proportion of the Scottish demersal fishing fleet are equipped with REM technology. A CCTV system developed by Archipelago Marine Research [1] logs video from four cameras continuously during fishing trips lasting about a week. Fishing nets are emptied into a hopper and moved by conveyor to a sorting area. Non-viable fish (e.g. small specimens, non-commercial species) and debris are left on the belt and returned via a discard chute. A view of the belt near the discard area is critical to CQMS. A sample of video frames is shown in Fig. 1, left column. The environment is challenging with long periods of inactivity punctuated by high throughput. The fish are randomly oriented and often occluded by other fish and debris. Fishermen are reluctant to adopt mechanical arrangements that distribute specimens more evenly as they fear this will affect throughput. Hands and arms of crew members are evident in many frames.

A team of Marine Scotland compliance officers monitor the accuracy of documented numbers, sizes and species of fish caught by sampling each vessel's record [33]. Manually reviewing the video footage to classify, count and measure the catch is a tedious and costly task that represents a bottleneck in CQMS.

3 Background

The first attempts to automatically grade and sort fish were reported in the 1980s by Tayama et al. [43], who used shape descriptors derived from binary silhouettes to discriminate between 9 fish species with 90% accuracy. Further work refined approaches for classification

[1] http://www.archipelago.ca

of species using primarily shape and colour features [4, 24, 25, 40, 41, 44]. All these systems require fish to be presented individually and engineer solutions that enforce this constraint.

Our approach is inspired by contemporary work using machine learning as recent research has delivered impressive results on object recognition and localization, driven by the availability of very large labelled data sets such as LabelMe [36] and ImageNet [13]. We focus particularly on convolutional neural networks (CNNs), since these have produced state of the art image classification results [23, 26, 28, 38, 45]. CNNs demonstrate a large learning capacity and can easily be trained using currently available parallel GPU architectures and open source tools [5, 6, 14].

A good overview of approaches for counting objects in images can be found in [29]. They tend to fall within four categories: counting by detection and localization; counting by regression; counting by segmentation; and finally approaches that exploit density functions. Many of these approaches are not currently suitable for our problem as they operate on single images, with no current avenue for tracking the movement of individual objects in a video. Segmentation methods are often broadly based on pixel classification, edge detection or super-pixel based approaches. The former, (e.g. [10]) classify a pixel given its surrounding patch. The resulting pixel classes are used to divide the image into labelled regions. Traditional CNN-based pixel classifiers extract one patch per pixel at training and prediction time. This is computationally wasteful since the convolutional nature of the layers of a CNN mean that many redundant computations will be performed. This is addressed in [22] where complete images are processed in one go resulting in a significant speed-up. Edge detectors generate an edge probability (and sometimes direction) map that can be used to divide the image into regions. Superpixel algorithms such as SLIC [3] group pixels into small continuous regions called superpixels, that can be merged into regions using algorithms such as Normalized Cuts [37] or CNN based approaches [18]. Accurate full segmentation appears to be the most promising avenue for addressing the project requirement of isolating individuals for classification and measurement.

4 The Dataset

Our training and testing footage was captured at VGA resolution from an analogue camera and recorded digitally. It consists of 52 videos from 12 different conveyor belts. Footage from 6 belts was unusable due to permanent structures occluding large portions of the belt, having insufficient field of view or having the camera lens frequently clouded by spatter. A segmented dataset was required in order to train and evaluate the segmentation system. It consists of still frames extracted from segments of video where the belt was moving in order to avoid acquiring multiple frames with nearly identical content. Ground truth segmentations were prepared manually using a browser-based image labelling tool hat allows the user to annotate images with polygonal labels. Our tool can operate as both a plugin for IPython Notebook [35] or within a website and is available as an open-source project [2]. The resulting dataset is described in Table 1.

Fixed camera positioning results in the belt occupying a constant region in the footage. This region, along with an estimate of its physical size were used to derive a perspective transformation that was used to transform these regions into rectilinear space with a constant physical size to pixel ratio. The resulting images were used for training as described in Section 5.

