Fish Classification in Context of Noisy Images

Adamu Ali-Gombe(✉), Eyad Elyan, and Chrisina Jayne

Robert Gordon University, Aberdeen, Scotland
{a.ali-gombe,e.elyan,c.p.jayne}@rgu.ac.uk

Abstract. In this paper, we analysed the performance of deep convolutional neural networks on noisy images of fish species. Thorough experiments using four variants of noisy and challenging dataset was carried out. Different deep convolutional models were evaluated. Firstly, we trained models on noisy dataset of fishing boat images. Our second approach trained the models on a new dataset generated by annotating fish instances only from the initial set of images. Lastly, we trained the models by synthesizing more data through the application of affine transforms and random noise. Results indicate that deep convolutional network performance deteriorate in the absence of well annotated training set. This opens direction for future research in automatic image annotation.

1 Introduction

Fish detection and recognition is important for conservation agencies, marine live scientist, fishing industry and Governments to maintain fish supply and balance in the ecosystem. Increase in continental reef monitoring and deep sea surveillance has created the need for more imagery analysis. Images are generated by mounted cameras that capture continues data for marine biologist. The rate at which data is generated by underwater cameras, fishing boat cameras, automatic underwater vehicles (AUV) and conveyor belt cameras challenge human manual approach to count and sort dish species. Therefore, image based techniques are now more popular in this domain [1,2,20].

Because of its economic importance, a lot of approaches have been proposed in detection and classification of fishes. Researchers employ specialised software and hardware to monitor the marine eco-system. This has helped them in studying fish species behaviour [20], classifying fishes into different species [3,12], count individual species and also track their movements [8]. To support growing needs of the research community, competitions such as Kaggle[1] and Seaclef/LifeClef[2] provides richly annotated datasets for researchers aiming at pushing the research boundaries.

However, challenges still exist in identifying fish species from these images and videos. In this domain, images obtained here are largely noisy and are affected by illumination. Furthermore, camouflage and presence of multiple

[1] https://www.kaggle.com.
[2] http://www.imageclef.org/lifeclef/2016/sea.

G. Boracchi et al. (Eds.): EANN 2017, CCIS 744, pp. 216–226, 2017.
DOI: 10.1007/978-3-319-65172-9_19

objects in a frame affect segmentation and subsequent localization of object of interest. Hence, successful techniques rely heavily on preprocessing to achieve good results [1,2,8,10].

In this paper, we investigate the performance of state of the art convolutional neural network in context of noisy images. Our hypothesis is that deep learning based methods performances will deteriorate when lacking clean and well labelled set of images. To demonstrate this, we build an experimental framework to test using a challenging and complex set of images provided by kaggle[3]. The rest of the paper is organised as follows: Sect. 2 review related literatures in fish classification and related techniques. Section 3 outline methods employed with datasets and models used in this work. Section 4 discusses the results and evaluations in details. Section 5 contains the final remark.

2 Related Literatures

Fish classification is gradually becoming an interesting area in computer vision. Papp et al. research was motivated by the Seaclef of LifeClef competition to detect and track coral reefs from under water videos and recognize individual whales from images [15]. In recognizing individual whales, they applied segmentation after which SIFT-features (Scale Invariant Transform) and descriptors are generated. SIFT keys offer great resistance to deformation [11]. Image description was based on bag of words representation generated from Gaussian mixture model with a similarity measure calculated using RBF (Radial Basis Function). In 2010, Spampinato et al. applied texture, boundary and shape features in detecting and tracking fish species [20]. This is to assist marine biologist in sieving through massive videos from eco-grid feeds of reefs. Scientist are interested in studying fish behaviours in relation to aquatic movements. In their experiment, features were derived from grey level histograms using Gabor filters and grey level co-occurrence matrix (GLCM). Classification was carried out using discriminant analysis. To track clusters, group trajectories were build by clustering individual trajectories using I-kmeans. Li applied R-CNN (region convolutional neural network) [6] to provide real time detection of fish species [10]. He also demonstrated how segmentation could be achieved based on ROI (region of interest). Subsequently, they employed region proposal network (RPN) [16] which returns the best ROI scores at the ROI pooling layer [9]. This procedure was repeated on the same data set as in [10] and got a MAP of 82.7% which is slightly higher than the previous. But most importantly, this was at the expense of significantly larger training time. Closely related to this, is the work done by Zhang et al. in applying objectness to detect fishes from under water images [22]. Bridget et al. proposed a Haar classifier with a field programmable gate arrays framework for detecting fish species [1].

