Hierarchical classification, counting and length measurement of fish using a stacking model approach

Raja sekar Shantha kumar, Andreas Hermann, and Daniel Stepputtis

Thünen-Institut für Ostseefischerei (OF),
Alter Hafen Süd 2, 18069 Rostock (Germany)

Abstract In this paper, the development of a hierarchical fish classification framework is presented. The conventional data collection technique for the commercial fish stock assessment is a labour intensive and time consuming procedure. The purpose of this project is to develop a framework that classifies fish species on two level semantic hierarchy label, to count the number of fishes and to measure the length of four different fish species using a small dataset. In stage 1 of the framework, the YOLOv3 convolutional neural network is used to accomplish level one semantic hierarchy label, to count the number of fishes and to measure the length of the detected fish. In stage 2, the features from the images are extracted using the VGG16 convolutional neural network. In stage 3, the stacked generalization technique is implemented to reduce the generalization error and to accomplish level two semantic hierarchy label. The classification accuracy of the stack model is 94%. The root mean square error of the fish length measurement is 1.23 cm. The accuracy in counting the number of fish depends on the detection accuracy of the stage 1 model and the classification accuracy of the stack models. Further, the results can be improved by increasing the size and diversity of the dataset.

Keywords Convolution neural network, stacked generalization, stock assessment

DOI: 10.58895/ksp/1000124383-28 erschienen in:

Forum Bildverarbeitung 2020

DOI: 10.5445/KSP/1000124383 | https://www.ksp.kit.edu/site/books/m/10.58895/ksp/1000124383/

1 Introduction

Biological sampling is a vital procedure in marine data collection to study commercial fish stock. The conventional techniques in use include sorting the catch into species, measuring the length and counting the number of the individual catch. Since this process is labour intensive and time consuming, marine scientists are attempting to develop a deep learning framework to automate this process.

The convolutional neural network (CNN) is such an efficient deep learning technique for classifying images. A collection of tensorflow models trained using different datasets to detect common objects is given by [1]. In general, a single CNN architecture includes two parts, multiple trainable stages (feature extractor) followed by a supervised classifier (deep neural network) [2]. French et al. [3] have used CNN for detecting and counting fishes in the video footage captured on operational trawlers.

Deeper CNN's with a large number of model parameters and also trained on a huge number of examples drastically improves the classification accuracy [4]. Simonyan et al. [5] proposed a network called VGG16 in ILSVRC 2014, trained on ImageNet [6] dataset, achieves 92.7% test accuracy on the testing data. ImageNet is a dataset of nearly 15 million common object images with around 22,000 categories. ILSVRC14 uses a subset of the ImageNet dataset with 1000 images per class (1000 categories).

While there are so many fish species in the world, only a few small open source fish datasets [7] [8] are available. Practically, it is not possible to develop a generalized fish detection model using currently available datasets. To increase classification accuracy using a small dataset, Siddiqui et al. [9] used a cross-layer pooling algorithm with the CNN as feature extractor and support vector machine as a classifier to classify fish species such as P. porosus, P. emeryii and etc.

In general, a single deep learning model (feature extractor and a classifier) trained on small datasets can bias to the dataset used for the training and not performing well on unseen data (overfitting) [10]. Wolpert [11] proposed a method called stacked generalization which uses a number of base models and a single meta model to minimize the generalization error.

Human has the ability to classify a fish in a semantic hierarchy i.e Fish → Flatfish → Dab. While conventional CNN achieved remarkable performance on visual recognition, they do not recognize the object on the natural paradigm of hierarchy. Hence, there is a need in the marine field to develop a framework that allows us to classify fish species in the semantic hierarchy. Inspired by the method proposed by Wolpert [11] and combined with semantic hierarchical label classification, we propose a framework to (a) detect, (b) classify fish in the two level semantic hierarchy, (c) count the number and measure the length of fish.

2 Dataset

We used two public and one own dataset to train the models. The two public datasets are "Open images dataset" [8] and "QUT FISH dataset" [7]. The examples in the public datasets are labled with the level one label of the semantic hierarchy (Fish). The own dataset is captured in the laboratory at "Thünen-Institute (OF)" and at the fishery research vessel "Solea". Therefore, the dataset is named "Thünen dataset" and has both level one and two labels of the semantic hierarchy as shown in figure 2.1. Where the level two hierarchy refers to the fish species. Figure 2.2 show example images from "Thünen dataset".