Belt	# frames	Resolution in rectilinear space	Total fish identified	# empty frames
A	83	280x210	524	7
B	42	197x158	373	3
C	44	443x256	733	0
D	84	187x175	509	8
E	147	329x224	1010	54
F	43	273x186	462	3

Table 1: The Dataset

5 Approach

Our segmentation pipeline consists of two steps: firstly, foreground segmentation is used to separate salient regions of fish from background and non-salient objects (conveyor belt, detritus, occlusions from people, etc.). Then continuous regions of foreground pixels, often comprising several fish, are separated from one another so that they can be properly counted and classified in later stages. The most promising approaches for foreground segmentation were CNN-based approaches, in particular the N^4-Fields [21] image transformation algorithm. Rather than operating as a classifier or a regressor that directly generates a result for each pixel in the output image, it uses the CNN to generate codeword vectors for the output image at a reduced resolution (half resolution in [21]). These codewords are used to select small output image patches from the training set that are averaged together to produce the predicted output image. In [21] this approach is used for both edge detection with the Berkeley Segmentation Dataset [31] (we use its edge detection capabilities in the fish separation phase described below) and pixel prediction with the DRIVE dataset [39].

The N^4-Fields training procedure is as follows. **(1)** Extract overlapping patches from the output images in the training set (segmentation masks for foreground segmentation). As in [21] we use 16x16 patches, however we use a stride of 4. **(2)** Apply PCA to reduce the dimensionality; as in [21] we retain 16 dimensions. These PCA codewords act as a proxy for the output patch. **(3)** Train a CNN based regressor to map input patches – whose size depends on the CNN architecture; 48x48 for our network – to the PCA codeword of the output map that resides at the centre of the input patch. **(4)** Run the input patches from the training set through the CNN to get its approximation of the PCA codewords and build a dictionary that maps them to the corresponding output patches. The steps for transforming an image are as follows. **(1)** Create an output image of the appropriate size initialised to zero. **(2)** extract overlapping patches from the input image and apply the CNN to generate a codeword vector for each input patch. **(3)** Use a nearest neighbours algorithm to select N closest output patches from the dictionary generated in training step 4 ($N = 25$ for foreground segmentation). **(4)** Cumulatively add the output patches to the output image and scale as appropriate so that each output pixel is the mean of the corresponding pixels within the output patches that cover it.

Our N^4-Fields based foreground segmenter is trained to map input patches to foreground masks. It generates a real valued output image that gives the probability that each pixel is a foreground (fish) pixel. This probability map is smoothed with a Gaussian filter with $\sigma = 4.5$, after which the pixels whose values are ≥ 0.5 are marked as foreground pixels. We used N^4-Fields in preference to a pixel classifier as it was slightly more accurate.

Once we have isolated foreground objects we use the N^4-Fields algorithm again, this time to predict edge maps. As in [21], we use the *alternative encoding* that was inspired by [15], in which the output vector whose dimensionality is reduced by PCA is a bit-vector of pairwise pixel differences obtained from a label map rather than an edge map. For each predicted

codeword feature vector, the 64 nearest neighbours are selected from the codeword-patch dictionary and are blended with their contributions weighted by inverse distance.

Our separation segmentation process steps are: **(1)** predict an edge map using the above and smooth it with a Gaussian filter with $\sigma = 2$. **(2)** The foreground mask generated by the foreground segmenter is split by discarding pixels whose edge strengths are above the edge threshold with $t = 0.045$. The remaining foreground regions are assigned unique labels. **(3)** The Watershed [7] algorithm is applied to the edge strength map to spread the region labels to their boundaries. **(4)** Regions with area smaller than 100 pixels are discarded; those remaining are the predicted individual fish.

We tested a variety of neural network architectures for both segmentation by pixel classifier and N^4-Fields. We found that deep architectures that use small convolution kernels tend to perform better, as suggested in [38]. Our system was implemented using the Lasagne deep neural network library [14] and Theano [5, 6], running on the Anaconda Python distribution [1].