Established algorithms where also tested on noisy and real videos/images. Boudhane's experiment was based on images the authors acquired from the Baltic sea using AUV [2]. Their idea is to isolate fish from a turbid and noisy

[3] https://www.kaggle.com/c/the-nature-conservancy-fisheries-monitoring.

under water images. The authors made use of Poison-Gaussian theory in denoising and enhancing of image quality. They also applied posterior and log likely-hood probabilities in detecting objects in the images. The goal is to enable marine researchers to monitor underwater marine life through the AUV feed. SIFT, Viola and Jones and Kalman filters were used in detecting and tracking of fish by Ekaterina et al. [8]. Their approach was based on applying techniques on wild videos and those obtained in a controlled environment. Background subtraction technique with some algorithms they developed assist in isolating fish species. The authors reported a 73% accuracy on real videos. And they argued that not all established solutions are applicable to real images although they perform excellently on synthetic datasets. Noisy fish images were classified using SVM by Hossein et al. in [7]. They used Gaussian mix model for background subtraction and kalman filter in tracking fish species. However, they reported a significant drop in the detection accuracy of 40.1% in low quality images compared to 91.7% accuracy in high resolution images.

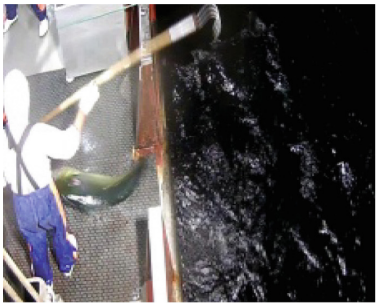
Similar studies were also conducted to investigate the quality of detection, classification and tracking algorithms on fish datasets. Comparison between PCA, SIFT and Viola and Jones performances in detecting and recognizing fish species from images was carried out by Matai et al. in [12]. Dataset used was privately collected and results suggested that more datasets used for training will increase the performance. Ogunlana et al. in [14] used an SVM in classifying fish species with 74.56% accuracy. However, the size of dataset used was small and some assumptions were made which might not hold in other cases. Rodriguez et al. in [17] suggested that artificial radius immune algorithm combined (ARIA) combined with PCA-features and a KNN classifier achieves better results in fish classification. Training was carried out using six species of fishes in formaldehyde. Nguyen et al. in [13] investigated a combination of GMM, kalman filters and frame-differencing in detecting and tracking fish species. Their approach shows robustness to different scenarios considered during experimentation such as speed, clarity of water and appearance than other approaches.

In this paper, we propose an experimental framework that study performance of convolutional neural networks in the context of noisy images. Results obtained shows that when images are well annotated, performances improves.

3 Dataset

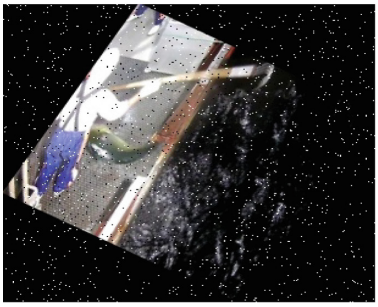
In this paper we used a dataset of images provided by kaggle[4]. It contains 3777 images of fish. The fish categories include Albacore tuna, Bigeye tuna, Yellowfin tuna, Dolphin, Lampris guttatus, Sharks, other categories and images with no Fish, labelled as ALB, BET, YFT, LAG, DOL, SHARK, OTHER and NoF. It is worth pointing out that these images are extracted from video footage of fishing boat. Fish detection in these images is challenging even to the humans. Light variation in images, presence of multiple objects, pulse variation and partial

[4] https://www.kaggle.com/c/the-nature-conservancy-fisheries-monitoring.

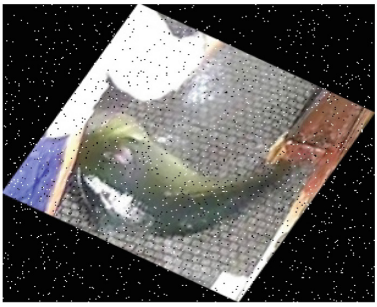


(a) Original Image.

