Time-unfolding Object Existence Detection in Low-quality Underwater Videos using Convolutional Neural Networks

Helmut Tödtmann[1,4], Matthias Vahl[1], Uwe Freiherr von Lukas[1,2] and Torsten Ullrich[3,4]
[1]*Fraunhofer Institute for Computer Graphics Research IGD, Rostock, Germany*
[2]*University of Rostock, Institute for Computer Science, Rostock, Germany*
[3]*Fraunhofer Austria Research GmbH, Visual Computing Graz, Austria*
[4]*Institute of Computer Graphics and Knowledge Visualization, Graz University of Technology, Austria*
helmut.toedtmann@igd-r.fraunhofer.de, torsten.ullrich@fraunhofer.at

Keywords: Convolutional Neural Network, Deep Learning, Environmental Monitoring, Implicit Segmentation, Detection.

Abstract: Monitoring the environment for early recognition of changes is necessary for assessing the success of renaturation measures on a facts basis. It is also used in fisheries and livestock production for monitoring and for quality assurance. The goal of the presented system is to count sea trouts annually over the course of several months. Sea trouts are detected with underwater camera systems triggered by motion sensors. Such a scenario generates many videos that have to be evaluated manually. This article describes the techniques used to automate the image evaluation process. An effective method has been developed to classify videos and determine the times of occurrence of sea trouts, while significantly reducing the annotation effort. A convolutional neural network has been trained via supervised learning. The underlying images are frame compositions automatically extracted from videos on which sea trouts are to be detected. The accuracy of the resulting detection system reaches values of up to 97.7 %.

1 INTRODUCTION

It is of great interest to the world to keep nature and environment in balance and to maintain a healthy state. Based on the Red List of Endangered Species (Winkler et al., 1991) the population of sea trouts was seriously threatened. As a consequence, in the time from from 1992 to 1999 renaturation programs attempted to repopulate these regions as pilot project. The success of these programs has been documented and guidelines were derived, which serve as a basis for further population recovery projects in other areas; the success of this project led to similar projects in 30 further areas (Schwevers and Adam, 2020). The measured results have been published (Mannerla et al., 2011), but in 2015 difficulties in correctly monitoring the populations have been discovered (Pedersen et al., 2017), because the underlying data was insufficient, erroneous and did not allow to distinguish between different populations. Consequently, a new method of regular counting was developed: Underwater video cameras were installed in selected reference areas in order to precisely count all incoming and outgoing sea trouts as shown in Figure 1. Sea trouts are a very special kind of fish. Once every year, sea trouts leave the salt water of the Baltic Sea and battle their way upstream into freshwater, where they form hollows in sandy riverbeds to lay their eggs. But at numerous spots along the way, their path is blocked by dams or weirs. In many rivers, the sea trouts have already been driven to extinction. Current measuring methods are counting the hollows or fishing a reference part of the water. All these methods are labor-intensive, locally limited and sometimes dangerous to the stock. Environmental institutes in Europe have been commissioned to monitor sea trout numbers. Along the rivers, members of each institute are constructing bottle necks that sea trout have to tra-

Figure 1: Camera setups in rivers are used to precisely count sea trouts. The photo has been taken at a low water level.

verse and are monitoring them with video cameras. But evaluating the images is still a work-intensive process: an employee needs 1416 hours only to watch over 340 000 videos that have been captured at all bottle necks over the course of five months. An example of the video frames to evaluate is shown in Figure 2. It illustrates that even humans need training to detect

Figure 2: Video frames have to be evaluated manually or automatically. The top row shows *no sea trout* whereas the bottom row shows a positive result (*sea trout*).

and classify sea trout in video images.

This is where this article comes in. The new automatic system described here takes a common personal computer only five days to do the same work using a machine learning approach. Another advantage of deploying artificial intelligence (AI) is that it makes genuine wide-scale surveillance of sea trout stocks possible, whereas previously only five or six rivers could be monitored. Furthermore this approach allows to replace the usual costly video annotation process with a fast and easy alternative, while obtaining the information of temporal occurrence of sea trouts in a video.