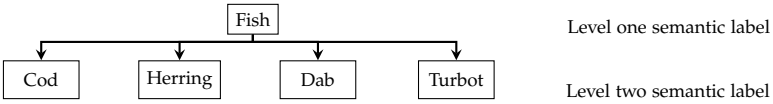
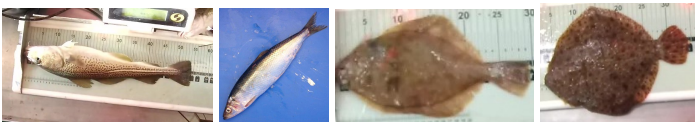


Figure 2.1: Hierarchical annotation of the dataset

Further to train the base models, "Thünen dataset" is divided into training data and testing data as shown in figure 2.3.

3 Classification Procedure

The developed framework has three stages, stage 1 – detection and classification of level one label of the semantic hierarchy, stage 2 –



(a) (b) (c) (d)

Figure 2.2: (a) Cod, (b) Herring, (c) Dab, (d) Turbot

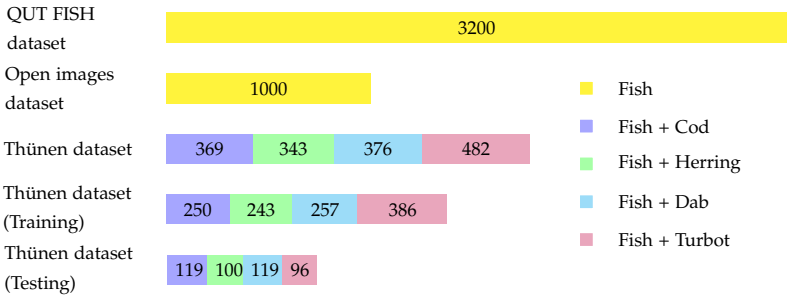


Figure 2.3: List of datasets used in training

feature extraction and stage 3 – classification of level two label of the semantic hierarchy as shown in figure 3.1. In stage 1, YOLOv3 CNN is used to detect the fish and to accomplish level one label of the semantic hierarchy. The detected fish is cropped and in stage 2, the features are extracted using VGG16 CNN. Stage 3 of the framework has a stack model with 2 layers. Layer 1 has three base models and layer 2 has a single meta model. The extracted features are used to train the three base models of the stack layer 1. Later, the prediction probabilities of the three base models are used to train the meta model of the stack layer 2. In stage 3, the level two label of the semantic hierarchy is accomplished.

3.1 YOLOv3 object detector

To detect a fish, a real time single shot object detector YOLOv3 [12] convolutional neural network is used. The YOLOv3 network is trained on the COCO dataset [13] to detect 80 common objects where

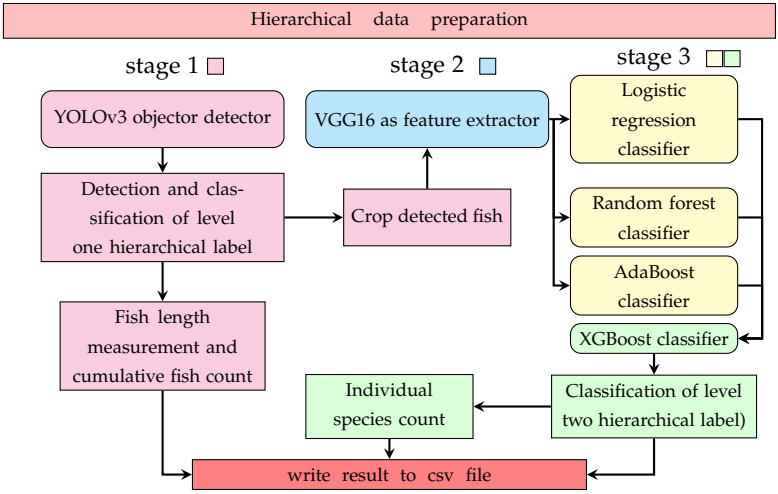


Figure 3.1: The flowchart representation of the classification procedure

fish is not one among those classes. We implemented transfer learning [14] to detect single class, Fish. Both ”QUT FISH dataset” and ”Open image dataset” with the level one label of the semantic hierarchy are used to train the model. The ”Thünen dataset” (entire dataset) with the level one label of the semantic hierarchy is used to evaluate the model performance. Figure 5.1 shows the training and validation curve of YOLOv3.

