

FACULTY 03: MATHEMATICS/COMPUTER SCIENCE

Master's Thesis in Digital Media

Semiautomatic Detection and Measurement of Marine Life on Underwater Stereoscopic Photographs Using a CNN

Author: Matriculation Number: First Examiner: Second Examiner: Submission Date:

Yvonne Jenniges 4526791 Prof. Dr.-Ing. Udo Frese Prof. Dr. Philipp Fischer 16.11.2020





Declaration

I hereby confirm that this thesis is my own work and that I have not submitted it for any other examination purposes. I have not used any sources or resources other than those which are indicated. All passages that are taken verbatim or in spirit from publications have been marked as such.

Erklärung

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbstständig angefertigt und nicht anderweitig zu Prüfungszwecken vorgelegt habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel verwendet. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen sind, sind als solche kenntlich gemacht.

Bremen, 16.11.2020

Yvonne Jenniges

Abstract

The need for automation technology in the field of marine monitoring is exemplified by the Alfred-Wegener-Institute, Helmholtz-Centre for Polar and Marine Research (AWI). The institute maintains an underwater observatory off Spitzbergen, which consists of sensors and a stereo camera system. The effort for the evaluation of the image pairs produced daily is high: The processing comprises two programs that enable the manual labelling of organisms and their measurement. Since a full automation by a neural network failed so far, this work presents a semi-automatic solution for the annotation process. For this purpose, this thesis is two-parted. On the one hand, the existing network is improved by the use of heatmaps instead of segmentations. Thus, the number of false positives can be reduced significantly though the detection ability still does not suffice for a full automation. On the other hand, a software called MarOMarker is developed which enables the correction of the network output. The program is implemented using User-Centred Design methods and improves the current workflow of annotating organisms at the AWI. In terms of measured usability, the software achieves a score of 89.58 on the System Usability Scale.

Kurzfassung

Der Bedarf an Autmatisierungstechniken im Bereich der biologischen Meeresüberwachung zeigt sich beispielhaft am Alfred-Wegener-Institut, Helmholtz-Zentrum für Polar- und Meeresforschung (AWI). Dieses unterhält ein Unterwasserbeobachtungszentrum vor Spitzbergen, das u.a. ein Stereokamerasystem beinhaltet. Der Aufwand für die Auswertung der täglich prodzuierten Bildpaare ist hoch: Sie umfasst zwei Programme mit deren Hilfe manuell Tiere markiert und vermessen werden. Eine Vollautomatisierung mithilfe eines neuronalen Netzes ist bisher gescheitert, weshalb in dieser Arbeit eine halbautomatische Lösung entwickelt wird. Deshalb ist die Masterarbeit zweiteilig. Einerseits wird das exisiterende Netz durch die Anwendung von Heatmaps statt einer Segmentierung verbessert und andererseits wird die Software MarOMarker implementiert, die die Korrektur der Netzwerk-Ausgabe ermöglicht. Das Netz konnte im Vergleich zum Vorgägner die Anzahl an false positives signifikant verringern, erkennt aber nicht genügend Tiere für eine Vollautomatisierung. Die mittels User-Centred Design Methoden entwickelte Benutzeroberfläche kann den aktuellen Arbeitsablauf zur Organismenmarkierung am AWI verbessern und erreicht im Bereich der wahrgenommenen Benutzbarkeit eine Wertung von 89.58 nach der System Usability Scale.

Contents

Al	ostrac	ct	iii						
Kι	urzfas	ssung	iv						
1.	Intro	oduction	1						
	1.1.	Goals	1						
	1.2.	Structure	2						
2.	Stat	e of the Art	3						
	2.1.	Machine Learning	3						
		2.1.1. Neural Networks	4						
		2.1.2. Convolutional Neural Network	5						
		2.1.3. Transfer Learning	5						
		2.1.4. Encoder-Decoder Architecture	7						
		2.1.5. Evaluation Metrics	8						
	2.2.	User-Centred Design	10						
		2.2.1. UCD Methods	10						
		2.2.2. UCD Methods in Software Engineering	13						
	2.3.	Marine Organism Detection on Stereoscopic Images	14						
		2.3.1. Related Work	14						
		2.3.2. The Underlying Bachelor Thesis	14						
3.	Des	Design, Implementation and Evaluation of the Neural Network							
	3.1.	Equipment	16						
		3.1.1. Hardware	16						
		3.1.2. Software	16						
	3.2.	Data	17						
		3.2.1. Data Acquisition	17						
		3.2.2. Dataset Generation	17						
		3.2.3. Test Train Validation Split	18						
	3.3.	Neural Network	20						
		3.3.1. Data Preparation	21						
		3.3.2. Minimal Neural Network	21						
		3.3.3. Transfer Learning Using MobileNetV2	23						
		3.3.4. Network for all Classes	25						
		3.3.5. Encoding the Connection between Heads and Tails	27						
	3.4.	Training and Validation	29						

Contents

	3.5.	Post Processing
	3.0.	Evaluation
		2.6.2 Network for all Classes
		2.6.2. Inetwork for all classes
	27	S.O.S. Connection Encoding
	3.7. 20	Comparison to Longer's Wark
	3.8.	Comparison to Jasper's work
4.	Desi	gn, Implementation and Evaluation of the User Interface 46
	4.1.	Equipment
		4.1.1. Hardware
		4.1.2. Software
	4.2.	Initial Situation
		4.2.1. User Description
		4.2.2. Current Workflow
		4.2.3. Positive Aspects and Potential Improvements
	4.3.	Requirement Analysis
		4.3.1. Functional Requirements
		4.3.2. Non-Functional Requirements
	4.4.	Personas and Scenarios
		4.4.1. Scenario for Jonas Hipp
		4.4.2. Scenario for Susanne Fuchs
	4.5.	Task Analysis - MarOMarker 55
	4.6.	Prototype
		4.6.1. Digital Paper Prototype
		4.6.2. Prototype Structure 58
		4.6.3. Test Persons
		4.6.4. Conduction
		4.6.5. Evaluation
	4.7.	Design Realization
		4.7.1. Task: Set user ID and camera configuration
		4.7.2. Task: Load data and Set Output File
		4.7.3. Task: Run Neural Network
		4.7.4. Task: Edit Images
		4.7.5. Task: Match Animals on Left and Right Images
	48	Usability Evaluation 68
	1.0.	481 Usability Questionnaire
		482 Test Structure 68
		483 Test Persons 60
		484 Results 60
		10.1. 1.Could
E	Con	214.com

6.	Outlook	74				
A.	Interview Protocols	76				
	A.1. Expert Interview with Prof. Dr. Philipp Fischer, Head of the AWI Center for Scientific Diving (17.03.2020)	76				
	A.2. Interview Guide	78				
	A.3. Interview with User C (30.06.2020)	79				
	A.4. Interview with User D (29.07.2020)	85				
B.	Hierarchical Task Analyses	97				
	B.1. Macro Scheduler	97				
	B.2. Stereo Marker	98				
C.	Prototype	100				
	C.1. Slideshow	100				
	C.2. Test Protocol Proband A	112				
	C.3. Test Protocol Proband B	115				
	C.4. Test Protocol Proband C	123				
D.	Usability Questionnaire	128				
E.	Tables	131				
Lis	List of Figures					
Lis	List of Tables					
Ac	Acronyms					
Bi	Bibliography					

1. Introduction

Monitoring marine ecosystems provides biological insights that are crucial to detect changes and evaluate the impact of direct human actions, like pollution and overexploitation, as well as global processes, like climate change and ocean acidification. [1] The gained information presents valuable input for biologists, but also for economical and political stakeholders to e.g. decide on measures for fisheries and ecosystem protection. [2]

There is a range of projects dealing with monitoring marine environments, one being the Coastal Observing System for Northern and Arctic Seas (COSYNA) [3]. Within this framework, the Alfred-Wegener-Institute, Helmholtz-Centre for Polar and Marine Research (AWI) maintains an underwater observatory off the Norwegian archipelago Spitzbergen. It comprises of multiple sensors and a stereo camera system intended for abundance surveys and organism size measurements. The two cameras produce 48 image pairs per day that are manually evaluated in a two-step procedure: First, all organisms are roughly classified and localized on the photos of the left camera. In a second program, head and tail positions are specified on both images enabling the measurement of organisms. Since the labelling task is time-consuming and monotonous, an automation of the process is of practical use. The underwater photographs present multiple challenges: light attenuation, non-uniform lighting, shadows, colour reduction and suspended particles [4]. The present dataset not only exhibits varying lighting conditions and low contrast, but also alternating backgrounds. Therefore and because deep learning outperforms traditional computer vision methods in many areas including object detection [5], machine learning approaches are favoured for this task.

An attempt to automate the annotation process was already taken by Jasper who implemented a neural network applying semantic segmentation. Besides finding positions, it coarsely classifies the organisms into fish, crustacea, chaetoghnatha, jellyfish and unidentified. The positional errors are however rated too large to allow for a fully automatic system. [6]

1.1. Goals

The aim of this thesis is the development of a semi-automatic solution for marine organism detection and classification. Therefore, the thesis consists of two parts: the development of a neural network to predict position and class of organisms and the realization of a User Interface (UI) to enable the correction of the neural network predictions.

For the neural network, this works has the goal to investigate the performance of a Convolu-

tional Neural Network (CNN) that uses a pre-trained MobileNetV2 as backbone and heatmaps to encode position data. Also, the question will be answered if this approach can improve the predecessor network by Jasper [6]. Because the available training images differ in basic properties from the images used for pre-training the backbone, four ways of incorporating the backbone into the training process are examined by freezing and unfreezing layers. Thus, the most beneficial way of integrating the backbone will be determined. The heatmap encoding raises the problem that organism heads and tails need to be matched since they are computed separately. To solve the matching task, three different strategies are investigated: coarse segmentations, vector fields and a distance-based measure in post-processing. Moreover, the effects of class weights are tested to compensate for the imbalances in the present dataset that contains mostly fish and crustacea, but only few examples of the remaining classes. The influence of two different numbers of training epochs on the realized networks is another factor that will be studied.

For the UI, the main purpose is the implementation of a usable product that supports a fast workflow and is easily learnable since the software is supposed to be used frequently by an alternating user group. To ensure the accomplishment of the usability goal, the User-Centred Design (UCD) approach will be selected to conduct the development process.

1.2. Structure

Initially, relevant background information about machine learning, including a summary of the predecessor work by Jasper [6], and about UCD is described in Chapter 2. In Chapter 3, the development process of the neural network is depicted beginning with the utilized equipment and the dataset. The chapter continues by presenting the various network architectures and configurations tested within the scope of this thesis. After denoting the post-processing, the models are evaluated and compared leading to a final prediction pipeline that produces an output in the desired format. The results are then briefly compared to Jasper's findings. Chapter 4 is dedicated to the UI. As the neural network chapter, it starts with outlining the equipment. Then, the workflow that is currently conducted at the AWI to label the organisms is analysed yielding, along with interviews, requirements. With personas and scenarios, a general work process is established from the user perspective which is refined in a task analysis of the future program called Marine Organism Marker (MarOMarker). The collected information subsequently forms the base for a prototype test whose results are used to adapt the design. The design is then implemented and evaluated in a final usability test and a questionnaire. Lastly, the neural network and UI parts are summarized and an outlook is given.

2. State of the Art

This chapter covers relevant foundations of both scientific fields, machine learning and UCD. For machine learning, neural networks and convolutional layers are explained, as well as the principle of transfer learning along with the backbone examples of VGG16 and MobileNetV2. Moreover, the encoder-decoder architecture is presented as a way for neural networks to output images at full resolution. Afterwards, metrics relevant for the realized neural networks are defined. For UCD, the general development strategy is described besides specific methods applied in the field. Afterwards, the integration of UCD techniques in software engineering is outlined. Lastly, some related work is introduced, including the predecessor work on the problem of automatising the organism annotation process.

2.1. Machine Learning

The field of machine learning deals with algorithms which learn from data. Learning is a process during which the performance conducting a specific task improves. Mitchell formulates it as follows:

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." - T. M. Mitchell [7]

Examples of typical machine learning tasks T include classification, regression, translation and anomaly detection. [8] In order to evaluate and improve the performance on a task, a measure P, also called metric, is employed, e.g. accuracy or error rate. As input for a task, a dataset serves as a base for the knowledge acquisition, the experience E, of the algorithm. The type of the dataset allows for a coarse categorization of the learning algorithms into unsupervised and supervised: If the dataset contains annotations, i.e. predefined labels for its data points, the algorithm is said to be supervised, otherwise unsupervised. This affects the calculation of the loss, the measure used to iteratively improve the network, since supervised algorithms can take the desired output into account, contrary to unsupervised ones. [8] Figure 2.1 illustrates the general scheme for supervised learning: The network receives the i-th input from the training data x_i and predicts the output \hat{y}_i . The loss function then evaluates the performance of the network by comparing the computed output \hat{y}_i with the desired output y_i . Summed up over all data points, an optimizer minimizes the loss and updates the parameters θ of the neural network accordingly. [9]



Figure 2.1.: Supervised Learning (adapted from [9])

2.1.1. Neural Networks

The algorithm that maps the input data \overline{X} , where $\overline{X} = [x_0...x_n]$ with n features to the output data \hat{y} , is called neural network. Inspired by neuroscience, the basic building block of such a network is called perceptron (Figure 2.2). It contains one weight w_i per input x_i . Firstly, each weight is multiplied by the respective input, resembling the biological structure of dendrites. Secondly, the results are summed up and added to a bias term. Lastly, the sum is passed to a non-linear element resulting in the prediction \hat{y}_i or, in terms of neuroscience, the axon. The biological neuron is represented by the summation and non-linear step. Mathematically, the perceptron is equivalent to a linear regression followed by a non-linearity. One perceptron forms the smallest possible neural network with one input layer and one



Figure 2.2.: Perceptron (adapted from [10])

output layer. Combining multiple perceptrons results in intermediate layers, commonly referred to as hidden layers. Adding more hidden layers increases the depth of the network while adding more units to a layer increases its width. Networks that have a large amount of layers are called deep neural networks. [10]

4

2.1.2. Convolutional Neural Network

CNNs are utilized often when working with image data and are therefore the foundation of the neural networks developed in this thesis. This type of network deals with grid-like structured data which, in the case of images, is a 2D-pixel-grid. The major difference from other networks is that CNNs incorporate at least one convolutional layer.

Convolutional Layer

In a convolutional layer, the linear operation of convolution is applied to the data. The intuitive understanding of the operation is the successive shift of a filter over an image. For the example of a 2D image *X* as input and a 2D kernel *W*, their convolution results in a tensor *Y*, which, in the scope of neural networks, is also called feature map:

$$Y(i,j,c') = (X * W)(i,j) = \sum_{k} \sum_{l} \sum_{c} W(k,l,c,c') X(i+k,j+l) + b(c')$$
(2.1)

where

k,l	$\in [-rac{f-1}{2},+rac{f-1}{2}]$	indices to iterate over the kernel
С	$\in [0, C - 1]$	with <i>C</i> the number of input channels
с′	$\in [0, C'-1]$	with C' the number of output channels
Χ	$\in \mathbb{R}^{w imes h imes C}$	input image with w and h its width and height,
W	$\in \mathbb{R}^{f \times f}$	kernel
Y	$\in \mathbb{R}^{i imes j}$	feature map
b	$\in \mathbb{R}^{C'}$	bias

The entries in the kernel matrix W, i.e. the weights, as well as the bias b, are learnable parameters of the neural network. Thus, the total number of parameters computes to $CC'f^2 + C'$ and the computational cost to $whCC'f^2$. [8, 9]

2.1.3. Transfer Learning

Traditional machine learning methods require a large amount of input data to generate plausible predictions. A challenge that can prove difficult to overcome is the gathering of enough data for a specific task, which is also true for the present application. Transfer learning addresses this problem by reusing knowledge from one task in another related task. [11] This can have positive effects on the initial performance, the speed of learning, as well as the final performance [12]. A practical example for transfer learning is the usage of an existing pre-trained model to serve as starting point for a model with a similar purpose. Since the pre-trained model has already acquired knowledge about basic features, the new model needs less data and resources for the training process. For image classification, a range of pre-trained models are available. The deep learning API Keras [13] offers e.g. DenseNet, VGG16, VGG19, MobileNetV2 and ResNet50, all trained on the ImageNet dataset [14]. In this work, VGG16 and MobileNetV2 become relevant and are therefore described in the following.

VGG16

In 2015, Simonyan and Zisserman from the Visual Geometry Group, University of Oxford, described the VGG architecture for image classification used in the ImageNet challenge the year before. It incorporates a number of 3 × 3 convolution-Batch Normalization (BN)-Rectified Linear Unit (ReLU) blocks and five maximum pooling layers in between, each halving the resolution. Along with the downsampling, the channel number is doubled, starting from 64 and finishing at 512. Three fully connected layers form the end of the network. The authors experimented with different numbers of layers carrying weights, the two best-performing ones, VGG16 and VGG19, containing 16 and 19 weight layers respectively. Figure 2.3 depicts the VGG16 architecture, which achieved a top-1 accuracy of 74.4% on ImageNet. Moreover, Simonyan and Zisserman found that their models have good generalization abilities to other datasets. [15]



Figure 2.3.: VGG16 Architecture (based on [15])

MobileNetV2

MobileNetV2 is another neural network for image classification. It is especially designed to run on mobiles or devices with limited resources. The main idea to reduce memory load is the usage of the so-called *Inverted Residual with Linear Bottleneck* introduced by Sandler et al. in 2018 [16]. The module separates channel-wise and spatial processing of convolutional layers by replacing a normal convolutional layer with a 1×1 convolution for the channel calculations and 3×3 depth-wise convolution for spatial computations. The architecture is displayed in Figure 2.4. Similar to VGG16, pooling follows five times on a number of computing units. However, the computing units of MobileNetV2 make use of the previously mentioned split of the convolutional layers hidden in the Inverted Residual (IRes) blocks: An IRes block consists of a 1×1 convolution with BN and ReLU6 increasing the channel number by factor t, followed by a 3×3 depthwise convolution with BN and ReLU6. Lastly, a final 1×1 convolution that returns to the initial number of channels *c*. Moreover, a shortcut directly connects the beginning and the end of the IRes element (Figure 2.5a). The Pool-IRes





Figure 2.4.: MobileNetV2 Architecture (adapted from [9])

block additionally performs stride pooling right after the depth-wise convolution. It skips the shortcut and uses the last convolutional layer to build the desired number of output channels c' (Figure 2.5b). The output is a classification on 1000 classes. On ImageNet, MobileNetV2 achieves a top-1 accuracy of 74.7%. [16]



Figure 2.5.: MobileNetV2 blocks

2.1.4. Encoder-Decoder Architecture

There are different ways to encode data, depending on the application of the neural network. The previously described architectures depict image classifiers outputting classes and their probabilities for a given image. However, for some tasks, an image output is required. This is the case for instance for semantic segmentation where a class is assigned to each pixel and for heatmaps where classes are localized by making the related pixels brighter. Such output can be generated by replacing the fully connected layer(s) at the end of classification networks like MobileNetV2 and VGG16 by convolutional layers. Typically, the output is then only in 1/32 of the input resolution due to the pooling steps. The encoder-decoder architecture

addresses this topic and presents a way for networks that perform image downsampling to produce image output at full resolution. The encoder phase performs downsampling whilst increasing the number of channels to detect features in the image. For MobileNetV2, this corresponds to the original network structure with a convolution operation instead of the fully connected layer. In the head or decoder phase, the resolution is increased incrementally again. Therefore, a network that aims for a full resolution output and downsamples in five stages with the encoder will also have five stages with the decoder. The downsampling layers alternate with convolutional layers, taking the upsampled image as input as well as information from the encoder in the respective resolution, right before the pooling operation (shortcut). The shortcuts ensure that the upsampling takes the existing knowledge about the relevant resolution stages into account. [9]

Encoder with Heatmaps

One application area for the encoder-decoder structure is detection tasks by calculating heatmaps. Here, the output is a coordinate position rather than an object outline, like it is e.g. computed with bounding boxes. The basic idea is to encode position information on an image by creating an empty, i.e. black image that is painted white where the seeked objects are localized. The hence generated heatmap contains values in the range [0,1] providing information about the probability of the presence of a class per pixel. In post-processing, coordinates can be extracted from the heatmaps for instance by applying thresholding and non-maximum suppression. [9]

2.1.5. Evaluation Metrics

Depending on the type of machine learning problem, the performance of the implemented model during training, validation and testing can be measured by various metrics. Relevant metrics for this work are described in the following. For the output of continuous variables, the Mean Absolute Error (MAE) and Mean Squared Error (MSE) are two common evaluation measures as depicted in [17]. For classifiers, which predict class labels for an input sample, performance measures include confusion matrix, accuracy, recall, precision and F1 score. The description of the classification metrics are taken from [18].

Mean Absolute Error

A very basic performance measure is the absolute difference between groundtruth and prediction, averaged over the data points. The MAE is calculated as follows

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$
(2.2)

where *n* is the number of data points.

Mean Squared Error

The MSE squares the distance between groundtruth and prediction before summing and averaging. Hence is penalizes outliers stronger than the MAE.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$
(2.3)

where *n* is the number of data points.

Confusion Matrix

For a binary classification problem, i.e. one which has two possible output classes, a positive and a negative class, the prediction on a sample can be one of four results: a correctly predicted positive class (True Positive (TP)), a correctly predicted negative class (True Negative (TN)), a wrongly predicted positive class (False Positive (FP)) and a wrongly predicted negative class (False Negative (FN)). Their count can be presented in a binary confusion matrix (Table 2.1). For multi-class classifications, this matrix is extended such that the number of rows and columns equals the class count. The matrix gives information about the predicted classes and their actual Ground Truth (GT) class.