2 RELATED WORK

The presented system uses artificial intelligence techniques and applies them to a non-computer science domain. In order to address both areas, computer science and the field of application, the related work from both areas will be examined.

2.1 Convolutional Neural Networks

An extensive survey on deep learning theory and architectures is given by (Alom et al., 2019). The authors motivate and explain the following concepts which are crucial to understand the details of the related work and the approach: supervised learning, convolutional neural network, pooling layer, activation function, optimization methods, transfer learning.

When choosing any deep learning approach for the task of finding objects in images or videos, it is important to distinguish between classification, detection and segmentation. Assuming that a video is simply a sequence of images, *classification* is the process of deciding whether an image belongs to a particular category among a set of given categories. *Detection* attempts to find objects of a particular category within an image and usually returns a list of bounding boxes. A *segmentation* task can be understood as a more sophisticated detection on a per-pixel level. It returns a binary mask, thus providing more precise results.

Krizhevsky, Sutskever, and Hinton introduced the first convolutional neural network for classification that achieved state of the art performance at the ImageNet Large Scale Visual Recognition Challenge in contrast to former used algorithms (Krizhevsky et al., 2012). They used five convolutional layers, of which some were followed by pooling layers, and additionally three fully connected layers, while making use of the rectified linear unit (ReLU) activation function. In "Going Deeper with Convolutions" (Szegedy et al., 2014) improved this approach with "GoogLeNet" (see Figure 3). The designed network consisting of 22 layers, that needed twelve times less learned parameters, while still achieving a far better result. Concerning detection, (Redmon et al., 2016) presented a milestone, outperforming the former state of the art of detection in terms of accuracy and speed, while only needing one forward pass. They used 24 convolutional layers followed by two fully connected layers. Instead of inception modules from "GoogLeNet" they make use of simple (1×1) reduction layers and (3×3) convolutional layers. In the field of segmentation, (He et al., 2017) developed "Mask R-CNN" which is based on the classification neural network called "ResNet" presented by (He et al., 2016). The network uses 101 layers. It consists of three main components, which compute the bounding boxes, the class identity, and the segmentation mask.

2.2 Fishes in Computer Vision

Deformable template matching can be used to classify two species of fish (Rova et al., 2007). The authors improved over previous Support Vector Machine (SVM)-based methods and reached an accuracy of up to 90%. However, the approach required manually cropped underwater video images.

Spampinato et al. introduced a framework for detecting, tracking, and counting fishes in

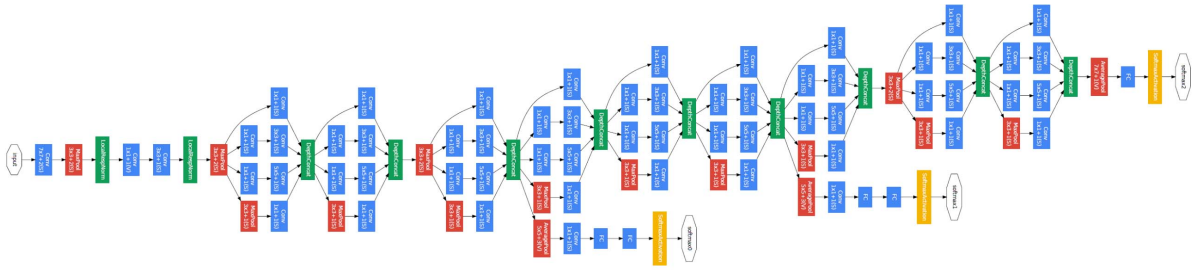

Figure 3: Structure of original "GoogLeNet" has been designed by (Szegedy et al., 2014).

videos (Spampinato et al., 2008). A classification accuracy of 93% is achieved while the overall counting success rate is about 85%. Their approach only worked on video data of controlled environments and required extensive human annotation beforehand.