3.2 VGG16 as feature extractor

To use pre-trained VGG16 CNN [5] as a feature extractor, the last few fully connected layers were removed (modified VGG16). The image propagates from the first layer to the last layer of the modified VGG16 (feature extractor) and outputs a volume of the shape 7$\times$ 7 $\times$ 512. This output volume is flattened into a feature vector of the dimensions 25,088.

To train and evaluate the base models in stage 3 (stack layer 1), the features from the ”Thünen training and testing data” are extracted and tabulated. The shape of the tabular datasets is (number of im-

ages $\times$ 25088). Figure 3.2 shows the pictorial representation of the feature map extracted from block1_conv2D layer of VGG16.

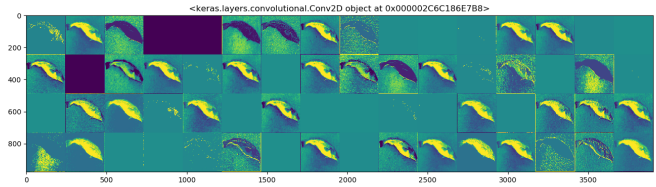


Figure 3.2: Feature map of an example image

3.3 Stacking model approach

Ensemble learning is a technique to reduce the variance of the model. Such technique for classification problems are majority voting [15], weighted majority voting [16] and stacking [11]. In majority voting, the final decision is made by a majority vote of the individual classifiers. Whereas in the weighted majority voting, the individual classifiers are assigned with different weights depending on the performance and the final decision is made by counting the weighted votes of the individual decisions [16].

The stacking or stacked generalization uses a concept meta classifier. The meta classifier is trained on the prediction probabilities of the individual base models to make the final prediction. This method reduces the generalization error and increases the prediction accuracy.

Base models

The base models used in the framework are logistic regression, random forest and AdaBoost classifier. These models are trained on the "Thünen training dataset" (features vectors) using K-fold cross validation ($K = 3$). The prediction probabilities of each base model are concatenated as shown in figure 3.3 and used as a training dataset to train the meta model.

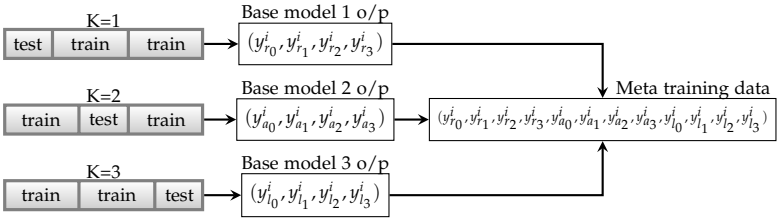


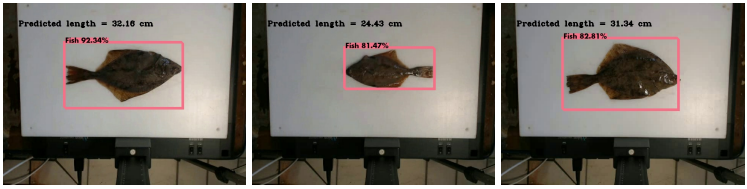
Figure 3.3: Meta training data

Meta model

The meta model used is XGBoost classifier and fitted on the prediction probabilities of the base models and the model performance is evaluated using the "Thünen testing dataset" (feature vectors).

4 Fish counting and length measurement

By using the YOLOv3 network, the overall number of fish (level one hierarchy) is counted. Similarly, the number per fish species is counted using the classification output of the stack model, following the previous detection of the YOLOv3 network.



(a) (b) (c)

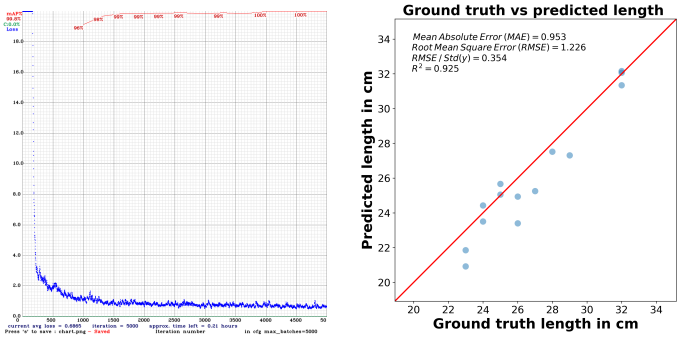
Figure 4.1: (a), (b) and (c) - Predicted length using YOLOv3