	GT positive	GT negative
Predicted as positive	TP	FP
Predicted as negative	FN	TN

Table 2.1.: Confusion matrix for binary classification (from [18])

Accuracy

A typical classification metric is accuracy which equals the ratio between correct predictions and the total number of predictions (Equation 2.4). However, this measure is susceptible to imbalanced data and can deliver the same result for differently performing models, e.g. if the values for TP and TN counts are switched.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$
(2.4)

Precision

The proportion of positive predictions that are classified correctly is called precision (Equation 2.5).

$$precision = \frac{TP}{TP + FP}$$
(2.5)

Recall

Recall, also known as sensitivity or hit rate, represents the ratio of correct positive classifications from the amount of all positive samples (Equation 2.6).

$$recall = \frac{TP}{TP + FN}$$
(2.6)

F1 Score

The F1 score is calculated as the harmonic mean of recall and precision (Equation 2.7). It is sensitive to the class distribution.

$$F1 = \frac{2 * precision * recall}{precision + recall}$$
(2.7)

2.2. User-Centred Design

Applications are developed for users to enable them achieving a goal with a positive experience. For example, Microsoft PowerPoint enables people to achieve the objective of creating a presentation in a structured and clear way. The intention to design and implement an application that meets the needs and requirements of its users gave rise to the UCD methodology. The UCD approach involves the user throughout the whole development process to ensure a high usability of the outcome tailored to its target group. [19] According to ISO 9241-210, the four main phases of UCD are understanding the context of use, specifying user requirements, producing design solutions and evaluating the design. Thus, a design that addresses the users' needs and provides a positive experience is iteratively inferred. [20] These definitions and procedures are not only valid for software, but also for general products.

2.2.1. UCD Methods

There is a wide range of methods available for UCD processes facilitating different goals. Seven important ones, namely interviews personas, scenarios, prototypes, usability guidelines and questionnaires are explained in the following and will be employed in the present work.

Interviews

UCD often requires the conduction of interviews in some form, for example when moderating usability studies, modeling scenarios and use cases, as well as collecting data about users, tasks and workflows. Wilson categorizes interviews into three fundamental types: structured, semi-structured and unstructured [21]. They differ in the way topics are discussed with the interviewee(s). Structured interviews consist of a fixed set of questions in a fixed order which is especially useful for collecting demographic information and opinions and to compare

results across groups. Semi-structured interviews follow a coarse question guide giving the opportunity for explorations. This allows both sides, the interviewer and interviewee, to add new subtopics to the discussion, clarify issues and hence promote comprehension. An unstructured interview is a free conversation about one or more predefined topics which allows to gain insights into experiences, emotional topics and previously unknown issues. [21]

Personas

Typically, an application is used by a range of different users who have distinct needs and goals. Personas are prototypical individuals, each representing one user type. Specifying the goals and tasks of the personas supports the determination of product functionalities and behaviour. Also, they facilitate discussions about design decisions and keeping the focus on the user throughout the development. [22]

Scenarios

Imaginative situations in which personas interact with the product under development are called scenarios. In this narrative way, creativity in the design process is encouraged. Similar to personas, scenarios also leverage discussions and idea sharing due to their intuitive storytelling nature, which finally supports requirement definition, as well as interaction design. [22]

Task Analysis

Understanding the way people perform a particular task can be gained through a Task Analysis (TA). Thereby, the task is broken down into its individual steps and sub-steps. Such an analysis is helpful to understand behaviour, tasks, as well as the knowledge necessary to process them, to assign tasks, compare systems and inform design. Depending on the goal of the TA, the tasks and subtasks can vary in level of detail. A common method in the field of Human-Computer Interaction (HCI) is the Hierarchical Task Analysis (HTA). It enriches the task decomposition with plans that specify the sequence and conditions for each task. [23]

Prototypes

Prototypes are a tool to communicate ideas on a product, test and improve them. Thus, understanding of the users, the addressed problem and the suggested solution is gained. The latter can hence be improved iteratively. Depending on the specific goal of the prototype, the fidelity of the prototype is selected, which indicates its level of similarity to the final product. Generally, there are three fidelity levels, but they can also be mixed. Low fidelity prototypes aim for testing core concepts of the design and user flows at the cost of restricted interaction possibilities. Typical examples include paper prototypes, wireframes and clickable prototypes. Mid fidelity prototypes include more context and can be used to test more detailed interactions and are realized e.g. as clickable and coded prototypes. High fidelity prototypes

are completely visually designed, filled with real content and are mostly interactive. Their goal is the testing of very detailed design parts and the design presentation to stakeholders. [24]

Usability Guidelines

For the design of interfaces, many guidelines have been developed that provide rules from a general until a specific level. These usability guidelines support the development of a usable UI. [25] Here, the focus is on collections of general usability principles providing advice for essential features. The most popular ones among them are Nielsen's ten heuristics [26], Shneiderman's eight golden rules [27] and Norman's seven principles [28]. The references pertain the most recent versions of the principles. [29]

In a direct comparison, Norman's list is more general and can therefore also be used outside of interface design. Shneiderman and Nielsen, on the other hand, explicitly refer to the latter subject area: Nielsen originally developed his heuristics for the heuristic evaluation of UIs, while Shneiderman pursued the intention of summarizing the key principles of interface design. Nielsen and Shneiderman are hence more relevant for this work because of their subject relevance and direct applicability. The two sets of rules contain many parallels, such as avoiding errors and reducing the amount of information that the user has to remember. Since principles for the design phase are seeked and Nielsen originally aimed for evaluation, Shneiderman's set of rules is described in the following and enriched with two of Nielsen's aspects.

- 1. *Strive for consistency* of the design in similar situations, e.g. in terminology, layouts and menus.
- 2. *Seek universal usability* to cover the requirements of different users, i.e. take disabilities, age, culture, level of expertise with the system, etc. into account.
- 3. *Offer informative feedback* as response to every user action, adapted to the level of frequency and importance of the action.
- 4. *Design dialogues to yield closure* such that the user knows when a group of action sequences is completed and can shift concentration to the next task.
- 5. *Prevent errors*. If this is not possible, provide simple error messages with recovery solutions.
- 6. Permit easy reversal of actions to reduce anxiety to try out unfamiliar features.
- 7. *Keep users in control* to avoid unexpected system behaviour that could lead to frustration. The system should respond to the user's actions.
- 8. *Reduce short-term memory load* because human information processing is limited to seven plus/minus two pieces of information (rule of thumb).

In addition, Nielsen's second principle, "Match between system and the real world", asks the designer to apply concepts that are familiar to the user and order information logically. His last heuristic is "Help and documentation" and asks for the simple retrievability of such even though the system should be usable without it.

Questionnaires

Questionnaires deliver mostly quantitative results, in contrast to e.g. interviews. They serve either the purpose of user and context analysis or of evaluating system usability. Depending on the task, different types of questionnaires can be deployed, for instance custom or standardized ones. [25] Standardized questionnaires offer many advantages, like replicability and comparability to other studies. For the assessment of perceived usability at the end of a study, the most widely used ones according to [30] are:

- Questionnaire for User Interaction Satisfaction (QUIS)
- Software Usability Measurement Inventory (SUMI)
- Post-Study System Usability Questionnaire (PSSUQ)
- System Usability Scale (SUS)

2.2.2. UCD Methods in Software Engineering

In software engineering, the development of an application mainly consists of five tasks that are executed according to the employed project management model: analyze, model, specify, realize and evaluate. Each of the steps can be enriched by UCD techniques.

During the analysis phase, the context is examined in which the software is to be embedded. Business modeling evinces fundamental framework conditions for the new solution. UCD complements this procedure by adding information about daily work, the people performing the task and their characteristics. The aim is to identify the needs and background of the users, e.g. by observations, interviews, focus groups or TAs. Moreover, the requirements collection is extended by those inferred from the UCD techniques. In the modeling phase, the new system is modeled and successively specified, for instance by defining the functionality and behaviour of the system. In this step, UCD introduces personas and scenarios to present the system from the user's perspective. This serves as a base for use-case models. Additionally, storyboards and interface prototypes support discussions between the stakeholders. The outcome of this phase, i.e. the description of the program to develop, is formally recorded in the specification phase. The specifications are transformed in the realization phase into a software architecture and a design. Design guidelines support the process of creating a consistent and usable UI. The evaluation can be done e.g. by software testing. UCD proposes inter alia usability tests, questionnaires and heuristics for this phase. [25]

2.3. Marine Organism Detection on Stereoscopic Images

Stereo imaging techniques, i.e. photo and video tools, are applied in marine sciences, mainly for abundance and habitat surveys, but also for other purposes, like fish monitoring, examination of fish school structures and sea floor surveys. [31] Compared to other monitoring methods, like nets and traps [32], it provides the advantage of being non-invasive, allowing e.g. the observation of endangered species. Also, it facilitates continuous recordings, as well as measurements of organisms by providing information for depth calculations [31].

The desired information about organism dimensions, abundance, etc. are retrieved by image processing steps. Since the manual animal labelling is time-consuming and hence costly, there are several works attempting to automate the underwater detection and classification of animals with machine learning algorithms. Two of them are outlined in the following besides the predecessor work.

2.3.1. Related Work

Li et al. implemented a Fast Regions with CNN features (Fast R-CNN) for the detection and recognition of fish species. They used videos of the LifeCLEF fish task from which they extracted relevant frames for the training. As output, bounding boxes are generated. With this approach, they achieved a mean Average Precision (mAP) of 81.4%. [33] Similarly, Villon et al. create bounding box outputs and use video frames for training the CNN. However, they manually cut thumbnails out of the frames, each containing only one fish. Additionally to eight fish species classes, they introduce three more classes: background, part of fish and fish. The background class compensates for the varying and complex natural backgrounds, while the part of fish class aims to make the network focus on complete fish instead of details. Some unknown fish species are added to the general fish class enabling the network to recognize fish of different species, too. The amount of training data for this class is kept below the one of specific species to keep the probability of assignment of a fish to a specific species higher than to the general fish class. [34]

2.3.2. The Underlying Bachelor Thesis

In contrast to the previously mentioned work, the present task requires the detection of coordinates instead of bounding boxes to ensure a precise measurement of the animals. In his Bachelor thesis "Erkennen und Vermessen von Meereslebewesen auf Stereo-Kamerabildern mit einem neuronalen Netz" [6] Jasper addresses this problem and explores the possibility of a full automation of the organism annotation process at the AWI. In order to classify organisms and detect positions of their heads and tails, Jasper implemented a neural network that performs semantic segmentation. Moreover, he realized a post-processing algorithm that assigns an animal on the left image to its counterpart on the right image by using the stereo-camera configuration information and block-matching logic. From the matching, animal lengths are computed automatically. The results show that the predictions of the

neural network exhibit a position error too large to allow a full automation. Therefore, he suggests testing different data encodings which is realized in the present thesis by heatmaps. The general animal classification, on the other hand, worked well, especially for groups represented often in the dataset, i.e. fish and crustacea. For improving the other groups, i.e. jellyfish, chaetognatha and unidentified, Jasper suggests more training data. Lastly, he concludes that the matching algorithm performs well enough to replace the manual length measurement step. [6]

3. Design, Implementation and Evaluation of the Neural Network

The first part of this thesis revolves around the development of a neural network addressing the automated recognition of heads and tails of marine organisms, as well as their classification. This chapter begins by describing the utilized hardware and software components before exploring the data, which is provided by the AWI and forms the base of the neural network training and testing. Afterwards, various architectures and configurations of the network are depicted and training configurations and post-processing steps are explained. The networks are then compared and evaluated in the evaluation section. Finally, the complete prediction pipeline as inferred from the previous analysis is summarized and its results are compared to Jasper's outcome.

3.1. Equipment

For the development of neural networks, different hardware and software requirements need to be considered. The utilized equipment, including computer components and programming libraries, are described in the following.

3.1.1. Hardware

The neural networks are trained and tested on a computer provided by the University of Bremen. The computer called Hilbert is equipped with a quad-core Intel(R) Core(TM) i7-4770 processor with a clock frequency of 3.4GHz. It provides 32GB RAM and a NVIDIA GeForce GTX Titan GK110 graphics card with 6GB graphics memory. The graphics card is especially useful since it allows for the training process to be conducted more efficiently.

3.1.2. Software

For the neural network development, the programming language Python is used with Keras [13] as deep learning API and Tensorflow as backend. The exact specifications on all utilized libraries and their versions can be found on the submitted data carrier in the folder NeuralNet in requirements.txt.

3.2. Data

The data forming the base of the neural network development is provided by the AWI Center for Scientific Diving. This section provides explanations about how the AWI obtains the data and how the present dataset is generated from this. Afterwards, the split into test, train and validation data is performed, before the data is prepared such that it can serve as input for the neural network.

3.2.1. Data Acquisition

The data generated by the AWI are underwater images along with position and species annotations of the shown organisms. The photos are taken within the framework of the COSYNA [3] project for which the AWI maintains three underwater observatories: one off Helgoland, Germany, one in the Eckernförder Bucht, Germany and one off Spitzbergen, Norway. The camera system installed in Norway as part of the underwater observatory Remote Optical System 1 (RemOS1) [35] is a stereo system, i.e. consisting of two cameras. Both cameras take a picture with a resolution of 4272×2848 pixels at the same time with a maximum time offset of 5ms. This results in one image from the left and one image from the right camera of the same scene. Usually, the system is triggered every half an hour resulting in 48 image pairs per day. Exceptions from this can be caused for example by wading work or technical errors.

After taking the photos, they are post-processed in multiple steps. Initially, they are timenormized, i.e. the corresponding left and right images are set to the exact same time and stamped digitally with time and location information. This simplifies further processing and allows to trace the photo back to its origin without the need for, but not replacing, a proper name of the photo. The conduction of the next step requires the usage of the software Macro Scheduler. Here, rough positions and general group information of photographed organisms on all left images are manually annotated. Lastly, the software Stereo Marker is used to mark the exact positions of organism heads and tails and to match the organisms on the left to organisms on the right image. Thus, the length of the animals can be measured using the stereoscopic camera configuration. In addition, the exact organism species can be added to the specification.

3.2.2. Dataset Generation

The procedure described in the previous section already generated and is still generating a large database of image pairs and respective annotations. This section describes which extract of them Jasper used in the predecessor work [6] and how he adapted the annotations. Since his dataset is used in this thesis, it needed to be altered to suit the present application. These changes are explained subsequently.

The dataset underlying the neural network training, testing and validation of this work

originates from the RemOS1 and were taken at a water depth of approximately eleven meters in the time period from 01.10.2017 until 16.01.2018. Due to the manual annotation being performed only on the left images, they serve as dataset for the machine learning task excluding the right images.

After undergoing the post-processing steps, Jasper reworked the annotations because many objects were labelled to be incomplete. Also, for a large amount of organisms, head and tail positions were almost identical and not in compliance with the actual image. Since he worked with a segmentation algorithm, he manually added segmentation information to every organism. The data processed in this way contains 28 different species classes with very different occurence frequencies. While for instance the fish species Gadus morhua (Linneaus, 1758) would be represented around 950 times, the Bolinopsis infundibulum (O.F. Müller, 1776) jellyfish was only sighted once. In order to have enough samples per class for training a neural network, the animals are assigned to five classes in [6]: fish, crustacea, chaeotognatha, jellyfish and unidentified, which are referred to as groups in this work. One row in the resulting label file contains information about an animal, its head and tail position, its group and segmentation information. To control the ratio of empty images, i.e. those not containing any animal, during training, testing and validation, Jasper created another label file with the images exhibiting no animals.

While the predecessor thesis works with segmentation data, for the planned network structure, only animal head and tail positions are required besides the group information. So, the desired columns are extracted to start with. Images that are empty, i.e. do not show any organism, are omitted from the training data. The reason for this is that the images with organisms already contain large empty areas from which the algorithm can learn what "no organism" means.

The high resolution of the photos proved to be problematic for the network training in two ways. Firstly, conducting the training on the GPU of the available hardware limited the maximum batch size to one, slowing down the training. Secondly, the size of the animals on the images vary widely from approximately 5 to 880 pixels. This is due to the different size of organisms but also to different distances to the cameras. In order to utilize transfer learning, e.g. with MobileNet, large objects would hardly be recognized because MobileNet was trained on structures of about 200 pixels, the input images from ImageNet having a size of 224 × 224 pixels. Therefore, the image resolution is reduced by 25% to 1088 × 736 pixels such that most animals are smaller than 200 pixels with the trade-off that very small organisms might not be captured.

3.2.3. Test Train Validation Split

The data was already split into test, train and validation sets by Jasper. He assigned every tenth image with animals to the test data and every 100th to the validation dataset. Similarly, the images without animals were split. Thus, he argues, the datasets represent the original

	Fish	Crustacea	Chaetognatha	Jellyfish	Unidentified
Number of images with	79	90	9	7	4
Total number of	183	191	11	9	5
Average number per image	0.19	0.19	0.01	0.01	0.01
Average head-tail distance	197.46	68.83	102.55	162.45	33.25

Table 3.1.: Test data statistics (rounded to two digits)

	Fish	Crustacea	Chaetognatha	Jellyfish	Unidentified
Number of images with	710	739	83	59	56
Total number of	1718	1493	109	85	69
Average number per image	1.25	1.09	0.08	0.06	0.05
Average head-tail distance	189.81	65.97	153.51	138.76	587.12

Table 3.2.: Training data statistics (rounded to two digits)

	Fish	Crustacea	Chaetognatha	Jellyfish	Unidentified
Number of images with	38	50	5	5	2
Total number of	68	100	5	17	4
Average number per image	0.89	1.31	0.07	0.22	0.05
Average head-tail distance	157.24	72.09	144.79	118.97	46.4

Table 3.3.: Validation data statistics (rounded to two digits)

class distribution independent of the season.

Due to the unavailability of the original validation dataset, this work generates it by subdividing the training data with animals. Validation data is used to follow the learning process and to tune hyperparameters of the model. To keep the class ratio and validate on all classes, 5% of the training images were taken from every class for validation. The small validation ratio is chosen because of the limited amount of data and the focus on comparing different models rather than tuning hyperparameters. Since images with multiple animal groups appear multiple times, duplicates are removed from the final validation dataset and entries allocated to both, validation and train, are removed from train.

This procedure results in the test, train and validation datasets summarized in tables Table 3.1, Table 3.2 and Table 3.3. The test dataset consists of 983 images showing a total of 399 animals, most of them fish and crustacea. On the 1376 training and the 76 validation images, the class distribution is expectedly similar.

3.3. Neural Network

The task to solve is the recognition of organism heads and tails on the given images. Moreover, a rough species assignment to fish, crustacea, chaetognatha, jellyfish or unidentified is required. Therefore, the problem is a combination of position detection and classification.

For the encoding of the position information, heatmaps are chosen. Jasper applies semantic segmentation to the same problem with the result of position errors too large for an automation and a high number of false positives [6]. Bounding boxes would be another option, but this leaves the question open of the exact head/tail positions necessary for length measurements, as well as for the determination of swimming directions. This is the advantage of heatmaps: After post-processing, coordinates can be generated. Moreover, a heatmap is a concept close to the functioning of a CNN due to its image-like structure which might be advantageous for the detection task. The class information will be encoded in channels. This means that the network will have one heatmap and therefore one output channel per class. The problem with the heatmap strategy is the assignment of heads and tails, i.e. the question of which head belongs to which tail.

Since an image output is required for heatmaps and downsampling is performed, an encoder and a decoder stage are necessary. To start with, the encoder is specified. For this, the task is reduced to the recognition of one class: fish heads. The minimal CNN described in Section 3.3.2 serves as a baseline model which is compared with four variations of transfer learning using MobileNetV2 as backbone from Section 3.3.3. Thus, the most beneficial option is determined and selected as encoder configuration in the subsequent experiments. Afterwards, the number of classes is increased to the desired count and a decoder stage is attached to the selected encoder. On this model, the influence of class weights is examined. The architecture and variations are depicted in Section 3.3.4. Lastly, the head-tail assignment problem is addressed by implementing three different approaches presented in Section 3.3.5: a provisional segmentation, vector fields and a closest match strategy. The various architectures require different GT encodings which are shown in Figure 3.1.

3.3.1. Data Preparation

Before the collected images of quarter resolution can serve as input for the neural networks, they need to be prepared. At first, they are scaled to the range [-1, 1] because this is a prerequisite of MobileNetV2, which is used as backbone. Additionally, padding is applied to the images such that their height and width is dividable by 32. This is necessary since the images are downsampled throughout the network by this factor. The padding with black pixels does not make a difference for the learning because in the applied heatmap encoding zero-valued, black pixels are counted as background.

3.3.2. Minimal Neural Network

Starting with a small neural network or a simple, established architecture is common practice to gain insights about the task to perform and can prevent complex errors. Therefore, the task of predicting the position of different organisms and different body parts is reduced to the recognition of fish heads for a minimal CNN. The implementation bases on the VGG16 architecture due to its successful performance on localization and classification and its use of basic layers.

Configuration and Architecture

For the simplest version of the network, one input data point equals an image x_i and its heatmap denoting the position of fish heads y_i . Since the images are in the RGB colour range, there are three input channels. The desired output is a heatmap that indicates fish heads, entailing one output channel. Binary-crossentropy serves as loss function because the network needs to distinguish between two classes "fish head" and "no fish head". The activation function of the last layer is set to sigmoid such that the network outputs a probability, i.e. a number between zero and one. In this case, it is the probability of a fish head at a position on the image. This probability forms the heatmap.

The architecture (Figure 3.2) follows the general style of VGG16: Computing units alternate with maximum pooling layers halving the resolution until a downsampling by factor 32 is achieved. One computational block consists of a 3×3 convolutional layer, followed by a BN layer and ReLU activation. The step-wise downsampling is preferable over one large downsampling step because for each step, features of different spatial levels can be detected. Taking a downsampling factor of two per resolution stage, five stages are necessary to achieve a total factor of 32. Because of the minimalistic approach and the few features to identify, the number of channels is only increased to eight throughout the network. Similarly, the number



(a) Head heatmap



(b) Tail heatmap



(c) Segmentation heatmap



(d) Head vector field

(e) Tail vector field

Figure 3.1.: GT encodings of test image 60 for fish (red points/blue crosses show head/tail coordinates, vector fields only show non-zero vectors)

of computational blocks per resolution stage is kept minimal and only increased towards the end to enlarge the receptive field.