6 Evaluation

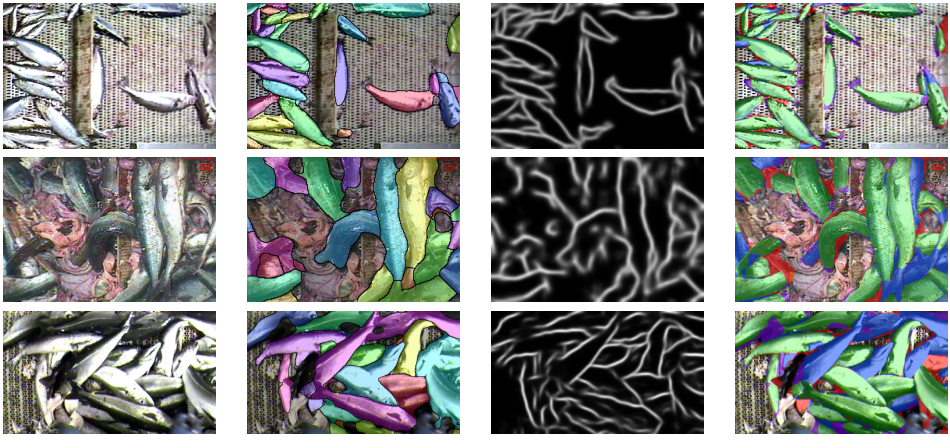

Figure 1: Segmentation results. Columns from left to right: original image, coloured predicted labels, predicted edge map, accuracy. In the accuracy images, red: false negative in foreground segmentation; purple: false positive in foreground segmentation, green: overlap between matching predicted and ground truth label pair, blue: no match between predicted and ground truth labels but agreement in foreground segmentation. Rows are from belts E, F, and C, respectively. Row 1 shows a successfully segmented frame. Rows 2-3 show more challenging frames in which the fish are mixed with guts, mostly obscuring the belt.

Our evaluation approach is the result of adapting the one used in the PASCAL VOC challenge [17] so we evaluate the accuracy of the foreground segmenter with precision-recall curves (Fig. 2). The images for each belt were treated as separate datasets and split into 5 folds for 5-fold cross validation. For each split, 4 folds were used as training data, while the 5th was split in half to be used as validation and test data. Example segmentation results can be seen in Fig. 1. Table 2 shows values for accuracy and intersection-over-union (IU) [17]. An evaluation of the accuracy of individual fish prediction is presented in Table 3. It is worth noting that the relative error was computed using the aggregate count in GT and prediction for all the images from a belt; the relative error rates for individual images are higher.

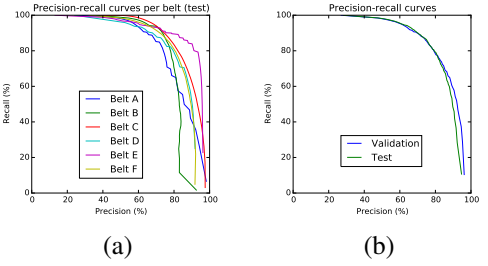

Figure 2: Foreground segmentation precision-recall curves; (a) individual belts; (b) all belts

Belts	Accuracy (%)		IU (%)	
	Val.	Test	Val.	Test
A	93.03	91.98	64.43	60.44
B	88.85	85.67	65.34	63.84
C	86.32	86.19	62.56	68.20
D	91.52	92.3	62.70	65.69
E	97.02	97.70	74.52	67.33
F	88.89	88.49	65.53	67.56
Avg	90.94	90.39	65.85	65.51

Table 2: Foregound segmentation accuracy and IU.

Belt	# fish		# fish rel err (%)		Label IU (%)		Area IU (%)	
	Val	Test	Val	Test	Val	Test	Val	Test
A	292	232	19.18	3.02	49.53	50.67	63.26	59.08
B	178	195	7.87	3.59	50.26	45.73	50.93	50.95
C	340	393	5.29	11.20	45.52	45.31	50.49	53.56
D	243	266	12.35	15.79	54.17	57.26	56.99	62.52
E	544	466	0.0	6.65	76.63	81.52	80.18	83.54
F	222	240	2.25	2.5	50.70	45.47	58.76	47.87
Avg	303.17	298.67	7.82	7.13	54.47	54.33	60.10	59.59

Table 3: Fish separation results. The first pair of columns (*# fish*) gives the GT value for the number of fish in the dataset. The *# fish rel err* column gives the relative error in the predicted number of fish across all images from the given belt; this was the discrepancy between the GT and predicted counts (number of fish found by segmentation) divided by the GT count. The final two pairs of columns evaluate the accuracy of the segmentation by correspondence between predicted and GT fish labels. This is performed by matching predicted and GT fish that overlap in a greedy pairwise fashion according to their area of overlap; the fish with the greatest overlap area are matched, then the remaining fish are matched in the same way until no more matches can be made. The results in the *Label IU* column pair show the IU measure; in this case $\frac{|M|}{|P|+|G|-|M|}$ where M is the set of matched fish labels, P is the set of predicted fish and G is the set of GT fish. The values for *Area IU* are calculated as $\frac{|M_p|}{|P_p|+|G_p|-|M_p|}$ where M_p is the set of pixels where matched fish labels intersect and G_p and P_p are the set of GT and predicted pixels respectively.