(b) Annotated Image.



(c) Original + Noise Image.



(d) Annotated + Noise Image.

Fig. 1. Sample images

occlusion makes fish recognition very challenging. A sample image from this dataset is shown in Fig. 1a below.

A second dataset was generated from the original images by annotating all the images using an annotation tool[5]. It contains 3777 fish images. Annotation was done by isolating individual fish instances from an image using a bounding box. The bounding box was made big enough to incorporate other surrounding objects to maintain variability in the training set. Annotated instances contain a complete fish with head and tail visible or partially occluded head or tail but not both. It was observed that object view angles and light variation with shadows differs in similar images. This difference was considered visible enough to distinguish adjacent image frames as such no further cleaning was required. Figure 1b shows the result of this process.

A third dataset was created from the previous datasets. Images where generated from both the original and annotated images. The motive behind this is to address the biased nature of image distribution among classes. We also intend to achieve optimum model performance with more data. The new dataset contains 12,275 images across 8 categories. These images were synthesized by applying random noise and affine transform. Images were distorted using varying degree

[5] http://sloth.readthedocs.io/en/latest/.

Table 1. Dataset summary

Dataset	Number of images	Noise	Affine
Original	3777	-	-
Annotated	3777	-	-
Original+Noise	12275	X	X
Annotated+Noise	12275	X	X

of rotation angles (between 15 and 105°) and noise intensities. It is similar to the work done by Dostovistkiy et al. in [4] to generate training samples. This is to create enough distortion to generate distinct images from the originals. The result is shown in Fig. 1c and d. A summary of the datasets is shown in the following Table 1.

4 Methods

This section describes the techniques used in the study. Details of CNN architecture and model initialization are also discussed.

4.1 VGG Network

VGG networks were proposed by the Oxford visual geometry group (VGG) [19]. These networks where 11, 13, 16 and 19 layers deep also known as VGG-11, VGG-13, VGG-16 and VGG-19. These models ranked first and second place in the ImageNet classification challenge in 2014. Their models are one of the most widely used CNN models in image classification today. For the purpose of this experiment, we considered an untrained VGG-16 network. The model contains 5 blocks of 13 convolution layers and 3 fully connected layers. It makes use of a filter size of 3 for all convolution layers. It also employs a max-pooling layer between successive convolution blocks with a unit stride for down sampling. The 3 fully connected layers contains 4096, 4096 and 1000 ReLu activated units respectively (see [19] for details). The VGG-16 network architecture is illustrated below (Fig. 2).

Given that the dataset used has only 8 categories, the final layer was replaced with an 8 way soft-max classifier to suit experiment.

4.2 Transfer Learning

The second model used was proposed by applying transfer learning to the VGG-16 model from a pre-trained network on ImageNet dataset. Transfer learning attempts to reproduce similar results from experience on previous task. It enables a new model to inherit learned parameters from another trained model. This has proved to be effective where training images are scarce [5,21]. The intuition

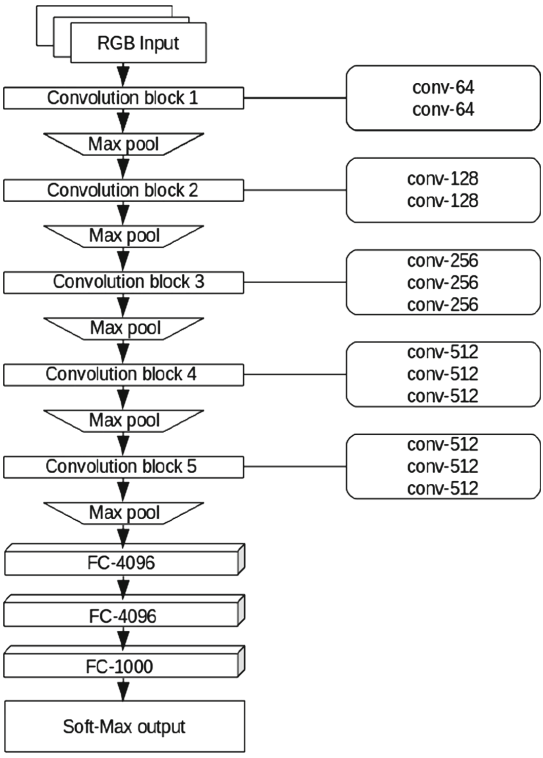


Fig. 2. VGG-16 architecture

behind this is to have a model that have already converged for comparison purposes. Model architecture is exactly the same as the one in Sect. 4.1 but its weights and biases were initialized from learned parameter from training on ImageNet dataset. The motivation behind using transfer learning is that given the size of the dataset, we try to fine tune the network as against learning new features from scratch with the hope that better results could be achieved.