Figure 3.2.: Architecture of the minimalistic CNN

Receptive field

The size of the receptive field of the network should cover the main part of a fish. The reason is that seeing the complete animal, the network has more indications to distinguish between different animal groups, here between fish and no fish. Moreover, the head is part of a body and therefore positioned relative to it, also supporting the network in finding the fish-head pattern. Considering the above described architecture without the additional computational blocks in resolution stage four and five, i.e. with only one computing unit per stage, has a receptive field of 142×142 pixels not covering the larger animals. Additionally, according to Luo et al. [36], the effective receptive field is smaller than theoretically. To hence capture middle-sized to large organisms, the receptive field is increased to 254×254 pixels by adding one convolution-BN-ReLU block after each, the second to last and the last resolution stage.

3.3.3. Transfer Learning Using MobileNetV2

Due to the relatively small data set, transfer learning could be a useful approach to train the neural network and improve recognition. For this, MobileNetV2 is chosen as backbone because it achieves good results on ImageNet classification and COCO object detection on the one hand and is computationally efficient on the other [16], so that the training of its layers can also be tested with the available hardware. In Keras, MobileNetV2 is available pre-trained on ImageNet.

Configuration and Architecture

Since the present network generates only a one-, later in this work an eleven-class output while MobileNet works with 1000 classes, an increase in the number of channels to 1280

towards the end of the network is not necessary. Therefore MobileNet is cut off after the last IRes block, which results in 320 channels. To retrain the network for the fish problem, a custom block is attached, which is based on the structure of a block in MobileNet: It complies with the structural separation of channel- and spatial-calculations, as well as the increase of channels in the middle. However, the custom block prepends a conv-BN-ReLU6 block to provide a layer to set the desired channel number before increasing it (see Figure 3.3a). As



Figure 3.3.: Architecture of the networks for fish heads using transfer learning

for the minimalist approach, the final output is achieved by a 1×1 convolution with sigmoid activation. The binary crossentropy loss is also adopted because the basic task remains same. Figure 3.3 shows the resulting network structure.

Variations

One challenge of the present data set is the lack of contrast, which distinguishes the available images from the ImageNet ones. Therefore, the created network is trained in four different ways elaborated in the following.

- 1. *No layers*: All layers of MobileNet are frozen in order to investigate whether the feature recognition of MobileNet together with a custom trainable block is comparable with the minimalist network.
- 2. *Last layers*: A common approach is tested in which the upper layers of the backbone are trained in order to adapt the more semantic features to the respective problem. For this, the last 28 out of the 150 backbone layers are unfrozen, corresponding to the three blocks after the last downsampling block.
- 3. All layers: Due to the lack of contrast, it is conceivable that the basic features of the

images already differ from those generated using ImageNet. This leads to the third approach, in which the entire backbone is trained.

4. *All layers delayed*: It is also tested whether it is advantageous to train the backbone for 50 epochs only after ten initial epochs during which the backbone is frozen (which results in a total of 60 epochs). This could be helpful because MobileNet already starts with very good weights while the custom part has not yet learned. This difference could lead to a strong gradient in the upper part, which could superimpose and negatively affect the training of the backbone.

3.3.4. Network for all Classes

After a basic architecture experiment and four transfer learning trials with networks that only work on heatmaps for fish heads, the prediction task is expanded to the remaining groups, i.e. jellyfish, crustaceans, chaetognathas and unidentified objects. For this, each organism group is divided into two classes, heads and tails, resulting in a total of ten classes. Also, a background class is appended. In addition to the thus specified encoder, a decoder stage is appended upsampling the image to half of the input resolution (368×544 pixels).

Configuration and Architecture

From the previous architectures, transfer learning and unfreezing all backbone layers after ten initial epochs proved to be the most promising method (see Section 3.6.1). Hence, it serves as encoder stage for the architecture for all animal classes with a few adjustments. First, the binary crossentropy loss with sigmoid activation is replaced by categorical crossentropy with softmax activation. This is important in order to receive output probabilities dependent on each other, i.e. to teach the network that one pixel cannot be two classes at the same time, e.g. fish head and a jellyfish tail. With binary crossentropy, the network assigns each pixel the probability only of the respective class, such that multiple class assignments are possible. For categorical crossentropy, the network has to be additionally provided with a background heatmap which is bright everywhere but at the other heatmaps' peaks and considered as eleventh class. It is necessary to ensure that the network is not forced to assign a background pixels to a group. This leads to the second adjustment: the raise of the number of channels. The output layer, which is supposed to return eleven heatmaps, needs eleven channels. Moreover, the channel number in the custom block should not fall below the desired number of output channels to prevent information loss. However, it should not be too large either to avert a large amount of weights. The channel number in the custom block is therefore set to eleven, too.

The decoder stage consists of a sequence of upsampling layers, custom blocks and shortcuts. Due to limitations of the available hardware, the images can only be upsampled to half of the input resolution. For a heatmap output, this is sufficiently precise because the aim is to identify only rough regions where a class is located on the image. Figure 3.4 shows the

complete architecture, starting with the encoder and finishing with the decoder below it. The shortcuts are indicated by red arrows.



Figure 3.4.: Architecture of the network for all classes

Variations

The unbalanced classes are an unsolved topic of the present dataset impeding the training of a network that also takes the minority classes into account. Jasper already encountered the problem during his work on the automatisation of marine organism detection [6]. There are multiple solutions, e.g. more data and class weights. The collection of more data was not possible within the given time frame because the raw data needs a significant amount of post-processing before it can be used. Class weighting is expected to improve underrepresented classes but can also cause over-fitting during training [37]. For this approach, Keras provides a parameter in its fitting function to set a weight for each class which, however, does not work for 3D data input like the given images. Sample weights would be another option to introduce weights, i.e. assigning a weight to every sample image. Here it is critical if multiple animal groups are present on an image because it is unclear which class weight should be chosen. Hence, the selected approach to put more emphasis on under-represented classes is a custom

loss function that calculates the desired weighted categorical crossentropy (Equation 3.1).

$$loss = -\sum_{channels} y_true * log(y_pred) * weight$$
(3.1)

where *weight* equals a 2D-tensor of the same size as the heatmaps and is filled with the weight of the class determined by the current channel. One class weight is computed as follows:

$$weight_{class} = \frac{num_samples_{max_class}}{num_samples_{class}}$$
(3.2)

where *num_samples_{max_class}* is the number of samples of the class with most samples and *num_samples_{class}* is the number of available training samples for the currently computed class. The resulting class weights are presented in Table 3.4. Since heads of one group have the same incidence as tails of the same group, the class weight for both, head and tail classes, are the same per group, e.g. the classes fish heads and fish tails have the same weights. The network for all classes is trained with and without class weights to examine the differences.

Classes	Weight
Background	1.00
Fish heads and tails	1.04
Crustacea heads and tails	1.00
Chaetognatha heads and tails	8.90
Jellyfish heads and tails	12.53
Unidentified heads and tails	13.20

Table 3.4.: Class weights (rounded to two digits)

3.3.5. Encoding the Connection between Heads and Tails

When heads and tails of each group are recognized automatically, the question arises which head belongs to which tail. This is a crucial information for the present application because without this knowledge one can certainly make statements about the frequency of each organism group but it is not possible to measure animals. Here, three solutions are considered: provisional segmentation, vector fields and matching closest heads and tails in post-processing. As discussed in Section 3.6.2, the weighted model trained for 110 epochs is chosen as a final model. Therefore, it serves as a base for the subsequent tests on the connection encoding for segmentation and for vector fields, whereby the latter is trained in low resolution, i.e. only the encoder stage since this is sufficient to indicate general directions. For the closest match strategy, all four models are compared.

Provisional Segmentation

The GT contains information about complete animals but loses it in the heatmap encoding. A solution to the head-tail correspondence problem could be to encode the matching infor-

mation and enter it into the training. One example is a provisional segmentation. For this, another heatmap is generated that is bright where animal bodies are. Since there is no simple way to obtain this information, bodies are approximated by a thick line connecting head and tail, hence "provisional" segmentation.

Configuration and Architecture

Adding another heatmap, i.e. another class to the current output layer could result in competing classes at the head and tail points because the network needs to decide weather to assign the head point to the respective head class or to the segmentation. Therefore, a one-channel output layer is added operating with binary crossentropy loss and sigmoid activation, like the networks for fish heads. Due to the additional channel, the number of channels in the custom block is increased to twelve.

Vector Fields

Another way to encode the match information are vector fields. The idea is to attach a vector to every head and tail pointing to the other end of the body. If the network outputs this direction and length hints, searching the matching head/tail can be reduced to e.g. finding two vectors which point towards each other or even only finding the closest tail point to the end of the vector starting from the head. For this, two vector fields are created as further network input that initially contain zero-vectors for every pixel. The first vector field then adds vectors pointing from head to tail of all animals, while the second field adds tail-head vectors.

Configuration and Architecture

Since the vector fields do not directly contribute to the classification task, a separate output layer is built. It has four channels, two for each vector field containing the x- and y-coordinates of the vectors respectively. A MSE loss and linear activation are utilized because the output are continuous numbers and outliers should be penalized stronger than smaller errors. To compensate for the number of channels, their number in the custom block is increased to 15.

Variations

The vectors of interest are only those which originate from head/tail regions defined by the respective heatmaps: The vectors pointing from head to tail only need to be evaluated on the bright pixels in head heatmaps, tail-head vectors on the bright pixels in tail heatmaps. In order to support the neural network to focus on those relevant vectors, a weighted MSE is implemented. It requires the sum of all GT head heatmaps and the sum of all GT tail heatmaps as additional inputs, resulting in two more channels compared to the basic vector implementation. The loss uses the two summed heatmaps as weights for the vector fields: The head-tail vectors are multiplied by the heads heatmap and the tail-head vectors by the tails heatmap. In accordance with the new input channels, the number of channels in the custom block is altered to 17.

Closest Match

The last approach is a post-processing step, i.e. happens after the training and image prediction. The problem to solve is an assignment problem where a list of heads must be assigned to a list of tails at minimum cost. The cost function is typically the euclidean distance which is adapted to the present case: Since the average angle between the line formed by head and tail and the horizontal for all organisms in the complete dataset is around 41°, i.e. organisms slightly tend towards horizontal swimming, errors in y-direction are penalized more by a higher weight. Interpreting 90°as 100%, the average angle translates to about 54%. Adding the calculated weights to a euclidean distance measure leads to the cost function depicted in Equation 3.3.

$$cost(head - tail) = \sqrt{0.46 * \Delta x^2 + 0.54 * \Delta y^2}$$
(3.3)

where each, head and tail, is a 2D-vector with x- and y-component. Subsequently, the cost function is to be minimized globally, which means to take all animal heads and tails on the considered image into account for the optimization. For this, the linear_sum_assignment algorithm from the scipy.optimize library is used which bases on the Hungarian algorithm to solve the assignment problem. The disadvantage of distance-based post-processing measures is that they infer the match solely from two lists of points without regarding visual hints. Especially if animals are located close to each other, this can lead to mismatches.

3.4. Training and Validation

The training and validation for the minimal CNN and the transfer learning networks was conducted using only images that show at least one fish to focus on their task and increase the training speed. The other networks use the complete dataset because their aim is the detection of all groups.

The fitting was done with a batch size of four for the minimal CNN and the transfer learning algorithm that trained the last layers and the one that trained no layers of the backbone. Due to computational limitations, the batch size has to be reduced to two for the subsequent architectures.

Each neural network is trained with 60 epochs in the first and 110 epochs in the second run to examine the effect of training length on the respective neural network. For the high resolution networks, the decoder stage is trained with 20 epochs.

3.5. Post Processing

The output of the high resolution networks cannot serve as input for the correction software right after predicting images. The software requires a format in which animals are specified,
i.e. one row in its dataframe depicts one organism, including its group, head and tail positions on the left image. To achieve such output the following actions are taken:

- 1. The heatmaps are condensed into coordinates.
- 2. The coordinates are rescaled to the original image format.
- 3. Heads and tails are matched to shape a complete animal.

The coordinate retrieval works with thresholding and non-maximum suppression. By deleting heatmap values that are below a limit value, noise is reduced. The openCV threshold function is used for this purpose. It sets all values in the heatmaps, which are previously scaled to the range [0, 255], below 110 to zero. Then the peak_local_max function from the scikit-image library finds the local maxima and aggregates points that are closer than 10 pixels. Figure 3.5 illustrates this procedure on a GT and on a predicted heatmap. The latter results in five TPs and two FPs. In the next step, animal heads and tails are matched. The three approaches implemented in this analysis are described in Section 3.3.5 and evaluated in Section 3.6.3.



(a) Extraction on GT heatmap

(b) Extraction on heatmap predicted by weighted model (110 epochs)

Figure 3.5.: Visualization of coordinate extraction from fish head heatmaps on test image 94 (white areas are heatmap, blue crosses are the GT coordinates, red points are the coordinates extracted from the heatmaps)

3.6. Evaluation

In this section, all previously discussed neural network models are evaluated. Comparing the minimal network and the different training procedures for transfer learning dependent on the number of training epochs results in the choice of the encoder for the final model. Afterwards, the task of fish head recognition in low resolution is extended to all classes, i.e. heads and tails of fish, crustacea, chaetognatha, jellyfish and unidentified in addition to the background class, in high resolution. Here, the influence of weights and more training epochs is examined. Lastly, the three different approaches to match heads and tails are investigated, namely provisional segmentation, vector fields and a closest match approach during post-processing.

3.6.1. Minimal Neural Network and Transfer Learning

For the recognition of fish heads, the minimal CNN and the four transfer learning variations are evaluated and compared to choose an encoder for the final architecture. The training and test MAE, as well as the predictions on an example image are used for the decision.

Figure 3.6a shows the MAE evolution of the five networks over 110 training epochs. It is visible that the minimal network evens out at the second-highest level, only exceeded by the network not training any backbone layer. All other transfer learning versions attain a significantly lower score of approximately 0.003, whereby the last layers network takes longest to reach this level. The error of the all layers network starts oscillating from epoch 70 on. The



(a) MAE of 110 epochs training (logarithmic y-scale) (b) Last 20 epochs of all layers delayed model

Figure 3.6.: Training MAE of minimal CNN and transfer learning variations

lower error level of most transfer learning versions compared to the minimal CNN might be explainable by the advantage from previously gained knowledge on basic image features. Nevertheless, without training any part of the backbone, the final performance is worse than the one of the minimal network. This might be attributable to the input images which differ considerably from the ImageNet images, especially by a lower contrast. By unfreezing the backbone, it can adapt to the new conditions.

The results from the model evaluations on the test dataset can be found in Appendix E.1. They support the result that the last layers, all layers and all layers delayed trained for 110 epochs achieve the overall best results whereby the delayed training can slightly outperform the other two variations.

The MAE for the different training lengths of 60 and 110 epochs are compared in a next step.

Figure 3.6b visualizes the last 20 epochs of the all layers delayed model and shows that the network improves from 60 to 110 epochs. The other models behave similarly.

Since the longer training generally improves the models, only the predictions of the 110 epoch models are visually reflected. The predictions on test image 39 are depicted in Figure 3.7 and represent the general trend. The minimal CNN and the frozen backbone network roughly detect fish bodies. The latter produces a heatmap that is hardly visible. The model training the rear MobileNet layers generates a weak heatmap only marking one out of three fish heads. Training all layers is very similar to the delayed training though a bit weaker. Both recognize two fish heads and the tail of the last fish.

Based on the above considerations, the encoder structure is selected. Since the minimal CNN and the no layers model achieve the worst results in terms of visualized heatmap, training and test MAE, they are excluded. The last layers network is also removed because compared to the remaining two, it takes longest to reach the same training MAE, has a higher test MAE and detects less fish heads. Comparing all layers and all layers delayed reveals that the latter has a more stable performance on the training error, a slightly better test MAE, as well as slightly stronger prediction heatmaps. Therefore, the neural network training all backbone layers after a delay of ten epochs is selected as encoder for the final network.

3.6.2. Network for all Classes

After the specification of the encoder (training the MobileNet backbone after a delay of ten epochs), four experiments are conducted for the task of detecting all organism classes: an unweighted model and a weighted model, each trained once for 60 and once for 110 epochs. Thus, effects of weights and training length are studied supporting the choice of a network architecture and configuration. The models are compared and evaluated in two steps. First, the heatmap outputs are considered. In a second step, the predictions of the models are assessed after coordinates have been extracted from the heatmaps.

Heatmap Evaluation

For the evaluation of the heatmaps, the MAE during training and test phase, as well as visualized heatmaps in comparison to the GT coordinates are inspected.

The training MAE is depicted in Figure 3.8 for selected phases. As displayed in Figure 3.8a, the decrease speed of the MAE during the encoder training is similar for all four models which is expected since they all start with the pre-trained MobileNet and train it after a delay of ten epochs. It is observable that the networks without weights even out at a smaller MAE than the weighted models towards the end of the encoder (Figure 3.8a) and also towards the end of the decoder (Figure 3.8b). Using 110 epochs instead of 60 further reduces the training error in both stages. The higher training error for the weighted models could be caused by the influence of the weights on the loss: The model is forced to improve on underrepresented



(a) Minimal CNN







(c) Last layers







(e) All layers delayed

Figure 3.7.: Fish head heatmaps predicted by different 110 epochs models overlayed on test image 39 (red points show GT head coordinates)



(a) MAE for encoder stage (logarithmic y-scale) (b) MAE for the second half of decoder stage

Figure 3.8.: MAE for all classes networks from selected training phases

classes. Since there are more examples of classes, which are now less weighted and hence less learned, the total error increases.

The test MAE shows a similar trend to the one of the training MAE for the unweighted models, but differs for the weighted ones. As during training, the test MAE of the unweighted models improves with more epochs (Figure 3.9a). This is especially true for the background class, which means less FPs, and fish heads. For fish tails, the error slightly increases. Furthermore, its is observable that the error is highest for background, fish and crustacea classes. The latter observation indicates that there might be a correlation between the total number of covered pixels by a class and the error. For instance, the background class, which has the highest errors for all models, also covers most pixels. Thus, many examples are available for this class which potentially raises the probability of false assignments and consecutively the error. Contrary to the unweighted models, the weighted model experiences an overall deterioration with more epochs mainly for the underrepresented classes, which are supposed to improve by weights, and the background class. Since fish and crustacea MAE remain at a similar level for both training durations, the higher error in the background class must be caused by the worse performance of the underrepresented classes. This indicates a higher amount of FPs, i.e. the network seems to try to detect organisms from underrepresented classes but is not able to. A probable reason is the lack of available training data and its low diversity such that the network is not able to learn a generalization. Additionally, the 60 epochs model shows the smallest MAE during testing but but this can be attributed to the good recognition of the background class denoting a small number of overall detections.

Test image 51 is used for an exemplary visualization of the prediction heatmaps of the four models since it is representative for the fish class and shows many fish in different sizes and positions. The predicted fish head heatmaps can be found in Figure 3.10. For the unweighted models, it is noticeable that both achieve a similar result: out of eight heads, they





Figure 3.9.: MAE for all classes networks in testing phase

recognize five correctly. The heatmap of the longer trained model additionally contains a tail that was wrongly recognized as head. The difference between the weighted models is more significant: The 60 epoch model recognizes five heads, but the fish head heatmap also contains six fish tails. The latter disappear with prolonged training, along with one head detection. This suggests that the networks first learn to recognize the general "end of fish" case before learning the difference between heads and tails. In the case of fish, this apparently occurs more slowly with weighted models than with unweighted models, which is consistent with the poorer MAE performance of the weighted models during training.

Coordinate Evaluation

The evaluation on coordinate level starts with analysing the error introduces by the coordinate extraction procedure. Afterwards, the coordinates computed from the predicted heatmaps are assessed using established metrics for classification, namely accuracy, precision, recall, F1 score and confusion matrices.

Before the analysis is conducted, the way in which the classification metrics are calculates is explained. The predictions are classified into TPs, TNs, FPs and FNs which is done by iterating over the calculated points. If an entry in the GT labels exists which has the same class as the predicted point and is less than 30 pixels away, the point is counted as TP. A distance of 30 pixels is selected because one GT point translates to a 25 pixels spot on the heatmap. A buffer of five pixels is granted for the prediction. To avoid that multiple points can be assigned to the same GT point, the GT point is then removed from the list considered in the assignment. If no GT point of the same class is located within the 30 pixel radius, the point is a FP. After all points are assigned, the points remaining in the GT list are added to the FN count. TNs consist in all background coordinates which are not assigned to another class, i.e. the number of image pixels (739 * 1088 = 800768) less TPs, FPs and FNs. Similarly, confusion matrices are calculated with the difference that all GT entries are compared to the

3. Design, Implementation and Evaluation of the Neural Network





(c) Weighted model, 60 epochs



(b) Unweighted model, 110 epochs



(d) Weighted model, 110 epochs

Figure 3.10.: Fish head heatmaps predicted by all classes models overlayed on test image 51 (red points show GT head coordinates)

point and not only those of the same class.

To examine the error introduced by the coordinate extraction procedure consisting of thresholding and local maxima extraction, it is applied to the GT heatmaps of the complete dataset. The majority of points was assigned correctly with 7440 TPs. With a total of 4261 animals, i.e. 8522 heads and tails, the score equals approximately 87.30%. The remaining heads and tails are FNs, meaning that they were not recognized. Taking a closer look at the FNs of some example images reveals that the unrecognized organisms were those labelled twice, i.e. they can be traced back to dataset inaccuracies. In order to check this for all 1082 FNs, they are compared to the respective GT entry by finding the GT animal that is closest, excluding identity. Thus, a mean minimum distance is calculated: On average, the undetected animals are as close as approximately seven pixels to the original GT entry supporting the previous indication of doubled annotations.