7 Conclusions

We have presented results obtained from a computer vision system for automatically quantifying fish in fisheries CCTV video. As far as we are aware, this is the first such system that attempts to operate on footage obtained in an unstructured operating environment; prior systems have required the installation of bespoke equipment to constrain the video acquisition thereby simplifying processing. The aggregate fish counts obtained were sufficiently accurate to meet the project requirements for 3 out of 6 belts; belts A, C and D which yielded high count errors presented particularly challenging conditions. The use of aggregation contributed to the good accuracy figures since errors in individual frames cancel one another out.

8 Acknowledgement

This work was funded by The Scotish Government, project ref: SYS/001/14, Automated Image Analysis for Fisheries CCTV. We would like to thank Marine Scotland fisheries scientists Rui Catarino, Rachel Kilburn and compliance officer Norman Fletcher for their help in delineating individual fish in training video frames.

References

[1] Continuum Analytics - Anaconda Python. http://store.continuum.io/cshop/anaconda/.

[2] UEA Computer Vision - Image Labelling Tool. http://bitbucket.org/ueacomputervision/image-labelling-tool.

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

[4] M. K. S. Alsmadi, K. B. Omar, S. A. Noah, and I. Almarashdah. Fish recognition based on the combination between robust feature selection, image segmentation and geometrical parameter techniques using artificial neural network and decision tree. *International Journal of Computer Science and Information Security*, 2(2):215–221, 2009. URL http://arxiv.org/abs/0912.0986.

[5] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[6] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral Presentation.

[7] S. Beucher and F. Meyer. The morphological approach to segmentation: the watershed transformation. Mathematical morphology in image processing. *Optical Engineering*, 34:433–481, 1993.

[8] G. Bradski. OpenCV. *Dr. Dobb's Journal of Software Tools*, 2000.

[9] CEFAS. Catch quota pilot project with remote electronic sensing: North Sea Cod, 2010. URL http://www.cefas.defra.gov.uk/media/361723/catch-quota-pilot-with-rem-application.doc.

[10] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2843–2851. Curran Associates, Inc., 2012.

[11] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[12] J. Dalskov and L. Kindt-Larsen. Final report: Fully documented fishery. Technical Report 204-2009, DTU Aqua, 2009. URL http://www.aqua.dtu.dk/english/~/media/institutter/aqua/publikationer/forskningsrapporter_201_250/204_09_final_report_of_fully_documented_fishery.ashx.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition (CVPR). IEEE Conference on*, pages 248–255, June 2009. doi: 10.1109/CVPR.2009.5206848.

[14] S. Dieleman et al. Lasagne. http://github.com/Lasagne/Lasagne.

[15] P. Dollár and C. L. Zitnick. Fast edge detection using structured forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014. URL http://arxiv.org/abs/1406.5549.

[16] European Union. Council Regulation (EC) No 2371/2002 of 20 December 2002 on the conservation of sustainable exploitation of fisheries resources under the Common Fisheries Policy. *Official Journal of the European Communities*, L 358, 2002.

[17] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) challenge. *International Journal of Computer Vision*, 88 (2):303–338, 2010.

[18] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35 (8):1915–1929, 2013.

[19] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Proceedings of the 13th Scandinavian Conference on Image Analysis*, LNCS 2749, pages 363–370, Gothenburg, Sweden, June-July 2003.

[20] L. Fiaschi, R. Nair, U. Koethe, and F.A. Hamprecht. Learning to count with regression forest and structured labels. In *Pattern Recognition (ICPR), 21st International Conference on*, pages 2685–2688, November 2012.

[21] Y. Ganin and V. S. Lempitsky. N^4-Fields: Neural Network Nearest Neighbor Fields for Image Transforms. In D. Cremers, I. Reid, H. Saito, and M.-H. Yang, editors, *12th Asian Conference on Computer Vision*, LNCS 9004, pages 536–551, Singapore, nov 2014. Springer.