The YOLOv3 network is used to predict the length of the fish. The object detection happens in the three scales and at three different layers of the YOLOv3 network, 82, 94 and 106. The input image of the shape (416, 416, 3) is downsampled by the factor (stride) 32, 16

and 8 at three detection layers and the resultant feature map has the shape of $13 \times 13 \times depth$, $26 \times 26 \times depth$ and $52 \times 52 \times depth$ respectively. For each cell in the resultant feature map, three bounding boxes are generated by the YOLOv3 network. The maximum probability of the bounding box containing a class is given by the product of objectness score and confidence. The real width b_w and the height b_h of the bounding box are computed by calculating the log-space transform (offset) to the predefined anchors. And to calculate the center coordinate (b_x, b_y) of the bounding box, a sigmoid function is used [12]. Figures 4.1 (a), (b) and (c) show three examples of the predicted length using the YOLOv3 network.

5 Results and Discussion

The training graph figure 5.1 (a) shows that the YOLOv3 network's training loss is decreasing gradually and reaches an average loss of 0.68. The mean average precision of the validation data reaches 100%.



(a) (b)

Figure 5.1: (a) YOLOv3 training curve (b) Fish length measurement plot

Figure 5.1 (b) shows the ground truth length vs predicted length of the fish plot. The root mean square error (RMSE) of the fish length measurement is 1.23 cm.

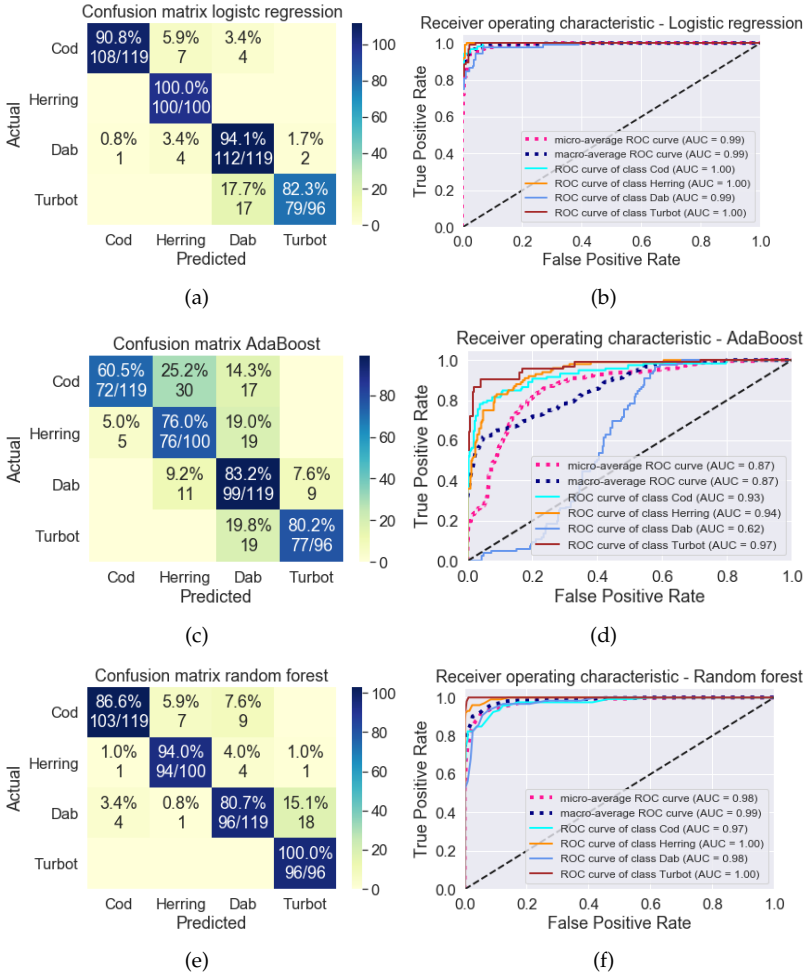
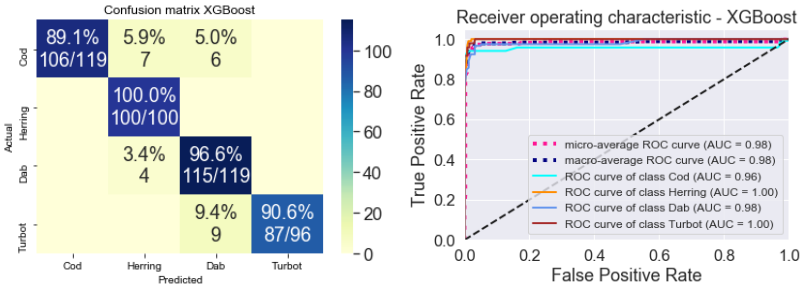