5 Experiments and Results

Both Experiments where ran on NVIDIA DGX-1 machine[6]. Full advantage of the multiple GPU system was taken and this significantly reduced training and test time. Models were implemented using keras[7] with tensorflow[8] back end. Before training was initiated, all images were resized to 224 by 224. This is to accommodate them in the VGG-16 model. Each model was trained using all datasets

[6] http://www.nvidia.com/object/deep-learning-system.html.
[7] https://keras.io/.
[8] https://www.tensorflow.org/.

described above. At the beginning of training, images were shuffled, then split into test and train with 75% of data used for training and the remaining 25% of data for testing. Experiment on VGG-16 was carried out using a learning rate of 10^{-2} over 16 epochs and training was done using stochastic gradient descend with a batch size of 32. A weight decay was chosen as a ratio of learning rate to number of epochs and a momentum of 0.9 was maintained. The settings were to ensure faster convergence of models. Dataset normalization was applied by simply dividing each pixel by 255 for both training and test set where as the original VGG-16 experiment normalize by subtracting the mean pixel value from each pixel. This does not affect model accuracy but training time. We also differ in the choice of weight decay because subsequent experiments revealed that a dynamic weight decay works better than a statically chosen one. Training batch size was significantly lower than the one proposed in VGG-16 because the problem has significantly smaller dataset. Moreover, with smaller batch size shorter gradient updates can be realized. Apart from resizing, no further preprocessing was applied. During testing, we did not employ random crop or other methods as in [19], images are resized and the network is allowed to freely process the images.

Initial training settings for VGG-16 model were maintained for transfer learning as well. However, all the layers of the pre-trained model were fine tuned. No layer was fixed, hence the model was allowed the freedom to update parameter values for better performance similar to the methodology in [18].

Log loss and accuracy metrics were used to evaluate the models. Multi-class logarithmic loss is shown in Eq. (1) below;

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} log(p_{ij}) \tag{1}$$

Where N and M represents the size of the sample and categories respectively, y_{ij} is the correct prediction of sample i being in category j, and p_{ij} is the estimated probability that the sample i belongs to the class j. Logarithm loss penalizes the accuracy of the classifiers on false positives. Probabilities where obtained as predictions from the soft-max layer in the networks. Table 2 below shows the log loss summary of the experiments conducted.

The accuracy of a classifier is the ratio of number of correct prediction from sample to the total number of samples to be predicted. Accuracy is represented as follows;

$$accuracy = \frac{number\ \ of\ \ correct\ \ predictions}{total\ \ number\ \ of\ \ all\ \ cases\ \ to\ \ be\ \ predicted} * 100 \tag{2}$$

Table 2. Models log loss

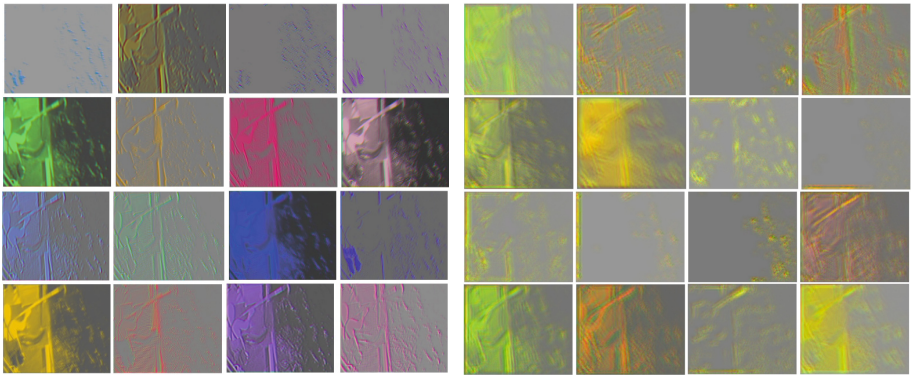
Model	Original	Annotated	Original+Noise	Annotated+Noise
VGG-16	0.54	1.20	0.12	0.38
VGG-16 (transfer)	18.45	27.88	0.10	30.61

Table 3 shows test accuracy of models.