3.	Design,	Imple	mentation	and	Evaluation	of	the	Neural	Network
	2001211	11110101	110111111111111111	~~~~~	L'011111111111111	~	1110	1 10000 000	110000000

	No w	veights	Weights				
	60 epochs	110 epochs	60 epochs	110 epochs			
True positives	393	430	248	395			
False positives	556	729	376	448			
False negatives	405	368	550	403			
Recall	49.24%	53.88 %	31.08%	49.50%			
Precision	41.41%	37.10%	39.74%	46.86 %			
F1 score	44.99%	43.94%	35.01%	48.14 %			

Table 3.5.: Coordinate evaluation for network with all classes on test data

The results for the coordinates generated from predictions of the four neural network experiments for all classes can be found in Table 3.5. Per class scores can be found in the tables in Appendix E. Generally, the cost of FPs and FNs is similar for this application, with a slight shift to FNs: To not detect animals is slightly worse than wrongly labelling animals because when using the correction software, the annotations will draw the attention of the user to the respective points in the image while no annotations would result in a situation similar to the one without the network: the user needs to scan the complete image thoroughly. However, too many FPs would also inhibit the work since the user focuses on deleting most of the labels instead of concentrating on the actually present animals. This means that recall is slightly more important than precision. Nevertheless, the F1 score is also a suitable metric to compare the models because the difference in cost for precision and recall is only small. The accuracy is a less meaningful metric in this case since the background class strongly outweighs all other classes. Due to this imbalance, the accuracy of all models is high. To put it differently, the number of detected animal coordinates is only a small fraction of the number of background pixels. This results in a large TN score.

In comparison to the other models, the unweighted model trained for 60 epochs has average scores. It performs similar to the weighted model trained for 110 epochs with the difference of a higher FP count resulting in a lower precision and F1 score. A longer training, i.e. talking about the unweighted model trained for 110 epochs, raises the number of correct detections by about 9% to 430. The model generally shows a high willingness to recognize organisms: It has the highest score of correctly and falsely predicted animals. This results on the one hand in a high recall and on the other in the lowest precision due to its large amount of FPs. In contrast, the weighted model trained for 60 epochs shows the lowest tendency to detect animals. The lowest amount of TPs combined with a high number of undetected animals entails the smallest recall of only 31%. The best performing model measured by precision is the weighted model trained on 110 epochs which also outperforms the others in F1 score. This means that the model best balances recall and precision, i.e. weights FPs and FNs similarly.

Based on these considerations, the weighted model trained for 60 epochs can be excluded

from the final model choice. The network shows very little tendency to recognize reflected in the smallest test error, caused by a very good recognition of the background class, and low classification scores, only improving with longer training. This finding is consistent with the training MAE observations: For this model, the error evens out at the highest level. The unweighted model trained for 60 epochs is on a similar level as the others regarding the classification metrics. Its F1 score is slightly higher than the one of the longer trained unweighted model, however, its recall, which is marginally more important than precision, is 4% lower. Therefore and due to the higher training and testing MAE compared to its longer trained version, this model is also excluded from the choice. The models that were trained longer remain. The unweighted model shows a high willingness to recognize, but thus also falsely recognizes background pixels as animals, while the weighted model finds a balance.

	Background	Fish head	Fish tail	Crustacea head	Crustacea tail	Chaetognatha head	Chaetognatha tail	Jellyfish head	Jellyfish tail	Unidentified head	Unidentified tail	Background	Fish head	Fish tail	Crustacea head	Crustacea tail	Chaetognatha head	Chaetognatha tail	Jellyfish head	Jellyfish tail	Unidentified head	Unidentified tail
Background	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Fish head	155	108	9	2	0	0	0	0	0	0	0	55	94	20	1	0	1	0	0	0	0	0
Fish tail	199	10	103	0	3	1	0	0	0	0	0	63	17	87	0	1	1	0	0	1	0	0
Crustacea head	179	4	2	109	16	3	1	1	0	1	0	114	1	0	108	4	1	0	0	0	3	0
Crustacea tail	167	0	0	3	88	0	2	0	1	0	1	78	0	0	0	85	0	1	0	0	0	3
Chaetognatha head	0	0	0	0	0	0	0	0	0	0	0	14	1	1	0	0	3	0	1	1	0	0
Chaetognatha tail	0	0	0	0	0	0	0	0	0	0	0	18	0	1	0	0	0	3	0	1	0	0
Jellyfish head	0	0	0	0	0	0	0	0	0	0	0	5	1	1	0	0	0	0	3	1	0	0
Jellyfish tail	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	2	0	0
Unidentified head	0	0	0	0	0	0	0	0	0	0	0	17	0	0	3	0	0	0	1	0	0	0
Unidentified tail	0	0	0	0	0	0	0	0	0	0	0	18	0	0	0	3	0	0	0	1	0	0

(a) Unweighted model, 110 epochs

(b) Weighted model, 110 epochs

Figure 3.11.: Confusion matrices for networks with all classes. The background detections, shown in the first row, are not counted and displayed as zero.

Consequently, the two models are examined more closely by confusion matrices (Figure 3.11) to reveal their classification performance per class. The unweighted model shows a strong tendency to only detect fish and crustacea, the animals that are most represented in the data set. The weighted model is spread a little wider and partially recognizes chaetognatha and jellyfish. Note that the maximum achievable number on the test data for chaetognatha is eleven and for jellyfish nine, which is quite low and implies that the model recognizes about 27% and 33% respectively. However, the weighed model confuses fish heads and tails more

often than the unweighted model. The lower performance in identifying fish is apparently the price for improving on the other species.

In summary, the unweighted models heavily focus on the classes with a higher occurrence, i.e. fish and crustacea. Adding epochs heightens recall, lowers precision and reduces the overall MAE during testing and training. The weighted models benefit from longer training too since the network is forced to adjust the loss to underrepresented classes and thus needs longer for reducing the error. Applying a longer training phase, the weighted model recognizes classes that are less common and shows more willingness to detect these animals than the unweighted models. Yet, the available training images for these groups do not suffice for the network to acquire a reasonable generalization ability on chaetognatha, jellyfish and unidentified. Despite this flaw, the weighted model is able to recognize some underrepresented animals in contrast to the unweighted model. This results in a preference for the weighted model.

3.6.3. Connection Encoding

After assembling a basic neural network for the recognition of all classes, the task of matching heads and tails on the outputted heatmaps remains. In order to choose a method, the three implemented approaches - provisional segmentation, vector fields and closest match - are evaluated in the following.

Provisional Segmentation

The performance of the provisional segmentation is investigated by the MAE during testing and training, as well as by an example image.

The training MAE for the heatmaps evens out at about 0.00041 which is similar to the error of the weighted model trained for 60 epochs. This indicates that the additional output layer for the segmentation decelerates the learning. The segmentation error ends at about 0.0009, more than double of the error of the other heatmaps. A reason could be that the segmentation is only rough, covers more pixels and is hence more prone to errors.

During testing, the heatmap MAE is with 0.00016 low compared to the four models for all classes. Again, the segmentation error of 0.00033 is about double of this. The low error can be explained by the large amount of empty images, i.e. containing many background pixels and few showing organisms, indicating a low ability to detect animals. The only model, out of the four compared for all classes, that has a smaller error, namely 0.00014, is the unweighted model trained for 60 epochs.

Investigating the error on test image 51 showing eight fish supports the hypothesis of a good performance during testing mainly due to few detections: The heatmap MAE rises up to 0.003, while the segmentation, fish head and tail error increase to approximately 0.008.

The background class MAE even jumps to 0.015. This is equivalent to a deterioration by a factor between 18 and 33. Figure 3.12 visualizes the segmentation, fish head and fish tail heatmaps of test image 51. It is apparent that all heatmaps look similar, i.e. the network cannot distinguish between heads, tails and bodies. A similar phenomenon is observed in Figure 3.10, where the unweighted model trained for 60 epochs is not able to differentiate between fish heads and tails. Nevertheless, the body segmentation roughly captures five fish which could serve as a base for further post-processing for the matching.

Generally, the model with the additional output layer for the segmentation shows many parallels to the weighted model trained for 60 epochs implying that the additional task slows down the training process. This is supported by the example image showing the confusion of the network about the differences between head, tail and body. Despite the decline in detection performance, the segmentation itself could deliver a starting position for head-tail matching during post-processing.



(a) Segmentation heatmap

(b) Fish head heatmap

(c) Fish tail heatmap

Figure 3.12.: Provisional segmentation of fish class in test image 51 (red points are GT fish heads, blue points are GT fish tails)

Vector Fields

The vector field approach is examined by images only because the visuals give sufficient hint that the method in the given configurations is not a suitable choice. The approaches are discussed exemplary on the predictions for test image 60.

As depicted in Figure 3.13, both, head-tail and tail-head vector fields, do not show a clear direction or differences between background vectors and vectors in the head and tail regions. In addition, the vectors are overall rather short, especially in the head-tail case. This could imply that the network does not know how and where to draw the vectors and that it learns that they are small most of the time.

In order to closer investigate the relevant vectors, the fish head heatmap is multiplied with the head-tail vector field. The same is done respectively for the tail-head vector field. Moreover, the vectors are multiplied by a factor of 10 for better visibility. Only the vector

pointing from head to tail of the visually shortest fish correctly indicates the direction. However, the vectors rather seem to follow an image-wide direction trend rather than pointing to a head or tail because per vector field, the vectors point in similar directions. Possibly, the network learned to assign a similar direction to heads and tails respectively.

For the focused vector fields, the directions are more diverse and accurate than the ones of the unfocused vector fields. But still, this is not enough for example to assign matches according to head-tail and tail-head vector intersections because the direction inaccuracies will lead to mismatches, especially for larger organism accumulations.

Closest Match

For inspecting the closest match approach, classification and three custom metrics are used as measure. Initially, the way in which the measures are computed is explained. The closest match strategy is then used on the GT to examine the error introduced by this procedure. Afterwards, the predicted heatmaps serve as input for a performance comparison of the four networks.

For the classification metrics calculation, each matched animal is compared to the GT. If a GT head of the same class can be found, that is closer than 30 pixels to the head of the matched animal and the same is true for the corresponding tail, the animal is counted as TP and the GT animal is removed from the list to avoid the possibility of a rematch. If no sufficiently close animal can be found in the GT, the calculated match is added to the FPs. As soon as all matched animals are compared, all entries remaining in the GT list increase the FN count. The TN count finally computes as the sum of pixels minus the TPs, FPs and FNs.

Additionally, the proportion of extracted coordinates that could be matched is examined by considering three custom scores. First, the proportion of all detections that can be matched is calculated by Equation 3.4: The number of matched animals is divided by the number of detected coordinates. The numerator needs to be doubled because in the matching, complete animals are counted, while in the coordinate evaluation heads and tails are counted separately. Similarly, the fraction of true detections that can be matched is computed by Equation 3.5, and the fraction of false detections that can be matches by Equation 3.6.

matched detections =
$$\frac{2 * (TP + FP)(matches)}{(TP + FP)(coordinates)}$$
(3.4)

$$matched true \ detections = \frac{2 * (TP)(matches)}{(TP)(coordinates)}$$
(3.5)

matched false detections =
$$\frac{2 * (FP)(matches)}{(FP)(coordinates)}$$
(3.6)

Testing the closest match approach on the GT coordinates is an important step in order to



(a) Head-tail vectors



(c) Head-tail vectors multiplied by fish head heatmap and factor 10



(e) Focused head-tail vectors multiplied by fish head heatmap and factor 10







(d) Tail-head vectors multiplied by fish tail heatmap and factor 10



(f) Focused tail-head vectors multiplied by fish tail heatmap and factor 10

Figure 3.13.: Vector field predictions on test image 60 (red points show GT head coordinates, blue points show GT tail coordinates)

evaluate the performance of the Hungarian algorithm with the custom weighted Euclidean distance as cost function independent from the remaining processing steps. The complete dataset - including training, testing and validation images - contains 4261 organisms. The described procedure is able to correctly match 3819 of them, equivalent to approximately 89.63% and hence most animals. As anticipated, problems arise for images showing animals close to each other which is true in particular for large organism accumulations. This is inferred from inspecting example images. Otherwise, the closest match strategy works well on the present dataset.

Applying the closest match algorithm to the coordinates extracted from the predicted heatmaps results in the scores shown in Table 3.6. The tables for per class scores can be found in Appendix E. Compared to the coordinate evaluation, all models show a decline in recall. This means that less complete animals can be found than there are detected coordinates. The inability to assign each head to a tail can be traced back to the matching process: The algorithm can only match all coordinates, if the number of heads and tails is the same. Otherwise, surplus heads or tails remain unmatched and decrease recall.

The precision generally improves, visible through the proportion of correctly recognized animals which is higher than the proportion of correctly identified coordinates. This indicates that the matching process filters out more false than true detections which is reinforced by the custom scores: The fraction of matched true recognitions outweighs the fraction of false ones for all models.

The F1 score drops for all models, but for the unweighted short-trained model. Thus, it achieves the highest F1 score, opposed to the coordinate evaluation where the weighted model trained for 110 epochs is in this position. In addition to the highest proportion of matched correct detections, it suggests that out of the found coordinates, the model can generate most correct and complete organisms.

In terms of custom metrics, the longer trained models are able to match around 80% of all coordinates and are generally able to assign more heads and tails than the unweighted models. This implies a higher number of predictions where the number of heads and tails are the same. Due to this very clear trend, it is derivable that the networks learn in later training phases that the number of heads and tails must be equal. This trend is also visible for the matched true and false detections with the exception of the first model which attains the highest fraction of matched true detections as mentioned above.

To summarize, the proportion of detected animals is between 40% and 50%. The matching process filters out some incorrectly detected coordinates which is desirable for the final outcome. In contrast to the coordinate evaluation, the unweighted model trained for 60 epochs outperforms the longer trained models in most scores but in recall, in the proportion of assigned false detections and in the proportion of all matched detections. Apparently, the

	No w	reights	Weights			
	60 epochs	110 epochs	60 epochs	110 epochs		
True positives	173	188	66	165		
False positives	142	276	98	175		
False negatives	226	211	333	234		
Recall	43.46%	47.11%	16.54%	41.35%		
Precision	54.92%	40.52%	40.24%	48.53%		
F1 score	48.52%	43.57%	23.44%	44.65%		
Matched detections	66.39%	80.07%	51.92%	80.66 %		
Matched true detections	88.04%	87.44%	53.23%	83.54%		
Matched false detections	51.08%	75.72%	52.13%	78.13%		

Table 3.6.: Closest match evaluation for network with all classes on test data

models learn just after a longer training that the count of heads and tails should be equal per class on an image.

Connection Encoding Choice

The above considerations only allow for the selection of the closest match approach to compute corresponding head and tail pairs. The segmentation heatmap itself might be able to support the matching process but not in the given configuration. The reason for this is the deterioration of the fish detection capabilities by introducing head, tail and body confusions. The vector fields approach yielded the worst results out of the three. Neither length nor direction of the vectors are sufficiently precise for the necessary output. In contrast, the closest match strategy is able to match most coordinates correctly though is struggles with organisms that are close to each other.

3.7. Final Prediction Pipeline

Based on the consideration in Section 3.6, a final prediction pipeline is inferred. For this, the general network structure is addressed, as well as the chosen encoder variant. After a summary of the epochs and weights choices, the matching strategy selection is recapitulated.

The neural network consists of an encoder and a decoder. The latter scales the output of the encoder stage back to half of the input image resolution. For the encoder, MobileNetV2 pre-trained on the ImageNet dataset serves as a backbone before a custom block. This causes the error to rapidly decline in the beginning because basic features are already learned by the backbone. After a delay of 10 epochs, all backbone layers are unfrozen to release them for training. Thus, the network can compensate for the low-contrast of the input images differing from ImageNet samples.

A prolonged training for 110 epochs generally improves performance on the one class networks, as well as on heatmap and coordinate level of the all classes networks. Moreover, weights are used to compensate for the class imbalances. This deteriorates the fish detection abilities, but forces the model to correctly and also incorrectly detect underrepresented classes, like jellyfish, chaetognatha and unidentified. This is a progress compared to the unweighted models since they do not recognize these classes at all.

For the matching of heads and tails, the closest match strategy is selected. The segmentation yielded promising results for the segmentation heatmap, but the model severely impairs for the head and tail classes. The vector fields in both configurations, normal and focused, generally do not output directions accurate enough to infer a matching. Therefore, after the model training and coordinate extraction using thresholding and non-maximum suppression, the Hungarian algorithm with weighted Euclidean distances is applied, i.e. the closest match approach. The hence generated output consists of a list of animals, each assigned to an organism group and containing head and tail coordinates on the left image of the stereoscopic camera. This serves as input for MarOMarker, the software to correct the predictions which is designed and developed in Chapter 4.

3.8. Comparison to Jasper's Work

In the following, the performance of the network as described in Section 3.7 is compared to the performance of the neural network of the predecessor work by Jasper [6]. Similar to Jasper's analysis, which implements a segmentation network for the same problem, the detection works best for the majority classes fish and crustacea. However, the number of FPs can be reduced significantly for these classes with the presented prediction pipeline: For fish, it is decreased from 702 to 59 and for crustacea from 448 to 79. Despite this improvement, the count of true detections is similar but for fish for which the implemented weighted model recognizes about 30 organisms less than Jasper's network. This is the cost for a higher probability to detect underrepresented animals. Nevertheless, the class imbalances remain a problem for the underrepresented classes in both works, Jasper's and the present one.

4. Design, Implementation and Evaluation of the User Interface

The second part of the thesis involves the development of a software forming the interface between the automatic organism detection by the neural network and the manual correction of the predictions by a user. The aim of the resulting software named MarOMarker is to replace and improve the current workflow to annotate organisms on stereoscopic underwater images at the AWI in terms of usability. The process bases on User-Centred Design (UCD) methods in order to achieve a high usability and user satisfaction.

To begin with, the development equipment is briefly outlined before the current workflow is examined to identify improvable, as well as conducive aspects by observations, interviews and hierarchical task analyses (HTAs). Afterwards, requirements are collected for the future system by processing the information about the current workflow and further interviews. The users are then characterized and translated into personas and scenarios. This leads to coarse specifications of tasks executable with the future system which are formalized in a HTA. From the task sequences and design guidelines, a design is derived that is implemented and evaluated prototypically. The evaluation reveals aspects for improvement, which flow into the subsequent realization of the software. Finally, the usability of MarOMarker is evaluated by the System Usability Scale (SUS) questionnaire following a test session.

4.1. Equipment

The equipment utilized for the development of the software MarOMarker is presented in the following. This includes hardware, as well as software components. Since the aim is the implementation of a Graphical User Interface (GUI), a library for the graphical realization is especially important.

4.1.1. Hardware

The developed MarOMarker software is implemented and tested on a laptop with a Windows 10 operating system. A two-core Intel(R) Core(TM) i5-5200U CPU is available with a clock frequency of 2.20GHz, as well as 8GB RAM and a NVIDIA GeForce 940M graphics card.

4.1.2. Software

MarOMarker is implemented with Python to enable a simple connection with calculations of the prediction pipeline. This includes pre-processing images before applying the neural network on them, the application of the neural network, as well as post-processing steps and the integration of the organism matcher and distance measurer of the predecessor thesis [6] also implemented in Python. For package management and deployment, Anaconda [38] is used. A list of utilized libraries and their versions is provided in the folder MarOMarker in requirements.txt. The PyQt5 [39] library serves as a means for the UI realization under the free GPLv3 license. PyQt5 is chosen over other options, e.g. Tkinter, because it offers a vast range of styling options, as well as adaptable and structured functionalities, like the signals and slots concept for connecting GUI elements to actions. The large number of functionalities supports expandability and integration of e.g. threads and databases. Elements are highly customizable enabling for the implementation of non-standard interactions. Moreover, PyQt5 supports multiple platforms, namely Windows, Linux, Mac OS X, various UNIX platforms, Android and iOS. In Addition, previously acquired experience with Qt in C++ is available and facilitates a fast training and implementation.

4.2. Initial Situation

At first, the initial situation is analyzed in order to become acquainted with the profile of future users (Section 4.2.1), understand current processes (Section 4.2.2), identify positive aspects and potential improvements (Section 4.2.3) and collect background information for personas and scenarios. The following three UCD methods serve this purpose: interviews, observations and HTAs.

The methods pursue different objectives. An expert interview (Appendix A.1) is conducted with Philipp Fischer, the Head of the AWI Center of Scientific Diving, and delivers, along with further, informal conversations, which could be interpreted as unstructured interviews, the user description. It also incorporates some requirements for the software to develop which are specified in Section 4.3. Since the interview follows a set of questions, but also allows for deviations, it is semi-structured. Two interviews with AWI employees (Appendix A.3, A.4) are of the same type and follow the interview guide specified in Appendix A.2. Thus, opinions can be gathered and tasks can be explored and deeper understood. The observations of users operating with the two existing programs mainly contribute to the HTAs but also reveal user-specific behaviours. The HTAs describe the current workflow and serves, along with the results from the other methods, as a base for improvements.

4.2.1. User Description

The primary users of the application to develop are those currently conducting the organism annotation process at the AWI. Their typical profile and goals are outlined in the following.

The users are mainly students on Bachelor or Master level from natural and environmental science courses, doctoral students in the same area and employees of AWI or cooperating institutes. This diverse group implies a wide age range of around 18 to 60 years. Common to all users are basic English and computer skills, as well as a scientific background. Expertise on species of underwater organisms varies from little to expert knowledge within the user group. Most employees come from European countries, mostly Germany, France, England and Sweden.

The main goal of all users is a simple and fast evaluation of the image pairs supported by an object detection algorithm. Since this is a daily task, some users aim for finishing the task as quickly as possible, while others analyse certain time periods due to biological interest. This involves e.g. questions about the frequency of certain animal species during certain seasons.