(Ravanbakhsh et al., 2015) detect fishes in underwater video with a shape-based approach. A Haar classifier is used for precise localisation of fish head and snout, effectively leading to a sub pixel accuracy for retaining the shape of the fish. For a set of 35 samples, they report an accuracy of up to 100%. However, this approach seems not to work in uncontrolled environments and requires images that contain completely visible fish.

An approach using convolutional neural networks has been presented by (French et al., 2015). They make us of a segmentation approach based on the "N^4 fields" algorithm and tested a variety of different architectures of which the best delivered a count accuracy of 92,87% (relative error of 7,13%). Unfortunately, this approach requires segmentation annotation for each video frame of the training set.

(Shafait et al., 2016) introduce image set classification with a one-nearest-neighbour classification. Given a set of already localised images of a fish with unknown species type, it is matched with all existing training sets; for which multiple exist for each class. Instead of calculating an average pairwise distance between all possible image combinations, they construct synthetic images as linear combination from previously calculated base images to be as similar as possible. The label of the training image set with the smallest distance is applied. They achieve an accuracy of 94,6%.

In "Comparison between Deep Learning and HOG+SVM Methods" (Villon et al., 2016) compared the combination of histogram of oriented gradients (HOG) and support-vector machines (SVM) against a customized GoogleNet architecture for detecting coral reef fishes. Training was done with labeled cropped objects from video frames, while testing was done with a sliding window approach on video frames. It was found that the deep learning approach performed better in general.

(Li et al., 2016) use a Fast RCNN approach on images of fish to distinguish between 12 classes and reach a mean average precision of 81.4%. However, the approach works only on images and requires bounding box annotation for each video frame of the training set.

An approach by (Sung et al., 2017) uses the "You Only Look Once" (YOLO) algorithm (Redmon et al., 2016) to detect fishes on underwater images. A grid search on the hyper parameters is performed to improve the performance. Also, their data set is enhanced with lots of positive and negative samples to create a more robust network and reach a classification accuracy of 93%.

In "Fish Recognition Using Convolutional Neural Network" the authors handcrafted several convolutional neural network architectures to perform a classification task on four species of fish (Ding et al., 2017). The best network achieved a classification accuracy of 96,5%.

Further improvements have been obtained by (Rathi et al., 2018). They also develop a novel convolutional neural network architecture that trains on denoised still images from a custom fish data set and includes a threshold-based segmentation as additional fourth image layer. The result is an classification accuracy of 96,29%.

An approach working on an input data set with a video quality comparable to the presented system has been presented by (Shevchenko et al., 2018). They apply three background subtraction methods on underwater videos to detect fishes. The best method achieves 60% accuracy.

2.3 Summary

While the field of deep learning for object recognition undergoes constant improvements, the domain of dealing with fishes in underwater images and videos has not yet been making use of all the newly developed possibilities to the full extent. Contrary to our needs, some of these approaches additionally deal

with tracking, shape retrieval and species classification, while most of them are restricted to images or even selected parts of images. Our use case of binary video classification with obtaining the times of occurrence of sea trout on vast amount of video data is not present in the available literature. To the best of the authors' knowledge the following approach has not been proposed before.

3 APPROACH

The used camera system has been selected in the previous research project so that no further intervention was possible. For the sake of completeness, the original video material is described here.

3.1 Data Set Description

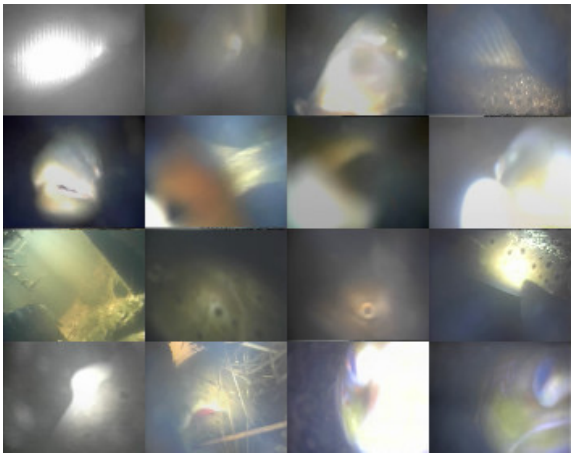

Figure 4: Selection of videos containing *sea trout*. The frame containing the most fish like features was selected.