Table 3. Test accuracy of models

Model	Original	Annotated	Original+Noise	Annotated+Noise
VGG-16	97.20%	90.17%	99.38%	98.00%
VGG-16 (transfer)	86.60%	79.81%	99.54%	77.80%

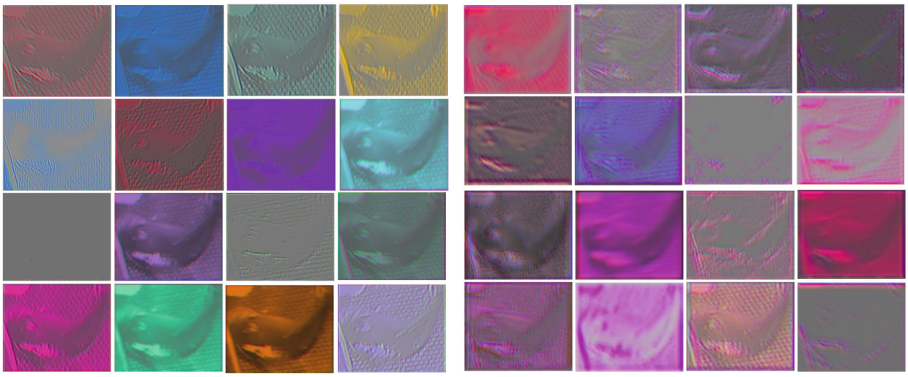
High accuracy of models was observed during testing on original dataset. This could be attributed partly to the fact that images were obtained from fishing boat cameras. In a still camera with 24 frames per second set up, not much difference exists between adjacent frames. Although the object view angles and illumination may vary. Again, closely looking at the feature maps from the network layers revealed that prominent background objects also contributed to this. Learning was tuned towards these objects as against the fish instance. This can be seen clearly in the cross section of feature maps from the first and fourth convolutional layers in the figure below (Fig. 3).



(a) Convolution Layer 1. (b) Convolution Layer 4.

Fig. 3. Feature maps from original image

This effect became more obvious as we go deeper into the network. When training is done on these noisy images, it over-fit on stationary objects that re-appear in images. This adds to the high accuracies recorded. However, these effects were minimal in the experiment with annotated dataset. Fish instances dominate images and this suggest that learning is based on object of interest. Fish parts are visible through the feature maps even as we go deeper into the network. A cross section of feature maps from the first and fourth convolutional layers is shown in the figure below (Fig. 4).



(a) Convolution Layer 1. (b) Convolution Layer 4.

Fig. 4. Feature maps from annotated image

Unbalanced nature of image distribution among classes is also associated with weird model results. This issue was addressed when generating more data for the experiment. Large margin between classes was checked to reduce the variance. Test results shows higher recall when more training data is available. Initial experiments with annotated data performed poorly than the original dataset but we observed increase in performance when more training examples become available. A summary of sensitivity analysis of VGG-16 model (untrained) on the datasets is shown in Table 4. Experiments on the new dataset showed significant increase in accuracy by both models but did not reduce the effects observed. However, transfer learning model log loss was far worse than expected. This could be associated with its strong confidence in false classifications. Another reason could be the variation between images used and the ImageNet images. Transfer learning works best when the two datasets are closely similar.

Table 4. Summary of VGG-16 model performance

Dataset	Precision	Recall	F1-score
Original	0.82	0.82	0.81
Annotated	0.43	0.54	0.47
Original+Noise	0.92	0.91	0.91
Annotated+Noise	0.96	0.96	0.96

6 Conclusion

Deep convolutional neural network performances in context of noisy images was studied. Results shows that in noisy images, the network learns general features

that are also common to all objects in the images. Features from these noisy prominent objects becomes more dominant as we go deeper into the network. As such, they prevent the network from learning specific fish features required for category classification. With well annotated images, the network learns deep features that are category specific and for correct classification. Transfer learning is an emerging area in CNN that has established its presence in recent literatures and has shown stringent results in recent times. But in this study, transfer learning from a pre-trained model on ImageNet was not effective. Learned features are transferable but a closely related dataset could have produced better results. These Results further solidifies that optimum performances are obtained when careful annotation of images is carried out. Manually annotating a massive dataset is challenging and automatic annotation require other techniques such as segmentation and objectness approaches to achieve good results. This opens new research direction in the area of image labelling and data annotation.