[22] A. Giusti, D. C. Ciresan, J. Masci, L. M. Gambardella, and J. Schmidhuber. Fast image scanning with deep max-pooling convolutional neural networks. Technical Report IDSIA-01-13, Dalle Molle Institute for Artificial Intelligence, 2013.

[23] K. He, X. Zhang, S. Ren, and J. Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. Technical report, Microsoft Research, 2015. URL http://arxiv.org/abs/1502.01852.

[24] B. G Hu, R.G. Gosine, L. X. Cao, and C.W. de Silva. Application of a fuzzy classification technique in computer grading of fish products. *Fuzzy Systems, IEEE Transactions on*, 6(1):144–152, Feb 1998. ISSN 1063-6706. doi: 10.1109/91.660814.

[25] J. Hu, D. Li, Q. Duan, Y. Han, G. Chen, and X. Si. Fish species classification by color, texture and multi-class support vector machine using computer vision. *Comput. Electron. Agric.*, 88:133–140, October 2012. ISSN 0168-1699. doi: 10.1016/j.compag.2012.07.008.

[26] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, 2015. URL http://arxiv.org/abs/1502.03167.

[27] L. Kindt-Larsen, E. Kirkegaard, and J. Dalskov. Fully documented fishery: A tool to support a catch quota management system. *ICES J. Mar. Sci.*, 68(8):1606–1610, 2011.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[29] V. Lempitsky and A. Zisserman. Learning to count objects in images. In J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1324–1332. Curran Associates, Inc., 2010.

[30] Marine Scotland. Report on Catch Quota Management using Remote Electronic Monitoring, 2011. URL http://www.scotland.gov.uk/Resource/Doc/1061/0120363.doc.

[31] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[32] H. McElderry. At-sea observing using video-based electronic monitoring. Technical report, Archipelago Marine Research Ltd., 2008. URL http://www.afma.gov.au/wp-content/uploads/2010/06/EM_Videobased_07.pdf.

[33] Coby L. Needle, Rosanne Dinsdale, Tanja B. Buch, Rui M. D. Catarino, Jim Drewery, and Nico Butler. Scottish science applications of remote electronic monitoring. *ICES Journal of Marine Science: Journal du Conseil*, 2014. doi: 10.1093/icesjms/fsu225. URL http://icesjms.oxfordjournals.org/content/early/2014/12/15/icesjms.fsu225.abstract.

[34] M. Owen, S. Fairweather, G. Kessels, and B. Christensen. Feasibility of operating remote surveillance systems in a marine environment. Technical report, Department of Conservation, New Zeland, 2003. URL http://www.doc.govt.nz/documents/science-and-technical/DSIS136.pdf.

[35] F. Pérez and B. E. Granger. IPython: a system for interactive scientific computing. *Computing in Science and Engineering*, 9(3):21–29, May 2007. ISSN 1521-9615. doi: 10.1109/MCSE.2007.53. URL http://ipython.org.

[36] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database and web-based tool for image annotation. *Int. J. Comput. Vision*, 77(1-3):157–173, May 2008. ISSN 0920-5691. doi: 10.1007/s11263-007-0090-8.

[37] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[38] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. 2015. URL http://arxiv.org/abs/1409.1556.

[39] J. Staal, M. D. Abràmoff, M. Niemeijer, M. A. Viergever, and B. van Ginneken. Ridge-based vessel segmentation in color images of the retina. *Medical Imaging, IEEE Transactions on*, 23(4):501–509, 2004.

[40] F. Storbeck and B. Daan. Fish species recognition using computer vision and a neural network. *Fisheries Research*, 51(1):11 – 15, 2001. ISSN 0165-7836.

[41] N. J. C. Strachan. Recognition of fish species by colour and shape. *Image and Vision Computing*, pages 2–10, 1993.

[42] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. In *Advances in Neural Information Processing Systems*, pages 2553–2561, 2013.

[43] I. Tayama, M. Shimdate, N. Kubuta, and Y. Nomura. Application of optical sensor for fish sorting. *Refigeration*, 57(661):1146–1150, 1982.

[44] D. J. White, C.D. J. White, C. Svellingen, and N. C. J. Strachan. Automated measurement of species and length of fish by computer vision. *Fisheries Research*, 80:203–210, 2006.

[45] R. Wu, S. Yan, Y. Shan, Q. Dang, and G. Sun. Deep image: Scaling up image recognition. *CoRR*, abs/1501.02876, 2015. URL http://arxiv.org/abs/1501.02876.