The installed camera systems recorded 341 238 videos in total. The videos show eight different rivers over the course of five months. Each video has a resolution of 704×576 pixels and a frame rate of 15 frames per second. Figure 4 and Figure 5 showcase visual variance, low quality and separability difficulty. The first 20 lines of each frame are reserved for overlaid video information and thus removed in the preparation process, leading to an effective resolution of 704×556 pixels. The video length varies between 5 and 300 seconds, the average duration of a video is about 15 seconds. Videos are stored as a Matroska Video (MKV) files, using the h264 codec. The average size of a video file is approximately 550 kilobytes. For generating labeled learning material 10 % of these videos were selected at equal time intervals and manually classified by the domain expert Uwe Friedrich with the tools described in the Section "Implementation". As a consequence, 307 114 videos remained unlabeled.

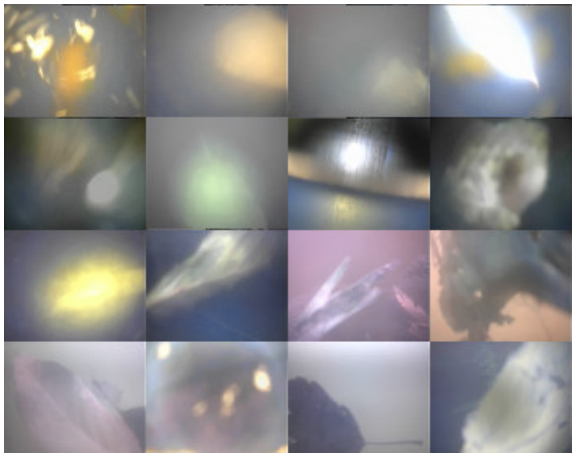

Figure 5: Selection of videos containing *no sea trout*. The frame containing the most fish like features was selected.

3.2 Detection by Classification

As mentioned in the Section "Related Work", almost all approaches of detecting objects in still images and videos require labeled data, for example bounding boxes or polygons marking these objects. For videos this labeling process is very time consuming, since a sufficient number of frames have to be annotated for each video. In our use case, any of these approaches were not usable, since the amount of available video data and the visual variance was too high to create labeled data for supervised learning. Instead, we developed an approach that only requires the information of whether a video contains sea trouts (see Figure 6). This approach can reliably detect the existence

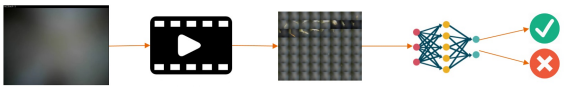

Figure 6: The pipeline starts with a video and returns a binary classification.

of sea trout in previously unseen videos and beyond that, can deliver information on which frames of a video contained sea trout, and potentially even where in these frames the sea trout was located. For each single video the workflow is as follows:

1. removing video header
2. time-unfolding convolution
3. implicit coarse segmentation
4. 2D classification

3.3 Time-unfolding Convolution

In order to enable classification on videos, a sufficient number of individual images from a video sequence are arranged next to each other in a grid layout. Figure 7 shows an example with 64 successive images of a video.

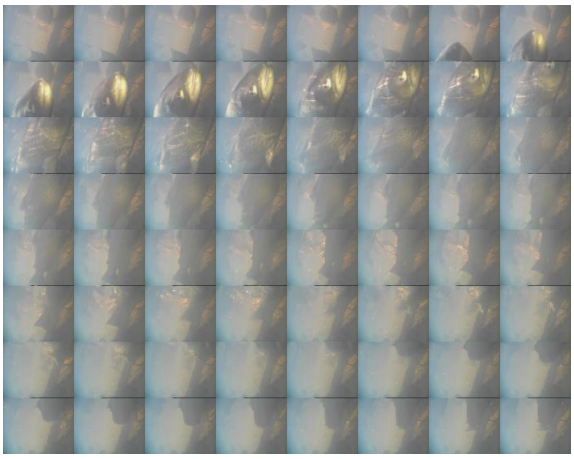

Figure 7: The image sequence of a video is arranged in a 2D layout.