Figure 5.2: Confusion matrix of (a) Logistic regression, (c) AdaBoost and (e) Random forest. ROC curve of (b) Logistic regression, (d) AdaBoost and (f) Random forest



(a) (b)

Figure 5.3: Xgboost (a) Confusion matrix and (b) Roc

From the computed confusion matrix, figures 5.2 (a), (c), (e) and 5.3 (a) the different metrics to evaluate the stack model performance are calculated and shown in table 1. Figure 5.2 (b), (d), (f) and figure 5.3 (b) show the receiver operating characteristic curve with the area under curve value for four different classes. Comparing the classification accuracy, precision and the recall of the meta model and base models, it is clear that the meta model XGBoost out performances all three base models.

Table 1: Results of the stack models

Classifier	Precision	Recall	f1-score	Simple Accuracy	Micro AUC
Random forest	0.90	0.90	0.90	0.90	0.98
Logistic regression	0.93	0.92	0.92	0.92	0.99
Adaboost	0.78	0.75	0.75	0.75	0.87
XGBoost	0.95	0.94	0.94	0.94	0.98

6 Conclusion

From the above results, it becomes clear that the classification accuracy, precision and the recall of the fish species can be increased using a stacked generalization. The disadvantage of this approach is computationally expensive to train the model and to tune the hyper

parameter. The predicted length measurement values have relatively high root mean square error (RMSE). Therefore, the applied simple method of length estimation might not suitable for many biological applications. Hence, for further improvement, we could add more data in the training set for better accuracy of object localization or we can implement a machine vision approach such as a stereo vision for length measurement.

References

1. J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," 07 2017, pp. 3296–3297.
2. H. Lee, R. Grosse, R. Ranganath, and A. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," 01 2009, p. 77.
3. G. French, M. Fisher, M. Mackiewicz, and C. Needle, "Convolutional neural networks for counting fish in fisheries surveillance video," 09 2015.
4. D. C. Ciresan, U. Meier, L. Gambardella, and J. Schmidhuber, "Deep big simple neural nets excel on handwritten digit recognition," *ArXiv*, vol. abs/1003.0358, 2010.
5. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.
6. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.
7. K. Anantharajah, Z. Ge, C. McCool, S. Denman, C. B. Fookes, P. Corke, D. W. Tjondronegoro, and S. Sridharan, "Local inter-session variability modelling for object classification," in *Winter Conference on Applications of Computer Vision (WACV), 2013 IEEE Conference on*, 2014.
8. I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, S. Kamali, M. Malloci, J. Pont-Tuset, A. Veit, S. Belongie, V. Gomes, A. Gupta, C. Sun, G. Chechik, D. Cai, Z. Feng, D. Narayanan, and K. Murphy, "Openimages: A public dataset for large-scale multi-label and multi-class image classification." *Dataset available from https://storage.googleapis.com/openimages/web/index.html*, 2017.

9. S. Siddiqui, I. Malik, F. Shafait, A. Mian, M. Shortis, and E. Harvey, "Automatic fish species classification in underwater videos: Exploiting pretrained deep neural network models to compensate for limited labelled data," *ICES Journal of Marine Science*, vol. 75, 05 2017.
10. X. Ying, "An overview of overfitting and its solutions," *Journal of Physics: Conference Series*, vol. 1168, p. 022022, feb 2019. [Online]. Available: https://doi.org/10.1088%2F1742-6596%2F1168%2F2%2F022022
11. D. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, pp. 241–259, 12 1992.
12. J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 04 2018.
13. T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2014.
14. K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big Data*, vol. 3, no. 1, p. 9, May 2016. [Online]. Available: https://doi.org/10.1186/s40537-016-0043-6
15. L. Lam and S. Y. Suen, "Application of majority voting to pattern recognition: an analysis of its behavior and performance," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 27, no. 5, pp. 553–568, 1997.
16. A. F. R. Rahman, H. Alam, and M. C. Fairhurst, "Multiple classifier combination for character recognition: Revisiting the majority voting system and its variations," in *Document Analysis Systems V*, D. Lopresti, J. Hu, and R. Kashi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 167–178.