4.2.2. Current Workflow

The current workflow involves three programs: Macro Scheduler [40], Matlab and Stereo Marker [41]. Firstly, the user utilizes Macro Scheduler to coarsely determine the position and species of organisms. To be more precise, Macro Scheduler is used to manage and start a custom macro called "Remos Image Analysis Step 1". But for simplicity reasons, this step is named Macro Scheduler. In a second step, all images are rectified by a Matlab routine which is necessary for the imminent measurement of the organisms. Thirdly, Stereo Marker is utilized to actually measure the organisms.

Task Analysis - Macro Scheduler

In the first stage of roughly annotating organisms, the users starts by data management. More precisely, a number of files, directories and parameters have to be specified and confirmed before each Macro Scheduler session. After a screen calibration, the marking process begins: The user clicks on an organism, selects its species from a list and repeats this as long as there are undetected animals on the image. With a button, the user can indicate that the current image is completed. When all images for the selected day are finished, the program closes automatically. A more fine-grained task sequence is depicted by the HTA in Appendix B.1. During one session, usually the images of one day, summing up to 48, are processed.

Task Analysis - Stereo Marker

The output file of the rough annotation in step one and rectified images serve as input for the program called Stereo Marker. This step normally happens as soon as one month of data is worked off in the first step. Stereo Marker consists of two windows: the controller and the stereoview window. The former displays the data table, offers options for data loading and presents a side-by-side view of the left and right image, which is used to determine the picture section displayed in the second window. The stereoview window provides the functionality for the organism measurement by showing the previously adjudged picture section. Besides an options panel for e.g. adapting contrast and brightness, a species list including species names and example images is screened to facilitate the species choice. The workflow with this program starts with the selection of one data row, equivalent to one animal, in the controller window. The row can be manipulated and deleted directly in the table if desired. Afterwards, head and tail are annotated on both images in the stereoview window. In the HTA in Appendix B.2, the tasks are decomposed in detail.

4.2.3. Positive Aspects and Potential Improvements

The two persons questioned in the semi-structured interview, as well as other AWI employees that were contacted throughout the thesis, mostly work with Macro Scheduler. Therefore, most feedback revolves around that program. Generally, the two programs are perceived to fulfill their respective purpose, though a range of possible improvements are discovered. Before both programs, Macro Scheduler and Stereo Marker, are discussed, two remarks are added that relate to the overall procedure.

Taking the whole process into account, one disadvantage can already be stated: A total of three different programs are necessary to finally be able to perform a length measurement on the animals, partly containing redundant work, e.g. when wrongly adding an animal point in Macro Scheduler. Aggregating the programs enhances simplicity and overview and reduces error susceptibility. Moreover, an improvement idea embracing both programs arised during the interviews: a handbook containing information about which animals there are, where and when they are typically found, how they look like, including example images, as well as further notes, like "usually in groups". This would on the one hand collect and centralize knowledge. On the other hand it is helpful to inexperienced users who can in this way profit from existing expertise.

Evaluation Macro Scheduler

In the following, positive and negative aspects about the usage of Macro Scheduler discovered throughout the analysis process are explained. Furthermore, potential improvements are evinced.

The interviewees depicted the usage of Macro Scheduler as purposive, but "a bit difficult/cumbersome". It is a program fast to use once the dialogues at the beginning of the session are completed. Only one click is necessary to add an organism, and another single click to set the species. The interviews also revealed that the filters are experienced as helpful to locate animals, as well as their transfer to the next image.

Nevertheless, a number of flaws demonstrate enhancement possibilities. As already visible in the HTA in section 4.2.2, the first step, the session preparation, is a long sequence of flat tasks. It mainly consists of working off successive dialogues that alternate with input confirmations. The interviewees described this task as tedious and annoying. Moreover, a serious concern applies to the invisible annotations: A click on an animal results in the creation of a new entry in the output table, but the point is not displayed visually on the image. Thus, the user has to remember which animals have already been added which is especially difficult for images comprising a large amount of organisms. This causes the user to click multiple times on the same organism in case of uncertainty. In addition, undo functionalities are lacking. It is not possible to revert an annotation once set and also to return to the previous image. Sometimes, screenshots of previous images are kept to be able to compare the current image with others facilitating the detection of difference and hence potential animals. Another drawback is the compilation of different programs that finally enable the usage of Macro Scheduler. The external image viewer presents one example: With every new image, Macro Scheduler reopens the external viewer with the respective image. This introduces short waiting times when switching the image and the necessity to keep the viewer software updated. Another example for an external tool is the screen magnifier. One interviewee uses it to zoom into the image which is a crucial tool to find smaller animals. Since Macro Scheduler is designed to enable the editing of the images of one day, it automatically closes afterwards. In the case that the user wants to continue working, the whole session preparation must be repeated.

The inferred negative aspects lead to improvement ideas. First, the session preparation needs to be simplified and reduced, e.g. to the selection of a date in a calendar. Options like user ID and the result file location could be integrated into the program and saved for the current user. Another important point is the visualization of organism annotations and the option to remove them. Additionally, providing the functionality to open the previous image increases flexibility and can support the identification of differences between images. For the latter, a functionality opening an image in a separate window is also thinkable. Integrating a zoom feature and an image viewer would foster the stand-alone usage of the software simplifying the application by directly providing all necessary features. This ensures that all required functionalities are available, but also not more, reducing the cognitive load for the user. Lastly, supplying the user the option to close the program when desired gives the user more control about when to finish and opens the option to continue working.

Evaluation Stereo Marker

For Stereo Marker, fewer user input is available. Nonetheless, some positive and negative aspects are discussed. Subsequently, improvements are proposed.

One strong positive point is the list of species showing example images. This is a useful feature, especially for newcomers who are not yet familiar with species names and appearances. Furthermore, when matching the organism on the left to its counterpart on the right image, epilines provide a helpful orientation and restriction of the search area.

Talking about weaknesses, some buttons on the user interface are unclear to inexperienced

users, including "as new" and "clear/new". It is also not obvious, how to add an organism in case that one was not recognized during the Macro Scheduler step. Furthermore, the direct manipulation of the table is a counter-intuitive and abstract way to manipulate data compared to a visual representation directly on the image. The task itself, as established in the interviews, presents difficulties, too. For instance, the matching process is perceived as challenging, as well as the measurement of bent animals. Since the program only provides straight lines to connect head and tail, the length of bent organisms cannot be directly inferred, but has to be estimated by offsetting the position of the tail point.

A potential improvement is to organize the user interface more clearly and convert the table manipulation to a visual task. The matching process is already supported by the epilines but could profit from a redesign or an automation as suggested and implemented by Jasper [6]. In addition, the issue of bent organisms could be addressed, for instance by making piece-wise lines available.

4.3. Requirement Analysis

On the basis of the expert interview (Appendix A.1), further informal conversations, the current workflow and its potential improvements, requirements are specified. They are grouped into functional and non-functional requirements to provide a structured overview.

4.3.1. Functional Requirements

The basic functions that the program should provide are specified in the functional requirements. To begin with, it is important that input data can be specified in the software. This includes the images to be processed, the neural network that predicts the position and group of the animals, as well as a camera configuration file that is required to calculate the animal lengths. Furthermore, an output file should be generated, whose storage location should be adaptable. When evaluating an image, the user needs the functionality to correct the predictions made by the neural network about position and animal group, to add and remove animals. Species and remarks should also be editable for each organism. It is moreover important to have the option to zoom in and out on the image so that the user can recognize smaller animals manually. In addition, the program has to support the manual or automatic measurement of animals, depending on the realization possibilities. However, the development focus is on position and group correction. Finally, it is necessary to be able to navigate between the images, since the user evaluates several images per session.

4.3.2. Non-Functional Requirements

Non-functional requirements relate to the way in which the system under development should behave, as opposed to the functional requirements which specify what it should do. This includes e.g. the language, which in this case is set to English. The program must also be able to scale to different screen sizes. It will run at AWI on computers with Windows operating systems, but should also be usable on Linux and Mac. The code must be clear and offer the possibility to expand the program later with further functionalities, for instance the application of filters to the image. Furthermore, users should have the option of closing the program at any time without losing work progress. The finished output file must be saved in CSV format and should contain the following columns: file ID, object remark, group, species, LX1, LY1, LX2, LY2, LX3, LY3, LX4, LY4, RX1, RY1, RX2, RY2, RX3, RY3, RX4, RY4, length, height, image remark, status, manually corrected, experiment ID and user ID. The status indicates whether the image has already been viewed, manually corrected whether the affected organism has been corrected by the user. The head and tail coordinates on the left and right picture are to be saved under L * and R *. Finally, the requirements for the usability of the program must be mentioned, which are the focus of UCD. In the present application, three aspects are particularly important: general usability, quick learnability with little training time, since the user group is subject to strong fluctuations and changes, and efficiency, which e.g. prevents the multiple reentering of information and enables the process to be carried out quickly.

4.4. Personas and Scenarios

After the general user characterization in Section 4.2.1, as well as interviews and observations of current employees working on the annotation process, two personas are created covering the essential properties of real world users. They represent the relatively wide age span, as well as the two most typical employment types: student worker and permanent employee. The student called Jonas Hipp and the employee Susanne Fuchs are introduced in Figure 4.1. The description continues by examining their situation, goals, frustration and self assessment of certain relevant abilities.

With the knowledge about the coarse functionalities of the future system and the prototypical users, the next step addresses the question of how Jonas and Susanne could interact with the program. The resulting scenarios apply motivations, goals and abilities of the personas to the context of using the system. Thus, the developer is put in the position and perspective of the users.

4.4.1. Scenario for Jonas Hipp

Jonas has been working full-time as an intern at AWI for two weeks now. A new program called MarOMarker was launched yesterday to perform the morning organism identification and measurement task. Jonas is a little frustrated because he had only just become familiar with the old program and is now supposed to learn a new one. So today he goes to work, starts his computer, gets a coffee and opens the new program. First, he is asked to enter a user ID and to select a camera configuration file. He already knows these settings from the old program. He simply selects the previously used configuration file, enters his abbreviation and clicks on "OK". At first glance, the program looks a bit empty, but he has not selected any

4. Design, Implementation and Evaluation of the User Interface

Pers	onas
Jonas Hipp	Susanne Fuchs
21 yearsGermanStudent at University of Oldenburg, Environmental SciencesStudent trainee at AWICosmopolitan, likes traveling	49 yearsFrench-GermanMaster in biologyEmployee at AWI Centre of Scientific DivingFriendly, supportive
Jonas is an energetic student who likes travelling, especially to places close to the ocean. He enjoys working at AWI because of his love for and interest in the ocean, as well as his manual skills.	Susanne is a friendly, helpful and open employee at AWI. She is communicative and speaks English fluently, as well as French due to her French father. Hence, she can easily talk with her German and international colleagues. A good working atmosphere is important to her.
Situation Jonas studies Environmental Sciences in his third semester at the University of Oldenburg. He works at AWI since it suits his marine interests, but he prefers working in the workshop over lengthy, monotone photo evaluations.	Situation Susanne has been employed at AWI for 16 years. She is actively researching in the field of marine biology and is excited about the results that her next studies will yield.
<u>Goals</u> He wants to finish his daily task of annotating organisms quickly.	<u>Goals</u> She wants to evaluate a specific time period and write a paper about her findings with a colleague.
Frustration If there are many organisms on an image delaying him to finish the task.	Frustration If images are missing, the visual conditions on the photos are bad or if she does not have enough time to do the evaluation.
Self assessment	Self assessment
Computer skills English skills Marine species expertise	Computer skills English skills Marine species expertise

Figure 4.1.: Personas

images yet. He quickly navigates to the menu thanks to its familiar positioning in the top bar. There, Jonas navigates to the data overview, where he selects the previous day's date. Quickly checking the input file and the output file, the monthly table, the menu takes him back to the home view, which now shows a large-format picture. Jonas sees that annotations have already been made in the photo and remembers that in this program an algorithm processes the pictures first and he only has to correct the outputted labels. Jonas does not have much knowledge of programming (he only knows some basics in R) and machine learning, but he hopes that his work can be finished faster this way. He immediately sees that the algorithm has apparently recognized an animal in the lower right corner, but none is shown there. As with Microsoft Word or Powerpoint, the editing options can be found at the top of the screen, where he immediately recognizes a garbage can as a familiar symbol for deleting elements. A tooltip confirms his assumption. So he clicks on the garbage can, whereupon it changes colour. Clicking on the animal marking in the picture deletes it. On closer inspection, Jonas notices that there is a fish in the picture, hardly distinguishable from the surroundings. Fortunately, he has just seen a plus sign in the toolbar, which he now uses with ease to add the organism again. Afterwards, Jonas shifts a few markings a little, using the zoom functionality to help him with the exact positioning. He also checks the animal groups, which is quick thanks to the colour coding, as soon as he has memorized the five colours and their associated animal group. He works his way through the images and is happy when he has finished with them faster than usual. Jonas closes the program, switches off the computer and makes his way to the workshop, where he is already looking forward to being allowed to wade an underwater sensor.

4.4.2. Scenario for Susanne Fuchs

When Susanne comes to work this morning, Jonas is already in the coffee kitchen. She gratefully accepts his offer to make her a coffee and asks whether he needs any help with the new program for organism labelling. Jonas negates, adds that he is happy about the intuitive use and goes straight back to work. Susanne is currently working on a paper that she is writing together with her colleague. The aim is to analyze the distribution of marine species that were recorded by the RemOs1 cameras off Spitzbergen in the period from October 2017 to December 2017. The period should be evaluated in two weeks. Since Susanne was already working with the new program yesterday, she is already relatively familiar with it. When she opens it, she sees that her user ID has been saved. When she takes a look at the settings, she discovers that the camera configuration file has also been saved. She switches to the data view to see what day she stopped annotating the species. It is October 3rd, 2017. The corresponding image folder is already selected as the input directory, as is the output file. Since she only wants to annotate the images on which no species identification has yet been carried out, she also selects the filter option "Species undetermined". Satisfied with the quick preparation of the program, Susanne switches back to the home view, where she begins to identify the species. She has already found out that she can use the A and D keys to quickly iterate through the organisms. While she gets to the next organism with one hand, she guides the mouse with the other hand and can use it to select the animal species from a list. Once she

is done with an image, she is curious to find out how well the algorithm's stereo matching works. Since she is already familiar with the shortcuts, she uses them to first switch to the image on the right and then to a view that contains both images. After she is satisfied with the positioning of the marks on both images, she wants to edit the next image. The program is similar to a photo viewer, which is why Susanne intuitively uses the arrow keys to perform the action. After two hours of productive work, Susanne stops satisfied.

4.5. Task Analysis - MarOMarker

The collected requirements, as well as the scenarios show the basic tasks that should be executable with the system. Before a design can be developed, these tasks are specified further. In this way, the necessary processes can be translated step by step into a design. This specification is carried out again with a HTA due to its adaptability of the level of detail and its intuitive, process-related form.

The resulting HTA, which is described below, is longer than the HTA of the existing programs, which was to be expected since the software under development combines the tasks of both and adds automation possibilities. Furthermore, some tasks need to be carried out only infrequently because the program takes over settings from the previous session when it is restarted. This includes e.g. adapting the user information and the camera configuration.

The operational sequence of the program should roughly include data loading, followed by the application of the neural network to it, the correction of the predictions and the assignment of animals on the left to those on the right image. Based on the given animal positions and camera configuration, the animal length can then be measured automatically. Generally, the functionalities are spatially distributed to a home, data and settings page: On the home page, the main annotation process is conducted. The data page provides options for data management, while the settings page for more global configurations, like the camera parameters and the user information.

0. Annotate multiple images

- 1. Find and edit user information
- 2. Find and edit camera configuration
- 3. Load data
- 4. Set output file
- 5. Run neural network
- 6. Edit current image
- 7. Open next/previous image
- 8. Match animals on left and right images
- 9. Exit the program

Plan 0: 1 if not set yet, 2 if not set yet, 3, 4 if not set yet, then 6 - 7as long as there is a next/previous image, 8, 9 if desired

- 1. Find and edit user information
 - 1.1 Click on the user profile button
 - 1.2 Choose menu option "Settings"
 - 1.3 Find the section "User"
 - 1.4 Edit user information

Plan 1: 1.1 or 1.2, then 1.3, then 1.4 if needed

- 2. Find and edit camera configuration
 - 2.1 Choose menu option "Settings"
 - 2.2 Find the section "Camera"
 - 2.3 Edit camera settings
 - 2.3.1 Load a configuration file
 - 2.3.2 Manually specify camera parameters
 - 2.3.2.1 Save configuration Plan 2: 2.1 2.2, then 2.3 if needed
- Plan 2.3: 2.3.1 if desired, else 2.3.2
- Plan 2.3.2: 2.3.2.1 if desired
- 3. Load data
 - 3.1 Choose menu option "Data"
 - 3.2 Select the date of the images to edit
 - 3.3 Set an image input directory
 - 3.4 Adapt data settings
 - 3.4.1 Change data filter option
 - 3.4.2 Change image prefix
 - 3.4.3 Change experiment ID
- Plan 3: 3.1, 3.2 or 3.3, 3.4 if desired

Plan 3.4: 3.4.1 if desired, 3.4.2 if desired, 3.4.3 if desired

- 4. Set output file
 - 4.1 Choose menu option "Data"
 - 4.2 Set a result file
- Plan 4: 4.1 4.2
- 5. Run neural network
 - 5.1 Choose menu option "Data"
 - 5.2 Click button to run neural network
- Plan 4: 5.1 5.2
- 6. Edit current image
 - 6.1 Choose menu option "Home"
 - 6.2 Add image remark

6.3 Add an image filter

6.4 Check predictions of the neural network

6.5 Correct organism

6.5.1 Add organism

6.5.2 Correct organism position

6.5.3 Correct organism group

6.5.4 Delete organism

6.6 Add information to organism

6.6.1 Specify organism species

6.6.2 Add remark to the organism

Plan 6: 6.1, 6.2 if needed, 6.3 if needed, 6.4, 6.5 as long as there is an incorrect organism, 6.6 if desired

Plan 6.5: 6.5.1 if organism is missing, 6.5.2 if organism is wrongly located, 6.5.3 if organism is assigned to the wrong group, 6.5.4 if organism is wrongly recognized

Plan 6.6: 6.6.1 if species classification is desired, 6.6.2 if desired

8. Match animals on left and right images

8.1 Choose menu option "Home"

8.2 Switch to view showing both, left and right, images

8.3 Click on an organism on left image

8.4 Click on the corresponding organism on the right image

Plan 8: 8.1 - 8.2, 8.3 - 8.4 as long as another organism needs to be matched

4.6. Prototype

On the basis of the collected requirements and HTA of MarOMarker, a first prototype is developed under consideration of the usability guidelines described in Section 2.2.1. It is evaluated with three test persons who have different levels of experience with marine species and with the current annotation process. The conduction and evaluation of the prototype aims for revealing design deficiencies, understanding how different users interact with the system and hence revealing interaction problems.

4.6.1. Digital Paper Prototype

Before realizing the prototype, its type and fidelity is determined. Then, the means of conducting the specified type are reflected and how the moderator and user can interact.

Usually, the testing process starts with a low fidelity level, like a paper prototype, to reveal improvement potential on a core level. A paper prototype would be a suitable choice for the present use case as it is basic, flexible and can be used for the relevant dragging and clicking interactions. Due to the currently ongoing Corona-crisis, however, a personal meeting for conducting this type of prototype is not possible. Wireframes and clickable prototypes on the other hand are too limited in their interaction possibilities for the present application, where e.g. dragging plays an important role. Therefore, the paper prototype concept is ported to a digital version which stays as close as possible to its physical counterpart. For this, a presentation software serves as a common platform to work on, replacing the physical table, while the addable objects within the software and images substitute the hand-drawn paper snippets. This raises the fidelity level because of the realization on the final device (a computer screen) but can still be used to meet the initially formulated goals.

For the cooperation between moderator and user, the collaboration tool USE Together [42] is chosen as a platform for conducting the prototype. It enables writing access of both parties to a shared screen. In addition to simultaneous work possibilities, the program displays multiple cursors and indicates click actions by adding a red circle next to the cursor that performed the click. Being able to observe user interactions in this way is crucial because it enables the moderator to react to and evaluate the user's actions.

Besides the choice of the collaboration tool, the organization of the elements of the prototype needs to be considerate as well. The limited work area of one screen for both, moderator and user, impedes hiding elements from the user. To clarify which elements the user can interact with, they are placed on the slide in the presentation software. Elements that appear in the process of the prototype conduction are placed besides the slide. Thus, the work area for the user, the slide, is explicitly separated from the work area of the moderator, the area besides the slides.

Another aspect of concern is the execution of user interactions. Here, the goal is to disrupt the thought processes and actions of the user as little as possible while at the same time giving enough hints about the activities of the program. For this, a simple speech concept is introduced: If an action by the user causes the program to react, the moderator says "action" when the reaction starts and "ok" when the moderator has finished the execution of the reaction. Thus, the user knows if an action triggers a program reaction, when it is finished and when he/she can continue.

4.6.2. Prototype Structure

Based on the framework conditions defined in the previous section, a digital version of a paper prototype is realized. It consists of a slideshow (complete slideshow in Appendix C.1) guiding the test person through four tasks. The tasks reflect the typical workflow of loading data, applying the neural network, correcting the outputted animal positions and groups and finally performing the length measurement.

Task One: Start Program

In order to become familiar with the action-ok-pattern and the way of interaction with the prototype, the first task asks the user to open the program:

"Open the program MarOMarker. Edit the Welcome-Screen."

The initial situation is a screenshot of a desktop with an icon of the MarOMarker software to be opened. After clicking on the icon, the start screen of the software is shown on the second slide.

Task Two: Load Data

After the program is opened, the home screen is shown (Figure 4.2) and the images of interest need to be loaded and analyzed by the neural network. This is the goal of task two:

"To start, you need to load the image data of the 01.06.2019 and apply the neural network to it. Get comfortable with the application, look around. Tell me, what you are thinking and doing."