Convolutional neural networks are "a natural way to embed translation invariance" (Deng et al., 2013), i.e. for the recognition of learned features it is irrelevant where these are located in the image, as long as they are present. That way, existing pre-trained neural networks for 2D classification can be used with little modifications as opposed to developing and training a 3D classification network from scratch. We found that resulting artificial edges between images did not hurt the overall classification performance, as these occur both in image grids containing and not containing sea trout and thus did not represent an important visual feature. On the one hand there must be enough single images to reliably detect even a short appearance of a sea trout in a video. On the other hand not any arbitrarily high number of single images can be used for calculation. In detail, small images would negate possible information retrieval, but big image grids would represent a limiting factor concerning the graphics memory and the general performance of the graphics card. An 8×8 grid has proven to be the perfect trade-off. In our tests a 4×4 grid often didn't capture short occurrences of fish, while a 16×16 grid implicated images which have been too small for detection. A video with width w, height h, and a scale value s, leads to s^2 evenly distributed sample frames; i.e. the three dimensional information is rearranged in two dimensions as follows:

$$f(x,y,t) \mapsto \left(\frac{x + w \cdot (t \bmod s)}{s}, \frac{y + h \cdot \lfloor t/s \rfloor}{s} \right)$$

with

$$t \in [0, s^2 - 1], x \in [0, w-1], y \in [0, h-1]$$

Using this scaled grid the resulting image has the same dimensions as a single frame from the original video (see Figure 7).

3.4 Implicit Coarse Segmentation

The used network structure is a modified and pre-trained "GoogLeNet" (see Figure 3 and (Szegedy et al., 2014)). We chose this architecture due to its good performance, which has been confirmed in our tests, while being fast. The input layer has been changed to match the dimension of the image grid. Consequently, the sizes of all following layers are increased, leading to a final inception layer with dimension 22×17 and the corresponding final pooling layer of dimension 16×11. In that regard the step of increasing the size of the input layer to match the image size is technically equivalent to removing pooling layers with or without their connected convolutional layers deeper in the network, like shown in (Zhou et al., 2016). This shifts the category of the algorithm from simple classification to implicit segmentation with very low resolution, neither requiring segmented training data, nor delivering segmented results. The output of the final inception layer can be interpreted as set of activation maps, indicating which parts of the grid image are responsible for the classification result. For an activation map A with width w and height h, the time point t of the highest activation can be found as follows: Within the range $R = [0, w-1] \times [0, h-1]$ the maximum (x, y) meets the equation

$$\exists (x,y) \in R \forall (u,v) \in R : A(u,v) \leq A(x,y).$$

Using the scaling factor s mentioned above the time t can be determined via

$$t = \frac{y \cdot s^2}{h} + \frac{x \cdot s}{w}.$$

It is important to note that the highest activation of the implicit coarse segmentation – the final value of t – is not directly used for the classification. It is rather an evaluation tool to ensure that the neural network indeed learned a set of useful features to detect sea trouts in the grid image. For example, Figure 8 is indicating that the most important parts of the grid image from Figure 7 are at $t = 7$ (first row to the right) and $t = 8$ (second row to the left), which clearly contain sea trout. The whole second row of the grid image also contains sea trout, but is not highly visually present in this activation map, which may indicate that either these frames are not necessary for the network to detect sea trout, or are present in another activation map.