References

1. Benson, B., Cho, J., Goshorn, D., Kastner, R.: Field programmable gate array (FPGA) based fish detection using Haar classifiers. Am. Acad. Underwater Sci. (2009)
2. Boudhane, M., Nsiri, B.: Underwater image processing method for fish localization and detection in submarine environment. J. Vis. Commun. Image Represent. **39**, 226–238 (2016)
3. Demertzis, K., Iliadis, L.: Detecting invasive species with a bio-inspired semi-supervised neurocomputing approach: the case of lagocephalus sceleratus. Neural Comput. Appl. 1–10
4. Dosovitskiy, A., Springenberg, J.T., Riedmiller, M., Brox, T.: Discriminative unsupervised feature learning with convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 766–774 (2014)
5. Erhan, D., Szegedy, C., Toshev, A., Anguelov, D.: Scalable object detection using deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2147–2154 (2014)
6. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
7. Hossain, E., Alam, S.S., Ali, A.A., Amin, M.A.: Fish activity tracking and species identification in underwater video. In: 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV), pp. 62–66. IEEE (2016)
8. Lantsova, E., Voitiuk, T., Zudilova, T., Kaarna, A.: Using low-quality video sequences for fish detection and tracking. In: SAI Computing Conference (SAI), pp. 426–433. IEEE (2016)
9. Li, X., Shang, M., Hao, J., Yang, Z.: Accelerating fish detection and recognition by sharing CNNs with objectness learning. In: OCEANS 2016-Shanghai, pp. 1–5. IEEE (2016)
10. Li, X., Shang, M., Qin, H., Chen, L.: Fast accurate fish detection and recognition of underwater images with fast R-CNN. In: OCEANS 2015-MTS/IEEE Washington, pp. 1–5. IEEE (2015)
11. Lowe, D.G.: Object recognition from local scale-invariant features. In: The Proceedings of the Seventh IEEE International Conference on Computer vision, vol. 2, pp. 1150–1157. IEEE (1999)

12. Matai, J., Kastner, R., Cutter Jr., G.R., Demer, D.A.: Automated techniques for detection and recognition of fishes using computer vision algorithms. In: Williams K., Rooper C., Harms, J. (eds.) NOAA Technical Memorandum NMFS-F/SPO-121, Report of the National Marine Fisheries Service Automated Image Processing Workshop, Seattle, Washington, 4–7 September 2010 (2010)
13. Nguyen, N.D., Huynh, K.N., Vo, N.N., van Pham, T.: Fish detection and movement tracking. In: 2015 International Conference on Advanced Technologies for Communications (ATC), pp. 484–489. IEEE (2015)
14. Ogunlana, S.O., Olabode, O., Oluwadare, S.A.A., Iwasokun, G.B.: Fish classification using support vector machine. Afr. J. Comput. ICT **8**(2), 75–82 (2015)
15. Papp, D., Lovas, D., Szűcs, G.: Object detection, classification, tracking and individual recognition for sea images and videos
16. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
17. Rodrigues, M.T., Freitas, M.H., Pádua, F.L., Gomes, R.M., Carrano, E.G.: Evaluating cluster detection algorithms and feature extraction techniques in automatic classification of fish species. Pattern Anal. Appl. **18**(4), 783–797 (2015)
18. Shin, H.C., Roth, H.R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., Summers, R.M.: Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. IEEE Trans. Med. Imaging **35**(5), 1285–1298 (2016)
19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
20. Spampinato, C., Giordano, D., Di Salvo, R., Chen-Burger, Y.H.J., Fisher, R.B., Nadarajan, G.: 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, pp. 45–50. ACM (2010)
21. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems, pp. 3320–3328 (2014)
22. Zhang, D., Kopanas, G., Desai, C., Chai, S., Piacentino, M.: Unsupervised underwater fish detection fusing flow and objectiveness. In: 2016 IEEE Winter Applications of Computer Vision Workshops (WACVW), pp. 1–7. IEEE (2016)