For this, the test person has to navigate to the data page of the software.



Figure 4.2.: First slide of task two: empty home page

Task Three: Correct Positions and Groups

The third task asks the proband to correct the predictions of the neural network:

"Now you are ready to correct the animal positions and groups. You do not need to add a species or a remark to an animal, that is optional. Adding a remark to the image is also optional. Tell me, what you are thinking and doing."

This task, like the others too, takes up on the previous task. It hence starts with the edited data page (Figure 4.3a). After the user navigates to the home page (Figure 4.3b), the correction process can begin. To cover the use cases of moving existing positions, adding and removing animals, as well as adapting the animal group, several animals are placed on the image: The head of fish one is detected slightly offset from its target destination, fish two is wrongly marked as jellyfish, fish three and the crustacea are not detected and one fish is detected where nothing is visible.



(a) Slide one: filled in data page

(b) Slide two: home page

Figure 4.3.: Task three: correct positions and groups

Task Four: Perform Length Measurement

Lastly, the length measurement needs to be performed in task four:

"For the last step, the measurement of the animals, the positions and groups of all animals on BOTH images have to be corrected. Moreover, the animals on the left image need to be matched to their equivalent on the right image. With this information, the program can calculate the length automatically. Tell me, what you are thinking and doing."

Starting with the screen showing the corrected animals on the left image (Figure 4.4a), the test person needs to navigate to the screens displaying the right image (Figure 4.4b) and both images (Figure 4.4c). On the right image, one fish is marked correctly, while the heads and tails of two animals are switched, one fish is marked as jellyfish and one annotation is drawn where no animal is visible. On the left-right screen with both images, the current matching connects fish one on the left (upper) image and fish two on the right (bottom) image.

4.6.3. Test Persons

For the testing of the prototype, three probands (Figure 4.5) are chosen according to the user profile. The test persons' age ranges between 27 and 54 years to mirror the previously observed age span. Moreover, two of them do not have any experience with the existing annotation software, while one works irregularly with Macro Scheduler and has not used Stereo Marker yet. The low-level annotation experience is common since a frequently reconfigured user group performs the annotation task: especially temporary employees, like trainees and students on different levels, but also permanent employees. Mostly, the users utilize Macro Scheduler and only rarely Stereo Marker. They are hence often not familiar with the latter software. Thus, the selected people are representative for the user group.



(a) Slide one: corrected organisms on left image

(b) Slide two: uncorrected organisms on right image



(c) Slide three: receivable matching step

Figure 4.4.: Task four: correct positions and groups (right image) and perform matching

4.6.4. Conduction

The three probands are asked to individually participate in a USE Together meeting with the moderator, who grants the proband remote access to her laptop. The prepared slides for the prototype are shown on the screen. The presentation serves as a platform to work on and at the same time as a classical presentation used to guide the user through the prototype test. Initially, personal information about the user is collected. Then, information about the prototype test process is given, as well as context information about the purpose of the annotation process, the concept of stereoscopic imaging and its use for the length measurement of animals. After the test person performed the first task to become familiar with the way of conduction, an overview about the tasks is given. Then, the proband is asked to process the remaining three tasks. Finally, around ten minutes are taken for remarks, open questions and discussion points. During the test, the moderator keeps in the background and gives additional information only if necessary, e.g. if the test person is stuck, has a question concerning the context or about the prototype procedure. The language used to communicate during the prototype is adapted, if possible, to the user's native language to increase comprehensibility. The whole process took around one hour per test person. Protocols of the sessions can be found in Appendix C.2, C.3 and C.4 whereby the trigger



Figure 4.5.: Probands for prototype

words "action" and "ok" were ignored. They also exclude the discussion parts which are integrated in Section 4.6.5.

4.6.5. Evaluation

The prototype test with the three test persons reveals features for which the design supported an intuitive usage and those that need some experimentation to understand their functioning. Moreover, aspects are discovered that are confusing to the users forming the base for design adaption ideas.

The prototype test reinforces most design decisions. All probands perceived the process of correcting predictions of the neural network as intuitive. For example, the test persons immediately applied dragging to correct animal positions. Moreover, the concept of a bounding box to indicate the current animal, as well as setting the current animal by clicking on it, did not need an explanation. This is also true for the adaption of animal attributes, i.e. group, species and remark using the animal-specifications-widget. Uncomplicated too was switching between images and between image modes, i.e. display left image, right image or both images. Interestingly, the encoding of a circle for the organism head and a cross for the tail was not questioned and unconsciously implemented without the explicit knowledge about it. The test persons could not explain in the discussion why and how they got to the conclusion that the encoding is not the other way around. Hence, all probands applied the right encoding when correcting positions and adding animals. Consciously using this knowledge to recognize that the labels of two animals on the right image were switched was more difficult and immediately done by only one test person. One possible explanation for the rather unconscious usage of the encoding is that the first image already shows two animals with the correct labels, i.e. a circle on the head and a cross on the tail. Moreover, the discussion confirmed that the probands agreed on the utilized encoding to be the intuitive one.

Besides those very intuitive features, some became clear after first use. Among them are the colours representing animal groups, as well as the location and the operating principle of

the add- and remove-mode. The button for the add-mode, a plus sign, was confused several times with a zoom functionality. Also the principle of activating a mode, adding/removing as many animals as desired and then deactivating the mode again was initially not understood. For instance, trying to drag a label into the bin, i.e. the icon of the remove-animal-button or first selecting an animal and then pressing the remove-animal-button were common first approaches to delete an animal. Nonetheless, after some attempts, all test persons figured out those functionalities and stated during the discussion that the features are simple to use once understood.

The test also reveals weaknesses of the design. Especially the data selection process and length measurement raise uncertainties. Finding the location where to adapt the input data was a simple task for the probands. The test person with more experience annotating images also directly used the calendar to select the data. The less experienced ones struggled with the task, one finally using the calendar by using the principle of exclusion and remembering the task to work on a specific day, one ignoring the calendar and setting the image directory and result file manually. The latter strategy is a common pattern in many applications: In order to load data, select it in the file explorer. For the data loading task, knowledge about the way the image data is stored is an advantage and supports the more abstracted calendar approach. The other problematic process was the length measurement, more precisely, where to perform it and how to match animals on the left image to animals on the right image. Two ways to ascertain where to measure the length were clicking "Length" in the animal-specifications-widget and switching to the view showing both, left and right image. Once found, the probands liked the view showing both images and found it helpful for the matching, but it was still unclear how to exactly process the task. Even after some trials, only one proband finished the task quickly and easily but not without confusions. The desire to manifest and confirm a match resulted for two probands in the idea of adding IDs to the animals and match these.

In order to improve the design and support a more intuitive usage, some design decisions are revised on the base of the prototype test. Firstly, the usage of the calendar to select data is adapted. In accordance with Shneiderman's eights rule, "Reduce short-term memory", the information about the image directory, result file and experiment ID derived from the date can be hidden, e.g. by disabling the area until a date is selected. Thus, the calendar usage is the only possible option. Moreover, a label requesting to select a date could be helpful. Secondly, the animal matching requires improvement. The main reason for the confusion in this task was that there is no confirmation for a match. The visual representation for the matching process is the same as for the selection of an animal, i.e. bounding boxes. The indication is missing that in the view showing both images more than a selection is happening. This can be done e.g. by a dialogue asking if this is the correct match, by briefly highlighting the current match or adding an ID to each animal. Thirdly, the location where the length measurement happens was unclear. During the discussions, the probands mentioned a similar problem with the transition from selecting data to correcting the predictions. To be

Shortcut	Functionality
1	show left image
2	show right image
3	show view with left and right image
а	go to previous organism
d	go to next organism
+	(de-)activate add mode
-	(de-)activate remove mode
left arrow	open previous image
right arrow	open next image
all arrows	navigation within the image when zoom is applied

Table 4.1.: Keyboard shortcuts on home page

more precise, they would not have known where to correct the images once data is selected and annotations are predicted. The buttons for running the neural network and for correcting the predictions are perceived as a great help for the navigation and provide an overview of parts of the annotation process. This leads to the idea to have a general guide for the whole procedure on the data page starting from data selection, over running the neural network, checking its predictions, matching animals on left and right image and finishing with length measurement. For each step, information and options are displayed. For example, the first step could be "Select a date" showing the data selection options. Such a guide gives the user an overview about the whole process, as well as information and control of each step. This also improves the navigation process by adding a button "Perform matching on Home screen (LR)" similar to the check-predictions-button in the respective step.

4.7. Design Realization

Based on the insights and results gained from the prototype testing, the design is adapted and implemented. An executable of MarOMarker tested on Windows 10 is enclosed on the data carrier. In this section, general design decisions and specific design implementations to some key tasks are explained.

Overall, the design is kept in plain gray colours to prevent a large number of stimuli. The exception is a bright blue to give the UI an appealing and pleasant look suiting the marine topic, as well as to provide a consistent highlighting colour. For important elements, tooltips are provided as a help in case of doubt about the functionality in accordance with Nielsen's tenth heursitic. They also include the respective keyboard shortcut if any is available for this functionality. The shortcuts are listed in Table 4.1 and enable frequent, experienced users to accelerate their workflow on the home page. When closing MarOMarker, the current program state and settings are saved and loaded on restart.

4.7.1. Task: Set user ID and camera configuration

The first-time user is greeted with a welcome screen requesting the input of a user ID and a camera configuration. Both fields are equipped with validators to prevent illegal input. Thus, the first two tasks mentioned in the HTA of the software in Section 4.5 are covered and only need repetition in the case that the options need correction. This is executable on the settings page. With the welcome screen, error states are avoided occuring when one of the settings is missing.

4.7.2. Task: Load data and Set Output File

After the welcome window or when reopening the software, the user is directed to the data page (Figure 4.6). This adaptation of the prototype, which navigates the user to the home page first, corresponds more to the workflow: Data must be selected first or the data selection from the previous session must be checked. Only then the actual annotation process is to be carried out.

The data is selected using a calendar. Multiple options on the screen especially confuse inexperienced users as the prototype test revealed. To encourage the user to first click on a date, other options for data selection are grayed out, i.e. are not usable as highlighted by Figure 4.6 by the red box on the right. The process guide on the left additionally provides information about the steps to be taken and their sequence. In the case of data loading, it asks the user to set a date and output directory. As soon as a date is selected, the other options become usable. Since the images at AWI are organized by month and date, a predefined image path searches the respective directory for images. Their count is displayed to give feedback about whether the program found images and how many. This also allows the user to draw conclusions about incomplete datasets, e.g. if the number is odd, the number of left and right images is probably different.

4.7.3. Task: Run Neural Network

When the data is loaded, it can be used as input for the neural network by the user clicking the button "Run neural network". If a neural network model has not yet been loaded into the software, an error message appears asking the user for a model specification on the settings page. Similar to the camera configuration file, the model can be searched via the file explorer. The model is also reloaded on program restart. As an informative feedback on the system status during the computation of the predictions, the program indicates the number of images that have already been processed. The found organisms then appear in the data table which is available to the user in a separate window for review purposes.

4.7.4. Task: Edit Images

The core of the annotation process is the labelling and correction of organisms on the home page which is therefore key of the software development. In the following, the general


4. Design, Implementation and Evaluation of the User Interface

Figure 4.6.: MarOMarker initial data page. Two functionality groups are highlighted by thick red boxes: process guide (left) and initially disabled options (right).

structure of the page is elucidated. Then, manipulation options are covered before moving on to the icon choice and the interaction possibilities on the displayed image.

The general impression of the home page is the one of a photo viewer being beneficial in the present case because the focus is the image which is displayed as large as possible to facilitate the annotation. The familiar structure to most users who have viewed images on a computer before allows for a fast orientation: Navigation between images is possible by arrows on the left and right, while image manipulation options are located on top.

The image manipulation possibilities are further subdivided, spatially separated and logically ordered. Two functionality groups result from this (circled by thick red lines in Figure 4.7): one concerning general image manipulations (left box in the figure) and one containing functionalities for working within the image (right box in the figure). Due to the spatial arrangement and corresponding to the Gestalt law of proximity, the functionalities of the two groups are perceived as distinct from the other group and as connected within the respective group. Since the typical users read from left to right, the general options relating to the complete image are placed on the left, i.e. switching options between left, right, and the display of both images, filters, as well as image remarks. The sequence again reflects the scheme "from general to specific". The second group of functionalities is placed at the center of the top bar giving a pleasant sense of symmetry, as well as being in focus and quickly accessible for the annotation task. This symmetry is promoted by arrow keys for iterating



4. Design, Implementation and Evaluation of the User Interface

Figure 4.7.: MarOMarker home page. Two functionality groups are highlighted by thick red boxes: general image manipulation options (left) and options to work within the image (right). Different animal groups have different colours.

through all organisms within the current image. On the left side of the symmetry axis, the program provides additive features, i.e. zoom and addition of animals, while it presents reductive ones, i.e. removal of animals and reversal of actions, on the right side. Adding and removal options are thus placed apart which prevents imprecise clicks resulting in the exact opposite action from the intention.

The actions triggered by the buttons are indicated by icons and tooltips only, i.e. no text buttons are added. One reason is a more simplistic interface and another one that image representations are perceived faster than text. The icons are selected carefully taking the users' experience into account. For example, the filter button resembles e.g. a coffee filter and is hence to interpret on a metaphoric level. Similarly, the magnifying glass is a classical zoom symbol. The link between a plus sign and adding is obvious in a mathematical sense and removing items is commonly associated with a bin as seen on Mac, Windows and Linux operating systems. Lastly, the undo arrow is a known symbol for reversing actions.

The interaction of the user with the elements on the image is done intuitively by clicking and dragging: With active add mode, the first click adds a head at the click position, the second a tail. With active remove mode, one click on a drawn animal removes it. The principles of adding and removing in modes that need activation and deactivation serves the use case of the addition/removal of multiple organisms at once. With both modes deactivated, head and tail positions can be adjusted by dragging. These principles are easily revertible within themselves, e.g. an accidentally drawn animal can be removed by applying the remove mode. Therefore, the undo button is not implemented yet. It would give further action reversal options to support the feeling of safety.

4.7.5. Task: Match Animals on Left and Right Images

The task of manually matching animals from the left to those on the right image is not realized within the scope of this thesis. However, a framework is delivered supporting the concept of semi-automation for this task: On the data page, a button is added to run an automatic matching algorithm. Similar to the detection, another button is added for checking the results. Afterwards the user can trigger the computer to measure the length of the organisms by a last button. Thus, the complete annotation process is semi-automated and visible to the user. Jasper implemented such an algorithm that rectifies the images and performs block matching to find corresponding animals on the right images [6]. His implementation was transferred to this software and connected to the respective button. Equally, his length measurement calculations are attached to the length measurement button.

4.8. Usability Evaluation

After the implementation of the software, a final usability test is conducted. The aim is on the one hand to evaluate the design decisions followed on the prototype and on the other hand to measure how well the program realization meets the initial goal of a usable software. For this, an adapted form of the SUS questionnaire is used after the usability test. The test is conducted and evaluated with five persons.

4.8.1. Usability Questionnaire

For the measurement of usability, the SUS [43] is applied because it is a widely established and expedient standardized questionnaire for this task. [44] The classical SUS is adapted to suit the compiled test setting. Firstly, a German translation is used [45] because it is the native language of all probands of this test. Secondly, two questions are added to the questionnaire: One is addressed to participants having experience with the existing workflow and asks if and why they think a quality enhancement can be attained by MarOMarker. This question is important to verify that the current workflow is improved by the new concept. The other field enables for further notes, like improvement ideas and aspects that struck the test person. Such a free-text field reveals valuable opinion information. The resulting questionnaire can be found in Appendix D and is provided to the participants via Google Forms [46].

4.8.2. Test Structure

The usability tests are conducted individually and remotely, like the prototype test. Again, the proband is granted access to the computer of the moderator where the software is installed.

The moderator begins with a short introduction and the presentation of the four tasks:

- 1. Load data of 28.04.2016 and run the neural network on it (to generate predictions).
- 2. Correct the predictions.
- 3. Match animals from left and right images.
- 4. Measure the length of the animals.

While working on the tasks, the test person is asked to think aloud, i.e. talk about visual impressions, confusions, intentions, etc. As soon as one task is completed, the moderator points to the next one to ensure a structured procedure. This is followed by a short discussion to communicate aspects that stood out during the test, to clarify questions and to gather possible ideas from the users. Finally, the test person is asked to complete the prepared questionnaire after the call.

4.8.3. Test Persons

As stated e.g. by Sauro, the generation of a valid and stable score for the SUS is already possible with five test persons [44]. Hence, five is the number of participants selected for the final usability evaluation of MarOMarker. This includes probands B and C, that already participated in the prototype test, and three others portrayed in Figure 4.8. The test persons thus cover the typical age range, different experience levels on the current annotation process, on marine organisms and the developed design.



Figure 4.8.: Additional probands for the usability test

4.8.4. Results

The results from the usability test and questionnaire are presented in this section. It starts with the achieved SUS score, continues with the two questions added to the questionnaire and finalizes with key points from the discussions.

A system is ranked as serviceable usually from a SUS score larger than 68. [45] In industry, 80 is a common benchmark to indicate above-average usability. [47] The implemented software achieves a SUS score of 89.58 out of 100 and is hence rated very well. According to Lewis' summary of findings around the SUS, the score achieved by MarOMarker is equivalent to an excellent performance on a scale of seven adjectives, the second best after best imaginable [47].

Moreover, the three participants with experience on the current workflow agreed on question eleven: They are all of the opinion that MarOMarker improves the annotation process, mainly driven by:

- 1. The combination of all steps in a single program is perceived as more practical than the usage of multiple subsequent programs, also reducing error susceptibility.
- 2. The labelling itself is mentioned. The visual representation of heads and tails is perceived as a great help which is not available in Macro Scheduler.
- 3. The integrated zoom functionalities, the image navigation options, as well as position and group correction possibilities are appreciated.

In the free-text section and during the discussions the positive impression is emphasized. The test persons value the support of the neural network and describe the program structure as comprehensible. They found that especially the process guide on the data page added after the prototype test enhances intelligibility of the workflow. Also, the selection of images via the calendar and of the output directory is found to be more intuitive and clear than in the current programs. The confusions about this aspect and the difficulties in navigation during the prototype test are diminished by the guide and the initial disabling of all options but the calendar on the data page. Additionally, the useres labelled the shortcuts to be useful for a frequent use and two of them described the complete process to be fun.

Nevertheless, the discussions revealed improvement opportunities focusing on two points: the neural network loading and the invisibility of the matching. The former was a problem in all tests: The probands first click the "Run neural network" button before realizing that a network needs to be loaded. The intention to realize this action on the settings and not on the data page is that the network only needs to be loaded once, also for multiple sessions, and hence does not need to be visible all the time. This will become apparent for more frequent users. But since there are many first-time users, it is worth considering to move the loading functionality to the data page. The second point revolves around the matching. After the automatic matching, there is no option but the table to see which animals belong to each other on the left and right image. Despite the manual matching correction not being realized yet, such a functionality could be useful for the time until the software provides a manual matching, but also when the automatic matching hardly needs improvement anymore. A solution could e.g. be the already previously mentioned IDs for the animals. Since the manual matching process is not realized in the software, it is not evaluated further.

Overall, the response to MarOMarker is positive, in terms of the perceived usability score, but also in written words despite minor improvement possibilities. The software consequently meets the initial goal to ameliorate usability of the current workflow.

5. Conclusion

Within the scope of this thesis, a neural network and the software MarOMarker are developed which present together a semi-automatic solution for the annotation of marine organisms on stereoscopic images. Subsequently, a summary of the findings from both parts is given.

The first part of this thesis examines the usage of a neural network with heatmap input for the task of localizing and classifying heads and tails of organisms. The architecture follows the encoder-decoder pattern to ouput heatmaps at half of the input resolution, instead of 1/32. For the encoder, MobileNetV2 pre-trained on ImageNet is applied as backbone. Its layers are trained after an initial delay along with the custom encoder layers which is found to be beneficial for the present task. The custom layers profit from the already acquired knowledge and afterwards, features learned in the base layers are retrained to compensate for the low-contrast images of the current dataset. Additionally, class weights are introduced to compensate for the imbalances between the organism classes. The weights cause the network to correctly find some animals of the underrepresented classes, i.e. chaetognatha, jellyfish and unidentified, but because of the low availability of training data, it cannot learn a satisfactory generalization resulting in more FPs for these classes. This indicates a low diversity of the available data. Furthermore, the implemented networks benefit from a longer training, i.e. 110 training epochs instead of 60. For the matching of heads and tails, the closest match strategy achieved overall good results and only fails for large animal accumulations and close organisms. The other strategies, namely provisional segmentation and vector fields, are not usable yet. For the segmentation, the main problem is that it introduces confusions between head, tail and body. The vector fields generally demonstrate difficulties in pointing towards the respective head/tail, though the focused version attained better results than the one with standard MSE loss. The inferred data pipeline works best for the majority classes, while struggling with the class imbalances with similar outcomes as the predecessor work by Jasper. In contrast to Jasper's work, however, the FP count could be reduced significantly.

The second part of the thesis involves the development of a software according to UCD methods. It has two main goals: replace the current organism annotation workflow at the AWI and enable the correction of the neural network output. Since the system relies on user interaction, the software aims for being a usable, quickly learnable and efficient alternative to the current programs. The design inferred from the UCD development process is implemented and examined with respect to theses aims. In terms of usability, the realized software MarOMarker achieves a SUS score of 89.58 indicates an excellent performance supported by the overall positive user response. Many features are applied intuitively, especially during the correction of the network predictions implying a low amount of learning

5. Conclusion

required. Some functionalities needed more learning but can be mastered by few attempts, like the add and remove modes and the data input. The users stated that they are confident to being able to reuse the program easily indicating that no further or more complicated knowledge acquisition is required. The add and remove modes might need some learning time, but increase efficiency since many animals can be created/deleted at once. Moreover, the semi-automation and the shortcuts have the potential to accelerate the annotation process. In comparison to the current workflow, various aspects are improved by MarOMarker as confirmed by the users. The new software combines all necessary functionalities, like zooming, annotation, measurement and automation options. The steps carried out in separate programs at the moment are thus reduced to one program, decreasing the probability of errors and facilitating overview. Compared to Macro Scheduler alone, the combination of organism annotation and measurement slightly increases the workload since two clicks per animal are necessary instead of one, as well as two clicks to choose the group as compared to one. This is countered by undo options and visual feedback from MarOMarker which is missing in Macro Scheduler, where it can lead to frustration, longer editing times and errors.