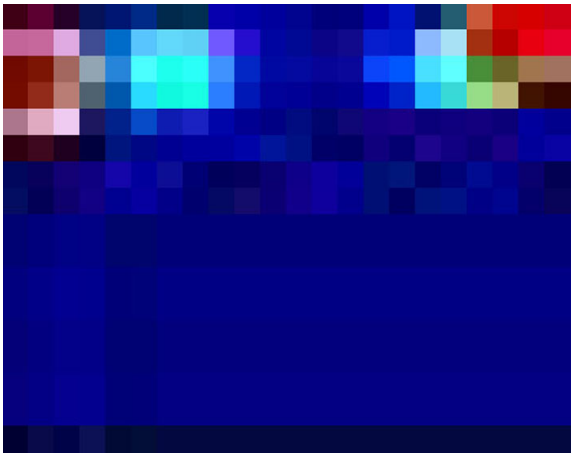

Figure 8: This visualisation shows the activation map that corresponds to Figure 7. The regions with a high activation (red) contain sea trouts, which can be verified in the input images shown in the grid layout.

4 IMPLEMENTATION

In order to realize the new approach several tools have been developed. The software has been programmed using the integrated development environment "Qt" (www.qt.io) in the programming language C++. The software relies on the NVIDIA fork of the framework Caffe (Jia et al., 2014) using GPU acceleration. As a consequence, at least a CUDA 8.0 compatible graphics card has to be present, including the appropriate drivers and libraries.

4.1 Annotation Tool

The annotation tool loads a custom set of videos, displays them in a list, while the currently selected video is played in a loop. By a single key stroke the annotator can decide if that video contains sea trout or not, after which the next video is selected. With this tool, an experienced annotator needs just a few seconds to label a video. Having completed the annotation of the complete set of videos, they are copied and split to separate folders according to their class.

4.2 Preparation Tool

From a set of videos, in each video the header is removed to prevent the learning of misleading information or introducing biases. Then the time-unfolding convolution is performed. The resulting grid images are stored as lossless compressed Portable Network Graphics (PNG) files.

4.3 Training the Network

During the training of the network the deep learning framework DIGITS (developer.nvidia.com/digits) is used. First, a data set for classification is created out of the grid images from the preparation tool. Half of the data is used for training, the other half is reserved for validation and testing. According to the number of different scenarios, class balancing is performed to assure each scenario is represented with about the same number of grid images. Likewise, class balancing is performed again with focus on the ratio of sea trout videos to non-sea trout videos. Class balancing is done with enriching the smaller class/scenario with duplicates of itself. The Adam optimizer (Kingma and J., 2015) is used with the learning rate $lr = 0.0001$ and the exponential decay $\gamma = 0.95$. The training of the network took about 75 hours on four simultaneously used GPUs of type GeForce GTX TITAN X with 12 GB of RAM each.

4.4 Analysis Tool

The trained network is loaded into the Caffe framework for C++. Afterwards, a custom set of unlabeled videos is loaded. The steps from the preparation tool are performed. For each resulting grid image, inference is performed with the convolutional neural network. The corresponding videos are copied and split into separate folders according to the classification result. Finally, a report is generated, listing all videos with their assigned class and classification confidence in the form of the softmax function.

5 EVALUATION

The trained network has been evaluated with unlabeled videos. The data set (see Section "Data Set Description") comprehends 307 114 unlabeled videos. The analysis tool has classified 5 235 of 307 114 videos as "sea trout". A manual control by the domain expert Uwe Friedrich confirmed that 5 098 of 5 235 videos were containing sea trout, resulting in a precision of 97.38%. Likewise a showcase sample of 6 000 videos from the 307 114 videos has been selected and then classified by both the expert and the analysis tool. The expert classified 938 of these videos as "sea trout", while the analysis tool classified 902 videos as "sea trout", resulting in a precision of 96.16 %. The results are listed in Table 1. For the first case, no accuracy can be reported due to the lack of ground truth data, for the second case, an accuracy

Table 1: Evaluation of the new system using "previously unseen" data.