Overall, the neural network is improved in comparison to the predecessor work by Jasper, but still requires improvement especially for underrepresented classes. This confirms the need for a software enabling a semi-automatic workflow, like MarOMarker. Thus, the temporal gap between the present and a point in time when the algorithm is precise enough for a full automation can be bridged. Additionally, the output of the software can be used for retraining, supporting the progress of the machine learning solution.

6. Outlook

The neural network, as well as the UI exhibit improvement and extension potential. As a guide for future work, some of them are listed in the following beginning with topics for the neural network and continuing with those for the UI.

In order to approach the class imbalance problem, the collection of more training images is suggested, especially for jellyfish and chaetongatha, since the low generalization ability of the weighted model indicates an insufficient data diversity. Subsequently, other techniques to approach this problem can be realized. The survey [48] summarizes a range of options for this, including data sampling and various custom losses. Moreover, the overall detection capabilities require improvement, for which multiple strategies are thinkable. For example, the implementation of a network that considers heads only or tails only could be beneficial. As discovered in this work, the neural network first learns to detect fish ends before it differentiates between heads and tails. Between these steps, it is observable on example images that some recognized heads/tails are lost. Removing the need to learn the differentiation between head and tail therefore might ameliorate performance. Another possibility is the modification of the model such that it takes an image pair as input, i.e. the left and the right image of the stereoscopic camera. Due to the relations in position and appearance of the animals on the image pair, the network might have a higher chance to verify detections, i.e. to further decrease FPs. Also for the connection encoding, some adaptions are thinkable to improve them and evaluate whether they can outperform the closest match algorithm and overcome its weaknesses. The segmentation network needs to learn the difference between the body parts, for which more channels could be expedient or a phased training, i.e. to first train on heads and tails and only afterwards on the body. The focused vector field strategy might benefit from adding more than one relevant arrow per head and tail: For all pixels covered by the groundtruth head heatmap, one vector could be be added to the vector field pointing towards the respective tail position. Similarly, the tail-head vector field can be adapted. Finally, the head-tail matching task can also be addressed by different approaches, like Regions with CNN features (R-CNNs) [49]. The two-stage detector first generates region proposals, i.e. regions that might contain an object (here an organism). Then it applies a CNN for feature extraction and a classification. In the second stage, the heatmap encoding implemented in this work could be utilized. Since this algorithm adds only one organism to a region, the area will only contain one head and one tail which belong to each other.

For the implemented UI, some features can be optimized, while there is large extension potential. To improve efficiency, the selection of species and group of an animal can be accelerated, e.g. by enabling the functionality of choosing the species/group of the lastly

edited animal. Threading could also be beneficial for the program since time-consuming tasks like loading the neural network, applying it and running the animal matching step, currently block the UI. With threading, the user could run the network and simultaneously work on the already processed images. To indicate the status of these processes, progress bars would be helpful. Additionally, the species list feature can be improved by providing the option to load different species lists that are sorted according to region. For the data loading part, a warning sign could be useful on the total number of images the program found for the given date if there are no images or if the number is odd, suggesting an unequal amount of left and right images. Besides, the software provides interfaces for extensions: The graphical representations of an undo button, a filter button, as well as the option for filtering data is already implemented, only lacking the functionality. In addition, the camera configuration file is already loaded though the connection to the calculations (which currently uses a different file that provides all necessary matrices) is absent. For the zoom feature, connecting the zoom level of the mouse wheel and the zoom slider would increase consistency. Lastly, completely new functionalities can be included, e.g. a handbook where all information about different species, their appearance and occurrence can be collected to centralize knowledge and to present a help especially for beginners. The feature could be sharable such that all users work on the same document and not only locally. For this, user accounts might be of advantage. Accounts could also facilitate data management, e.g. the centralized storage of result files of every person in a common file structure.

A. Interview Protocols

A.1. Expert Interview with Prof. Dr. Philipp Fischer, Head of the AWI Center for Scientific Diving (17.03.2020)

I: Was ist der aktuelle Arbeitsablauf?

PF: Dann ist es so, dass sich ein Student oder wer auch immer hinsetzt und morgens, jeden Morgen so eine Viertelstunde mit einem ersten Programm die Bilder durchschaut und schaut, ob Organismen drauf sind. Diese Organismen werden in acht Gruppen unterteilt: Fische, Krebse, Quallen, ... das sind acht Organismengruppen sozusagen und es wird auf jedem Bild die xy-Koordinate auf dem linken Bild des Organismus angeklickt, sodass wir auf dem linken Bild vom stereoskopischen Paar alle Organismen auf Interest mit einer xy-Koordinate draufhaben. Und da gibt es eine Tabelle dazu. So. Wenn wir dann etwas mehr Zeit haben oder wenn dann Leute kommen, nehmen die genau diese Bilder, das sind dann immer Monats- oder Tagesdaten, 48 Bildpaare, gehen in ein spezielles Programm rein, das ist ein stereoskopisches Programm, was wir entwickelt oder was wir geschrieben haben, nehmen diese schonmal vor getaggten Bilder und gehen jedes einzelne markierte Objekt durch, suchen das markierte Objekt, oder es wird angezeigt, auf dem linken Bild, die Schnauze wird angezeigt, bei einem Fisch, und parallel wir dazu das zweite Bild, das rechte Bild, angezeigt. Die Aufgabe desjenigen ist, den Fisch auf dem linken Bild Schnauze und Schwanz anzuklicken, von demselben Fisch Schnauze und Schwanz anzuklicken und dann zu beurteilen, welche Art das ist, es einzutragen und dann zum nächsten Fisch zu gehen oder zum nächsten Objekt zu gehen. Wenn er mit dem Bild fertig ist, das kann schnell gehen, wenn da nur zwei Fische drauf sind, ist das dann eben schnell, wenn es 30 Fische sind, dauert es lang, dann ist quasi unsere Datenanalyse von diesem Bild fertig. Dann wissen wir, im xyz-Raum, welcher Organismus welche Art, welches Tier, ist wo vorhanden, wir haben die Koordinaten, wir haben die Größe der Tiere und wir können mit diesen Daten weiterarbeiten.

I: Wer arbeitet momentan an dem Thema?

PF: Wir arbeiten immer daran. Da laufen seit Jahren immer wieder Arbeiten zur Arbeit an den Daten selbst dran. Momentan ist eine Arbeit, weil wir haben ja neben dieser stereoskopischen Kamera natürlich die ganzen hydrographischen Parameter, wie Wassertemperatur, Lichteinstrahlung, ... und mit diesen Daten wir im Prinzip dahingehend gearbeitet, dass wir korrelieren ok welche spezifischen Arten sind bei welchen Wassertemperaturen da, wie verändert sich das Artenspektrum seit 2012, wann sind denn die Monate of Interest – wir haben die letzten Jahre festgestellt, dass eben nicht die Sommermonate, sondern die Wintermonate in der absoluten Dunkelheit die biologisch interessantesten Parameter sind. Und wir schauen, welche Arten kommen in welchen Wassertiefen vor, denn das System fährt in der Wassertiefe auch jeden Tag auf eine andere Tiefe an. Das sind reine biologische Fragen, mit denen wir eigentlich permanent arbeiten.

I: Wie funktioniert das Abfahren der verschiedenen Wassertiefen?

PF: Mit den Wassertiefen ist das so. Wir haben auf Spitzbergen unter Wasser auf etwa 12m Wassertiefe ein Fundament, eine Basisstation, wo Pumpen, wo Sensoren mit allem drum und dran dran sind, wo wir year around messen. Also die ganzen hydrographischen Daten werden dort gemessen. Das ist Kabel-gebunden, das geht mit einem Unterwasser-Kabel dahin und sitzt dann eben. So und auf diesem System haben wir eine vertikal-profilierende Winde sitzen. Diese Winde trägt die stereoskopische Kamera, trägt aber auch nochmal verschiedene andere Parameter und Sensoren. Und die können wir von hieraus steuern, im Vertikalprofil fahren. Jeden Tag um 12:45 Uhr fährt in Spitzbergen oben, also 80 Grad sind das, die Anlage bis zu Wasseroberfläche hoch, nimmt während der Zeit Proben und fährt auch wieder runter. Und dann ist es so, dass wir die Wassersäule über 12m über diesem Fundament in 5 Schichten eingeteilt haben, von 9m, 7m, 5m, 3m und 1m Wassertiefe. Wenn die Kamera um 12:45 Uhr oder um 13 Uhr ihr Profil fertig gefahren hat, fährt sie auf eine von diesen fünf Wassertiefen. Die sind zufallsverteilt, das wird jeden Tag neu ausgelost und bleibt dort für 24 h stehen. Wenn wir eine ganze Woche mit fünf Tagen vorbei haben, ist jede von diesen fünf Stufen, einmal 24h abgescannt und am Wochenende, wenn die Kamera frei hat, wenn also alles gelaufen ist, fährt die auf den Grund und macht auf 9m oder auf 10m. Montag geht es wieder los, sucht sich eine Tiefe aus, macht erst ein komplettes Vertikalprofil, fährt dann auf eine bestimmte Tiefe, bleibt dort 24h, macht ihre 48 Bildpaare und so geht es weiter. Es hat sich gezeigt, dass durch die kontinuierliche Aufnahme über 48 Bildpaare pro Tag wir einen extrem guten Überblick bekommen, was da los ist.

I: Was ist die Problemstellung?

PF: Die Problemstellung ist, dass die vollständige manuell Auswertung der 48 stereoskopischen Bildpaare pro Tag zu aufwändig ist, um das Verfahren durchgängig anzuwenden. Es können daher immer nur kleine ausgewählte Zeitfenster analysiert werden so dass wichtige Ereignisse evtl. verpasst werden.

I: Was ist der Kontext

PF: Wir möchten eine Monitoring Methode für die Biodiversität höherer Organismen (Fische etc.) in marinen Ökosystemen entwickeln.

I: Auf welcher Plattform soll die Anwendung laufen? (Mobil, Web, Desktop, VR, ...) **PF**: Wir lassen die Anwendungen am PC auf VM's laufen.

I: Wer macht die Auswertung der Bildpaare?

PF: Das ist sehr unterschiedlich. Die Routine-Auswertung von den Bildern machen wir morgens und teilen die uns meistens auf. Bei den Teilaufnahmen von den stereoskopischen

Aufnahmen, das machen wir jeweils, wenn Diplomanten da sind oder wenn Bachelor-Studenten da sind oder wenn ich selbst an einem bestimmten Monat Interesse habe. Ich habe zum Beispiel die Phase 2013/2014 fast komplett selbst ausgewertet. Das kommt immer sehr darauf an, wer grade an diesen Arbeiten sitzt. Das ist sehr unterschiedlich. Es ist so, und zwar die erste Auswertung, dass man einfach nach Tieren schaut und die in Gruppen einteilt, das können irgendwelche Studenten oder Praktikanten machen, weil es bedarf keiner riesigen Expertise da einen Fisch zu erkennen. Die zweite Stufe wird bei uns in der Regel von Leuten gemacht, die ein bisschen Ahnung haben davon, weil die müssen entscheiden, was für eine Art ist das und das bedarf schon etwas mehr Expertenwissen.

I: Wer könnten weitere Nutzer der Anwendung sein?

PF: Der Nutzerkreis wäre riesig, wenn die Methode funktionieren würde. Viele aquatischen Ökologen würden diese Methode zum Monitoring verwenden.

I: Was wären vorstellbare Features oder Erweiterungen?

PF: Mehrstufig. Semiautomatische -> Automatische Erkennung von Objekten in Klassengruppen. Das Programm lernt aus den Eingaben des Menschen Automatische Vermessung der Objekte in länge x Höhe. Automatische Erkennung der unterschiedlichen Arten. Aufbau einer virtuellen Welt in der man sich bewegen kann.

I: Welche Rolle spielt die Bedienbarkeit?

PF: Solange der Mensch bei der Vermessung interagiert eine große Rolle.

I: Was muss am Ende der Masterarbeit grundlegend gewährleistet sein?

PF: Ein erster Ansatz, dass Schritt 1 (siehe oben) mit möglichst wenige Fehlern möglich ist. D.h. dass z.B. 60-70% der Organismen richtig erkannt und in Gruppen eingeteilt werden und diese auch richtig vermessen werden. Die restlichen 30% sollen mit Wahrscheinlichkeiten für die Korrektheit der Gruppierung angeben werden und der Bediener muss am Bildschirm korrigieren können. Aus dieser Eingabe muss da Programm lernen, um die Trefferquote zu verbessern.

A.2. Interview Guide

- 1. Seit wann arbeitest du schon beim AWI?
- 2. Was ist deine Rolle/Position beim AWI?
- 3. Welche Aufgaben hast du?
- 4. Wie sieht ein typischer Arbeitstag aus?
- 5. Wieso markierst du die Organismen auf Fotos?
- 6. Fragen zu Macro Scheduler und/oder Stereo Marker
 - 6.1 Wann hast du das Programm das letzte Mal benutzt?
 - 6.2 Wie oft benutzt du Macro Scheduler/Stereo Marker?
 - 6.3 Wie viel Zeit verwendest du bei einer Nutzung dafür?

6.4 Wie ist der Arbeitsablauf bei der Markierung/Vermessung der Organismen bei diesem Programm?

6.5 Welche zwei oder drei Dinge findest du an dem Programm gut?

6.6 Kannst du mir erklären, was dich bei dem aktuellen Prozess frustriert?

7. Wann planst du, Macro Scheduler/Stereo Marker nochmal zu benutzen? (Zielt auf auf einen weiteren Termin ab, an dem ich beobachten kann, wie der Nutzer das Programm tatsächlich nutzt.)

A.3. Interview with User C (30.06.2020)

I: Seit wann bist du denn schon am AWI?

C: Ich bin jetzt seit März hier. Ich mache Studenten-Nebenjob quasi. Seit März bin ich fast jeden Tag hier.

I: Okay und was machst du so? Was sind deine Aufgaben?

C: Ich mache viel Tauchunterstzützung eben aber auch in der Werkstatt einiges. Also wir müssen für unsere Sensoren Unterwasser viel Wartung teilweise machen und auch die Verkabelung und so. Da kümmere ich mich gerade viel drum. Dass einfach die ganzen Sensoren laufen. Genau und dass wir die Unterwasser anschließen können. Und nebenbei mache ich dann noch das mit der Bildauswertung, wenn ich Zeit habe.

I: Wird denn jeden Tag daran gearbeitet oder wie oft wird das gemacht?

C: Ja, also das habe ich eine Zeit lang gemacht, dass ich jeden Morgen daran gearbeitet habe. Aber wenn wir halt auch Tauchen gehen, sind wir halt teilweise den ganzen Tag eben weg und dann habe ich noch andere Projekte die fertig gemacht werden müssen. Deswegen komme ich da einfach nicht zu. Und Person F genauso. Also sie hat halt auch super viel zu tun.

I: Das habe ich gemerkt, heute hat sie es ja auch nicht geschafft. Aber schön, dass es dann mit uns heute geklappt hat. Okay, und wie sieht dann so ein typischer Tag bei dir aus?

C: Ich komme zur Arbeit und meistens gehe ich quasi direkt in die Werkstatt. Also im Moment bauen wir für ein Projekt eine Connector Box. Das ist quasi die Stromversorgung für die einzelnen Sensoren, also wir haben da sechs CTDs angeschlossen.

I: Was sind CTDs?

C: Eine CTD ist, die misst Temperatur, Leitfähigkeit und den Druck Unterwasser eben. Das sind so automatische Geräte. Damit kannst du eben so Langzeitdaten aufnehmen. Und da gibt es gerade ein neues Projekt und die müssen halt mit Strom versorgt werden, brauchen aber alle irgendwie unterschiedliche Voltzahlen und so.

I: Spannend. Und was studierst du?

C: Ich habe marine Umweltwissenschaften studiert in Oldenburg mit Schwerpunkt Biologie, aber auch marine Sensorik und so und genau, hab den Forschungstaucher in der Ausbildung, im Studium gemacht und war letztes Jahr auch mit auf Spitzbergen zum Tauchen und bin da ans Tauchzentrum gekommen.

I: Schön, also genau da die Bilder die ich jetzt auswerten darf. Du hast ja schon erzählt, dass du im Moment nicht so oft diese Programme benutzt zum Markieren. Wie oft machst du es

denn im Moment so?

C: Oh Gott, ich glaube ich habe es drei Wochen lang gar nicht gemacht, tatsächlich. Davor habe ich es ungefähr jeden Tag gemacht, so eine Stunde oder so. Das dauert dann teilweise doch länger. Also Person E meinte am Anfang ja ja man braucht irgendwie so eine Viertelstunde, halbe Stunde, um einen Tag zu machen. Das ist glaube ich grob unterschätzt. Du hast glaube ich 48 Bilder pro Tag, die machen ja alle halbe Stunde diese Bilder und du musst halt teilweise echt die genau im Raster irgendwie durchgehen. Ich bin auch mit Bildschirmlupe halt teilweise echt so lang gegangen. Und das dauert halt teilweise echt lange.

I: Aber du guckst dann immer nur auf dem linken Bild?

C: Genau, du guckst immer nur ein Bild an. Das ist dann automatisch das Linke.

I: Also ist es doch schon ein aufwendiger Prozess.

C: Ich habe sowas halt auch noch nie vorher gemacht. Ich kenne halt so ein paar Arten und so durch mein Studium, aber so Bildauswertung habe ich so noch nicht gemacht. Da muss man sich auch etwas reinfuchsen. Da ist mir auch aufgefallen, dass so kleine Crustaceen, also die sind wirklich so klein auf dem Bild auf irgendeinem Baumstamm zu erkennen. Weil die dann teilweise in Gruppen da sitzen. Und das siehst du manchmal nur, wenn du das vorherige Bild hast, wo sie nicht waren und dann siehst du nur den Unterschied zum vorherigen Bild, dass dann da auf einmal diese kleinen Flecken sind.

I: Ah okay, ich hatte auch so meine Schwierigkeiten als ich mir die Bilder angeguckt habe, da irgendetwas zu erkennen, außer so große Fische, die da lang schwammen. Und du hast jetzt hauptsächlich dieses Macro Scheduler benutzt?

C: Genau.

I: Hast du schonmal das andere verwendet?

C: Leider noch nicht, da sind wir noch nicht zu gekommen.

I: Okay, also bisher Macro Scheduler. Und wie ist da so der Arbeitsablauf? Also wenn du dann mit dem Programm Tiere markieren musst?

C: Ein bisschen schwierig. Also du kriegst am Anfang quasi zeigt er dir das Bild ja und dann musst du dem erstmal die Spanne von den Bildern geben, damit der einordnen kann, wie groß das Bild ist. Da muss man die beiden Ecken markieren, links oben und rechts unten. Und dann kannst du noch eine Korrektur eingeben, also dass der die Helligkeit erhöht oder eben die Farbe besser herausfiltert, dass du eben dann einen Farbkontrast erkennen kannst. I: Ja damit du dann die Organismen besser siehst auf dem Bild.

C: Genau, teilweise ist es halt super grün, weil halt einfach durch das Wasser rot und alles rausgefiltert wird. Genau und da gibts es dann so Korrekturen. Das ist ganz gut, das kann man für jeden Bild dann quasi extra einstellen. Wenn es dann neu kommt, kannst du dann nochmal einstellen, okay, ich hätte jetzt gerne diese Korrektur. Er nimmt aber normalerweise am Anfang das, was du beim Vorherigen gemacht hast, aber du kannst es dann nochmal manuell verändern.

I: Bringt denn das was? Hilft es, dass es voreingestellt bleibt?

C: Ja das ist schon gut, aber es ist auch wichtig, dass man es dann noch verändern kann, weil manchmal bist du ja an einer anderen Stelle. Also teilweise sind sind die Bilder ja vom Grund, aber dann hast du auch in der Wassersäule und da brauchst du manchmal einfach eine andere

Korrektur. Also ich finde die Voreinstellung auf jeden Fall gut, weil dann musst du dann nicht jedes Mal draufklicken, also es nicht jedes mal manuell einstellen. Und manchmal, wenn es nicht passt, ändert man es dann nochmal.

I: Okay und dann kannst du dann einfach auf dem Bild herum klicken und die Tiere markieren?

C: Genau, also so einfach ist das nicht. Du kannst dann quasi, also wenn das eingestellt ist, dann musst du irgendwie drei oder vier Sekunden warten, um einmal das Bild zu scannen quasi und dann kannst du anfangen zu klicken. Das Problem, das ich dabei gefunden habe ist, dass manchmal dieser Macro Scheduler, also du hast an der Seite so eine Leiste, wo der dir dann anzeigt, welchen Organismus du klicken kannst. Das sollte der dir eigentlich anzeigen, wenn du einmal aufs Bild klickst und dann zeigt der das rechts an. Manchmal macht der das aber nicht, wenn du zwischendurch z.B. die Bildschirmlupe benutzt hast oder so, erscheint das nicht. Dann klicke ich dann irgendwie fünf Mal und weiß dann nicht, okay, hat der mir jetzt diese Stelle markiert? Weil du siehst halt keine Markierung. Also das ist halt richtig nervig, dass du in dem Bild nicht siehst, was du schon markiert hast. D.h. wenn dann total viel da drin ist, vergisst du dann schon, okay habe ich denn jetzt schon, habe ich den nicht. Deswegen markiere ich den dann manchmal glaube ich fünf mal. Also so schlimm nicht, aber genau, es wäre ganz gut, wenn man da direkt noch eine Markierung drin hätte im Bild, dass man dann weiß, okay das ordne ich dem zu.