Test Size	Positives by System	Sea trout by Expert	Precision
307 114	5 235	5 098	**97.38%**
6 000	938	902	**96.16%**

of 97.70 % is achieved:

$$accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} = \mathbf{97.70\ \%}$$

The values of true/false positive/negative detections are listed in Table 2.

Table 2: Confusion matrix for selected set. Abbreviations are for true/false positive/negative.

6 000 Samples	Labeled *positive*	Labeled *negative*
Detected *true*	$TP = 902$	$FP = 102$
Detected *false*	$FN = 36$	$TN = 4960$

6 CONCLUSION

Demonstrated at sea trouts examples, a convolutional neural network has been trained via supervised learning. The underlying images are frame compositions automatically extracted from videos on which sea trouts are to be detected. The approach detects objects in underwater videos even in use cases with reduced quality: low resolution, minimal contrast, poor visibility.

6.1 Contribution

It was shown that time-unfolding convolution enables the use of still image classification rather than the usual detection or segmentation approaches, which in turn allows much faster annotation of ground truth data while also achieving very high accuracy. With the use of implicit segmentation, further statements can be made about the times of the occurrence of sea trout in the video. The accuracy of the resulting detection system reaches values of up to 97.7 %.

6.2 Benefit

The manual evaluation of videos does not scale. Within this project an employee would need 1416 hours only to watch all videos that have been captured. The automatic solution is not only much faster, since no interaction is necessary, the time required even becomes irrelevant. Now, the monitoring process scales and is not the limiting factor any more.

6.3 Outlook

In the future, network structure modifications to increase the implicit segmentation resolution will be explored, which could allow implicit detection of locations of sea trout features in a video frame. Likewise, we will evaluate how low the quality of the videos can be and what the optimal parameters for the time unfolding convolution are. Furthermore it is imaginable to improve on the process of class activation mapping and develop general purpose approaches for precisely detecting and segmenting arbitrary objects in videos with little to no information about position and times of occurrence.

ACKNOWLEDGEMENTS

The authors would like to thank Uwe Friedrich from the Institut für Fisch und Umwelt. All the data was generated and labeled in the project "Saisonale Ermittlung des Meerforellenbestandes in den Einzugsgebieten der Vorpommerschen Boddengewsser und der Mecklenburger Bucht mittels videooptischer Erfassung aufsteigender Individuen in Verbindung mit Kartierungsarbeiten und fischereibiologischen Untersuchungen", project number DRM-149, "Landesforschungsanstalt fr Landwirtschaft und Fischerei ".

The authors also thank Tim Dolereit, Tom Krause and Mohamad Albadawi from MAG of Fraunhofer Institute for Computer Graphics Research in Rostock for their valueable input.

Furthermore, the authors acknowledge the generous support of the Carinthian Government and the City of Klagenfurt within the innovation center *KI4Life*.

REFERENCES

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., Van Essen, B. C., Awwal, A. A. S., and Asari, V. K. (2019). A State-of-the-Art Survey on Deep Learning Theory and Architectures. *Electronics*, 8:292ff.

Deng, L., Abdel-Hamid, O., and Yu, D. (2013). A Deep Convolutional Neural Network using Heterogeneous Pooling for Trading Acoustic Invariance with Phonetic Confusion. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 38:6669–6673.

Ding, G., Song, Y., Guo, J., Feng, C., Li, G., and He, B. (2017). Fish Recognition Using Convolutional Neural Network. *Mts*, pages 0–3.

French, G., Fisher, M., Mackiewicz, M., and Needle, C. (2015). Convolutional Neural Networks for Counting Fish in Fisheries Surveillance Video. *Machine Vision of Animals and their Behaviour Workshop*, 7:1–10.

He, K., Gkioxari, G., Dollar, P., and Girshick, R. (2017). Mask R-CNN. *IEEE International Conference on Computer Vision (ICCV)*, 15:2980–2988.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 19:770–778.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional Architecture for Fast Feature Embedding. *International Conference on Multimedia*, 22:675–678.

Kingma, D. P. and J., B. L. (2015). Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations (ICLR)*, 3:19ff.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *International Conference on Neural Information Processing Systems*, 25:1097–1105.

Li, X., Shang, M., Qin, H., and Chen, L. (2016). Fast accurate fish detection and recognition of underwater images with Fast R-CNN. *OCEANS 2015 - MTS/IEEE Washington*, pages 1–5.

Mannerla, M., Andersson, M., Birzaks, J., Debowski, P., Degerman, E., Huhmarniemi, A., Häaggström, H., Ikonen, E., Jokikokko, E., Jutila, E., Kesler, M., Kesminas, V., Kontautas, A., Pedersen, S., Persson, J., Romakkaniemi, A., Saura, A., Shibaev, S., Titov, S., Tuus, H., Tylik, K., and Yrjänä, T. (2011). *Salmon and Sea Trout Populations and Rivers in the Baltic Sea*. Helsinki Commission, 1 edition.

Pedersen, S., Degerman, E., Debowski, P., and Petereit, C. (2017). Assessment and recruitment status of Baltic Sea trout populations. *Sea Trout: Science & Management: Proceedings of the International Sea Trout Symposium*, 2:423–441.

Rathi, D., Jain, S., and Indu, S. (2018). Underwater Fish Species Classification using Convolutional Neural Network and Deep Learning. *International Conference on Advances in Pattern Recognition (ICAPR)*, 9:344–349.

Ravanbakhsh, M., Shortis, M. R., Shafait, F., Mian, A., Harvey, E. S., and Seager, J. W. (2015). Automated Fish Detection in Underwater Images using Shape-based Level Sets. *Photogrammetric Record*, 30:46–62.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 19:779–788.

Rova, A., Mori, G., and Dill, L. M. (2007). One fish, two fish, butterfish, trumpeter: Recognizing fish in underwater video. *IAPR Conference on Machine Vision Applications (MVA)*, 7:404–407.

Schwevers, U. and Adam, B. (2020). *Fish Protection Technologies and Fish Ways for Downstream Migration*. Springer International Publishing, 1 edition.

Shafait, F., Mian, A., Shortis, M., Ghanem, B., Culverhouse, P. F., Edgington, D., Cline, D., Ravanbakhsh, M., Seager, J., and Harvey, E. S. (2016). Fish Identification from Videos Captured in Uncontrolled Underwater Environments. *ICES Journal of Marine Science: Journal du Conseil*, 73:2737–2746.

Shevchenko, V., Eerola, T., and Kaarna, A. (2018). Fish Detection from Low Visibility Underwater Videos. *International Conference on Pattern Recognition*, 24:1971–1976.

Spampinato, C., Chen-Burger, Y. H., Nadarajan, G., and Fisher, R. B. (2008). Detecting, Tracking and Counting Fish in Low Quality Unconstrained Underwater Videos. *International Conference on Computer Vision Theory and Applications (VISAPP)*, 3:514–519.

Sung, M., Yu, S.-C., and Girdhar, Y. (2017). Vision based Real-time Fish Detection Using Convolutional Neural Network. *OCEANS*, 7:1–6.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). Going Deeper with Convolutions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 17:1–9.

Villon, S., Chaumont, M., Subsol, G., Villéger, S., Claverie, T., and Mouillot, D. (2016). Coral Reef Fish Detection and Recognition in Underwater Videos by Supervised Machine Learning: Comparison between Deep Learning and HOG+SVM Methods. *Lecture Notes in Computer Science, Artificial Intelligence, and Bioinformatics*, 10016:160–171.

Winkler, H. M., Hamann, N., and Waterstraat, A. (1991). *Rote Liste der gefährdeten Rundmäuler, Süßwasser- und Wanderfischarten Mecklenburg-Vorpommerns*. Umweltministerium Mecklenburg-Vorpommern, 1 edition.

Zhou, B., Khosla, A., A., L., Oliva, A., and Torralba, A. (2016). Learning Deep Features for Discriminative Localization. *CVPR*.