I: So ein bisschen visuelles Feedback wär schon ganz hilfreich. Werden dann angezeigt, welche Organismen du schon markiert hast? In einer Tabelle oder so?

C: Genau, da steht rechts eben diese, du hast so einzelne Felder, wo halt draufsteht, dass es sich Benthische Crustace, pilatische Crustace und dann Fische und jellyfish und so was. Und daneben, wenn du das schon geklickt hast, dann steht daneben immer eine Zahl.

I: Ah okay, der zählt dir das dann einfach, okay.

C: Genau, also eine Zahl steht halt schon da, aber manchmal weißt du dann auch nicht, wie viele du angeklickt hast, bevor du die alle nochmal durchzählst.

I: Und wenn du dann alle Tiere gefunden hast?

C: Dann kannst du quasi all objects counted klicken und dann kommt das nächste Bild.

I: Und das Programm stoppt dann automatisch, wenn du mit einem Tag durch bist?

C: Genau, dann schließt es sich einfach. Zwischendrin stoppen ist irgendwie so ein bisschen schwierig, dann musst du den Macro Scheduler irgendwie immer so abbrechen. Das ist immer so ein bisschen nervig.

I: Ich hatte schonmal den Anfang so ein bisschen gesehen, da musst du dich ja durch durch viele Sachen durchklicken, oder? Also, du musst erstmal eingeben, okay hier ist meine ini-Datei und die Kamera-Konfigurationsdatei und ganz viele andere Sachen. Musst du die dann auch jedes Mal eingeben, wenn du das Programm startest?

C: Ja, genau und das ist auch so ein bisschen nervig, warte ich habe das irgendwo aufgeschrieben. Du musst halt jedes Mal dem wirklich das result file sagen und irgendwie kriegt der das auch nicht hin quasi den letzten Überordner zu nehmen. Das wäre ganz gut, weil du hast ja, du speicherst ja alle von einem Monat im gleichen Überordner, meine ich. Und das wär halt ganz schön, wenn das Programm halt einfach sagen würde, okay das ist der Ordner in dem ich das jetzt speichern will. Ist das okay? Und wenn er das jedes Mal vom letzten Mal nehmen würde, wäre es irgendwie sinnvoller.

I: Ja okay, dass man dann also seinen eigenen Ergebnis-Ordner dann hat und dann speichert der da die Sachen rein.

C: Genau, also habe ich auch. Ich habe auch dieses eine result file, das ist glaube ich einfach nur eine text file, was er halt speichert und da muss ich im Moment das jedes Mal neu eingeben, welches file will ich denn haben. Und da gibt es so viele Ordner und keine Ahnung was und dann dauert das auch immer ein bisschen, das zu finden.

I: Ich kann mir auch vorstellen, dass es etwas frustrierend ist, wenn man dann das Programm abbrechen möchte zwischendrin und dann da weiter machen will, wo man aufgehört hat.

C: Ja, ja. Also das geht, ich glaube da muss man das dann nicht nochmal neu eingeben, du musst das nur wenn du quasi einen neuen Tag anfängst nochmal neu machen. Aber ich bin mir grade nicht mehr sicher. Ich habe länger nicht mehr abgebrochen, weil ich da immer Angst habe, dass ich irgendwelche Bilder nicht speichere oder so.

I: Kann ich verstehen, okay. Welche Sachen findest du denn an dem Programm gut?

C: Also ich finde es gut, dass du die Auswahl hast zwischen unterschiedlichen Organismen, dass das dann einfach schon dasteht und du das machen kannst. Ich glaube ich fände es aber besser tatsächlich, wenn man irgendwie noch Bilder mit dabei hätte. Also dass man dann Beispielbilder hätte oder sowas.

I: Ich glaube, dass ist tatsächlich im zweiten Programm so, dass du da Beispielbilder kriegst. C: Ja genau, damit du es einfach genau bestimmen kannst. Aber vor allem für jemanden, der das so noch nicht gemacht hat, ist es am Anfang echt schwierig, wirklich zu wissen worauf muss ich achten. Und es wäre glaube ich ganz gut, wenn man das am Anfang direkt hätte. Ich habe auch schon angefangen, quasi Screenshots zu machen von den Bildern mit den Arten dann, damit ich weiß die gibts, weil auch wahrscheinlich Praktikanten kommen, die das machen müssen. Und dann dachte ich, man könnte denen dann die Bilder einmal zeigen und sagen guck mal, das kann es da geben und das ist der Organismus und das und das müsstest du markieren oder darauf muss man achten. Weil manche Sachen fallen einem am Anfang gar nicht auf.

I: Das ist natürlich ganz praktisch, wenn man so ein kleines Handbuch dann hat.

C: Genau, irgendwie so ein kleines Portfolio, wo du halt siehst, das könnte es geben oder diese kleinen Crustaceen sitzen eben auf Baumstämmen, meisten ins Gruppen, das sieht dann so aus.

I: Könntest du mir die Sachen vielleicht auch schicken?

C: Ja kann ich machen. Ich habe bisher noch nicht so viele, aber wenn ich weiter mache, dann schicke ich sie dir gerne rüber.

I: Das wäre super, weil es wäre natürlich auch eine Überlegung wert, dass man sowas dann auch in dieses Programm integriert, wo man mehr Informationen haben kann und einfach dann Infos über die Organismen auf den Bildern kriegt und wie die dann aussehen.

C: Ja, dass man weiß wie sie aussehen und wo man sie eventuell findet. Also na klar, ich weiß, wie ein Fisch aussieht oder ich weiß auch, wie eine Krabbe aussieht, aber dass sie jetzt immer in Grüppchen auf einem Baumstamm hocken.

I: Das wusste ich auch nicht.

C: Das hat Person E mir mal gesagt, als ich so meinte, du Person E, dass sind doch Crustaceen, aber muss ich die markieren oder wie erkenne ich die besser. Und da meinte er, ja ja die kommen so im März rum, da sind dann Grüppchen von denen. Das ist halt so eine Info, das wäre halt schon sinnvoll.

I: Auf jeden Fall. Hast du denn noch andere Sachen, die du an dem Programm gut findest? Wir haben ja jetzt schon wieder Verbesserungsmöglichkeiten gefunden.

C: Ich überlege grade, also an sich sind mir viele Verbesserungs-Sachen aufgefallen. Ich habe jetzt keine Liste gemacht, wo ich sage, das und das finde ich gut. Deswegen fällt mir grade nicht so viel dazu ein.

I: Okay, dann gehen wir auch gerne zu den Verbesserungsmöglichkeiten zurück bzw. den Sachen, die nicht so gut laufen. Ich meine, wir hatten da ja auch schon einige, die nicht so gut laufen, wie z.B. das ständige Eingeben von den neuen result files und auch später, dass keine Markierungen angezeigt werden oder dass die Sidebar manchmal nicht reagiert, wenn du auf Organismen geklickt hast. Hast du noch etwas anderes, was dir nicht gefällt?

C: Also was ich noch habe, du kannst nicht vor und zurück gehen. Also du kriegst ja automatisch, wenn du sagst, okay ich bin fertig, kriegst du automatisch das nächste Bild angezeigt. Und das letzte, das davor ist quasi weg, weil der das dann in einem anderen Ordner speichert für den zweiten Schritt eben. So funktioniert das. Und deswegen kannst du halt nicht, der geht halt immer automatisch durch die Bilder durch und dann kannst du halt nicht sagen, ich will zurück gehen. Ich hatte auch die Situation, dass ich auf einem Bild etwas gesehen habe, wo ich gedacht habe, war das jetzt auf dem vorherigen Bild auch schon da? Habe ich das einfach nicht gesehen? Und da würde ich ganz gerne zurück gehen können. Weil dann kannst du nochmal gucken, ok war das vorher jetzt schon da, muss ich das hier auch markieren. Und deswegen, ja ich weiß halt nicht, vor allem wenn du auf den Bildern nichts markierst, dann speichert der die auch nicht ein für den nächsten Schritt. Also glaube ich zumindest, das weiß ich nicht so genau.

I: Das würde ja Sinn machen, wenn du schon vorher keine Organismen markiert hast, dann braucht man den zweiten Schritt ja theoretisch nicht.

C: Ja klar, aber dann wäre es gut, wenn man zurückgehen könnte und sagen könnte, ah da habe ich jetzt etwas gesehen, es ist halt super schwer zu sehen und es könnte halt sein, dass es im vorherigen Bild auch schon da war und das wäre ja wichtig für unsere Daten.

I: Auf jeden Fall. Und wie sieht es aus, wenn du jetzt fälschlicherweise einen Organismus hinzugefügt hast, also wenn du jetzt aus Versehen auf eine Stelle geklickt hast, passiert das auch mal?

C: Ja das passiert auch mal, das ist auch mit der Sidebar so, ich ignoriere das dann meistens und markiere dann was, wenn dann noch etwas kommt. Aber das kannst du später in dem anderen Programm, also Person E hat uns das einmal gezeigt, wie der das ganze dann anzeigt. Dann zeigt der dir die ganze Markierung an und da vermisst du ja dann nochmal, sagst, was es ist und da könntest du dann auch die Punkte raus löschen, wenn du was falsch markiert hättest.

I: Aber das kannst du in dem ersten Programm noch nicht?

C: Ne. Also in dem ersten Programm gibt es nicht die Möglichkeit zu sagen, da ist jetzt doch nichts.

I: Das ist natürlich dann auch etwas doppelte Arbeit die dann anfällt. Dann sieht man auch gar nicht, wo man geklickt hat, was es jetzt war, macht dann weiter und dann im zweiten Schritt muss man das nochmal alles kontrollieren. Das ist ein bisschen redundant. Gibt es noch irgendwelche Sachen, die nicht so gut bei dem Programm laufen?

C: Es läuft an sich schon, es ist nur ein bisschen umständlich, sag ich mal. An sich finde ich es gut, man kommt dann immer zum nächsten Bild und kannst es dann automatisch klicken, aber ein paar Sachen würden die Arbeit so ein bisschen vereinfachen. Einerseits, wenn du sehen würdest, wo das ist, das hatten wir ja schon gesagt und was ich noch gut fände, wäre, wenn, weil häufig hast du den gleichen Bildausschnitt. Und es macht es dann viel einfacher zu sehen, ob da jetzt was neues ist oder nicht, wenn du das alte Bild daneben hast. Ich weiß nur nicht, ob man das umsetzen kann. Ich habe mir halt einfach einen Screenshot gemacht von dem Bild und das dann auf meinen zweiten Bildschirm rüber gezogen und bin dann immer durchgegangen. Dann siehst du es halt super schnell, dass da ein Unterschied ist. Das fällt einfach schneller auf, als wenn man jedes Bild einzeln durchgeht.

I: Das macht Sinn. Aber da würde es ja schon helfen, wenn du vor und zurück gehen könntest. Dann kannst du ja auch den Unterschied sehen.

C: Ja oder wenn du halt zwei Fenster offen haben kannst, dass du sagst hier habe ich das vorherige Bild, hier habe ich das nächste Bild und dann kannst du gucken, ok hier habe ich einen Unterschied.

I: Das wäre also auch schon mal hilfreich. Wenn dir noch etwas einfällt, was gut oder schlecht an dem Programm ist, kannst du mir das auch gerne immer schreiben zwischendurch, du hast ja meine E-Mail Adresse. Weißt du ungefähr, wann du das Programm nochmal benutzen willst?

C: Ich wollte diese Woche nochmal, aber ich weiß nicht, ob ich dazu komme und nächste Woche bin ich erstmal weg. Danach wollte ich mich wieder dran setzen.

I: Also falls du es benutzt, könntest du mir da vielleicht Bescheid sagen, sodass ich dir da einfach über die Schulter gucken kann, so per Screensharing? Damit ich den Ablauf einfach mal sehe, ich würde auch einfach nur silent listener sein, wirklich auch nichts machen, sondern nur zugucken.

C: Ja klar, können wir gerne machen.

I: Das wäre echt super. Dann könnte ich auch nochmal genau sagen erst klickt man hier, dann da, so ist der genaue Ablauf, damit man das auch schön formal analysieren kann. Das wäre noch sehr hilfreich für mich.

C: Es dauert halt nur relativ lange das zu machen, aber vielleicht reicht es dir ja schon die ersten paar Bilder mit anzugucken oder wie man anfängt und dann wiederholt es sich ja nur noch.

I: Ja schon, das würden wir dann einfach spontan schauen, würde ich sagen. Aber wie gesagt, wenn du halt Zeit hast, sag mir gerne Bescheid. Normalerweise bin ich zeitlich recht flexibel, weil ich an meiner Masterarbeit arbeite und es mir selber einteile und nicht so viel anderes Zeug mache. Aber ja, das wären dann auch alle meine Fragen gewesen. Da waren wir ja

doch recht schnell durch.

C: Ja ich halt leider mit dem zweiten Teil noch nicht gearbeitet.

I: Willst du denn mit dem zweiten Teil noch arbeiten?

C: Ja, ich habe mit März jetzt angefangen, ich bin mit dem Monat noch nicht ganz durch und dann wollte mit Person E noch eben zeigen, wie man das mit dem anderen Programm macht. I: Sobald du also einen Monat fertig hast, darfst du dann zu Schritt zwei übergehen.

C: Ja, ich muss nur mal gucken, ich habe das Gefühl, ich habe einige Organismen übersehen, aber da muss man schauen.

I: Guckt Person E da nicht auch nochmal drüber?

C: Ich glaube, er würde im zweiten Schritt auch nochmal drüber gucken, aber jetzt am Anfang bei der Auswertung hat er halt nicht großartig geguckt. Also er kommt immer mal vorbei und guckt mit auf den Bildschirm irgendwie so, aber ja. Ich habe gerade meine Notizen gefunden, ich hatte nämlich zwei Seiten mit Notizen, aber ich glaube, da habe ich alles. Automatisches result file, aber das hatten wir schon, genau, was man nicht machen kann, er schließt halt automatisch das komplette Programm wenn du mit einem Tag durch bist. Wenn man aber mal etwas mehr Zeit hat, will man aber direkt weiter machen und dann muss man nochmal alles von vorne machen.

I: Nochmal schön alles anklicken und eingeben.

C: Genau, also das wäre halt cool, wenn man so ein Programm dann offen lassen kann, sodass man mehrere Tage bearbeiten kann.

I: Okay, ja sehr gut. Da hast du mir auf jeden Fall schonmal sehr geholfen, danke. Und du machst dich jetzt zurück auf in die Werkstatt?

C: Ja wir haben gleich AG meeting in einer halben Stunde. Ich habe noch eine Platine gemacht gestern und die muss ich jetzt noch vorbereiten und ja.

I: Das hört sich gut an. Ja super, dann vielen Dank nochmal, dafür dass du dir die Zeit genommen hast, das hat mir Spaß gemacht, dich auch mal kennen zu lernen und wie gesagt, sag gerne Bescheid, wenn du das Programm nochmal benutzt. Ich bin dabei.

C: Ja ich überlege vielleicht entweder Donnerstag oder Freitag Vormittag. Ich muss jetzt mal gucken, wie ich jetzt voran komme, aber ich muss sowieso noch auf Teile warten, damit ich den Wassermelder bauen kann. Daher habe ich echt Zeit.

I: Das würde passen, wie gesagt, sag einfach Bescheid und dann gucken wir. Super, dann wünsche ich dir noch einen schönen Tag und bis zum nächsten Mal.

A.4. Interview with User D (29.07.2020)

I: Okay, seit wann arbeitest du denn schon am AWI?

D: Also richtig arbeiten seit erstem Juli, also noch nicht so lange. Allerdings habe ich hier schon meine Bachelorarbeit und meine Masterarbeit geschrieben. Ich war auch öfters als HiWi hier beschäftigt. Also ich war immer mal hier, sozusagen.

I: Und jetzt bist du aber festangestellt?

D: Ja, jetzt bin ich fest eingestellt. Ich habe halt meine Masterarbeit komplett mit dem

Programm gearbeitet und habe mehrere Monate die Bilder ausgewertet und ja, also habe halt damit gearbeitet. Ich soll jetzt halt nebenbei immer mal die Bildbearbeitung weitermachen, damit die Bilder halt einfach bearbeitet werden. Genau, und Person C eigentlich auch, mit der du schon Kontakt hattest. Es läuft so nebenher sag ich mal und man hat dann oft noch andere Projekte, die irgendwie jetzt ganz aktuell fertig werden müssen und dann ist es dann so, dass das manchmal ein bisschen vernachlässigt wird mit der Bildbearbeitung, weil das eben auch recht lange dauert.

I: Ja, das hatte mir Person C auch schon erzählt, dass sie das am Anfang noch jeden Tag gemacht hat und das letzte Mal, dass sie damit gearbeitet hat, war jetzt drei Wochen her, weil einfach einige Sachen dazwischen gekommen sind, die dann halt direkt erledigt werden müssen. Und du hast dann also schon mit beiden Programmen gearbeitet? Weil ist ja zum einen dieses Macro Scheduler und zum anderen ist das Stereo Marker.

D: Ja ich habe mit beiden Programmen gearbeitet. Ich habe jetzt gestern und vorgestern mich sozusagen nur in den ersten Teil eingearbeitet nochmal, also das was Person C auch gemacht hat. Ich habe halt grundsätzlich auch diesen zweiten Schritt, wo man die eigentliche Verbesserung macht, habe ich in meiner Masterarbeit auch gemacht, müsste ich mich jetzt auch nochmal einarbeiten, wie das nochmal genau war. Aber grundsätzlich habe ich das schon gemacht, das würde ich auch wieder hinkriegen.

I: Was hast du denn in deiner Masterarbeit gemacht?

D: Ich habe die Bilder von Spitzbergen halt ausgewertet über neun Monate, acht Monate, acht Monate. Ich habe also acht Monate Bilder ausgewertet und es ging quasi um die Fragestellung, wie sich die Fischgemeinschaft bzw. die allgemeine Community da oben verändert. Also es ist ja z.B. so, dass auf Spitzbergen ist es im Sommer ja immer hell und im Winter ja immer dunkel mit kurzen Übergangsphasen. Wir haben hier auf Helgoland ja auch so eine Kamera, die Bilder aufnimmt. Da sind halt auch teilweise Untersuchungen gemacht worden und da sieht man z.B. den Unterschied zwischen Tag und Nacht und auf Spitzbergen ist das halt ein bisschen anders. Da gibt es eher so saisonale Verschiebungen in der Artenzusammensetzung und da habe ich mich halt drauf konzentriert. Man hat halt schön gesehen, dass z.B. diese ganzen Krebse, diese großen Krebse, die da vorkommen, die hauptsächlich im Winter eben im Flachwasserbereich vorkommen, weil es da dunkel ist. Die kommen dahin, um sich zu paaren und im Sommer sind die halt teilweise komplett verschwunden, also die wandern dann scheinbar ins Tiefere. Keiner weiß genau, wo die abbleiben. Im Sommer hat man dafür dann eher Fische oder bestimmte Fischarten. So verschiebt sich das so ein bisschen. Das war so die Kernfrage, wie sich die Gemeinschaft zusammensetzt.

I: Das ist ja auch interessant. Aber acht Monate Auswertung ist schon eine Menge Arbeit oder?

D: Ja.

I: Was hast du denn studiert?

D: Marine Umweltwissenschaften in Oldenburg.

I: In Oldenburg, ja davon habe ich auch schon gehört.

D: Also, das ist so ähnlich wie marine Biologie, ein bisschen allgemeiner gestreut würde ich behaupten.

I: Aber dann auch in Bachelor und Master oder?

D: Ja, ich habe beides in Oldenburg gemacht.

I: Und was hast du dann für Aufgaben im Moment jetzt? Außer der Auswertung.

D: Ich kümmere mich im Moment ganz viel um die Sensoren hier. Wir haben auch verschiedenste Sensoren am Taucherzentrum, die betreut werden müssen. Und wenn z.B. gerade Projekte laufen, z.B. sechs verschiedene Sonden gleichzeitig unter Wasser sind, also CTDs, also Sonden die hauptsächlich Temperatur, Druck, Qualität und Leitfähigkeit messen und die sollen verglichen werden. Also die sind vorher alle kalibriert worden und wird halt geguckt, wie gleich messen die denn wirklich und wie verändert sich das über die Zeit. Und genau, da muss ich mich halt jeden Morgen hinsetzen und die Daten von den Sonden abrufen, abspeichern, umwandeln, dann werden die in so eine Datenbank eingeladen, sodass sich die anderen Institute sich das auch angucken können und runterladen. Das ist so eine Aufgabe hauptsächlich, die im Moment grade läuft. Ja, Bildbearbeitung, klar, also andere Sachen die anfallen, Sachen bestellen, in der Werkstatt irgendetwas reparieren und so.

I: Und wie sieht dann so ein typischer Arbeitstag bei dir aus? Also ich meine, du arbeitest jetzt noch nicht so lange, aber bisher typisch?

D: Ich komme zur Arbeit und setzte mich dann an meine Sensoren, da die Daten abzurufen, weil das dauert ziemlich lange.

I: Was heißt ziemlich lange?

D: Ja kommt auf die Sonde drauf an. Also die Seabird braucht 1.5h ungefähr, um 24h Daten runterzuladen. Das so als erstes. Das sind sechs Sonden, die muss ich alle nacheinander, ja die Kommunikation aufbauen, die Daten abrufen, das dauert also schon im Prinzip ja nicht den ganzen Vormittag, aber einen Teil des Vormittags. Dann habe ich ja wie gesagt dieses Dashboard, also das ist so eine Internetseite, wo die ganzen Daten direkt wenn sie in der Datenbank sind, auch bildlich dargestellt werden. Da muss ich dann gucken, ob das da auch wirklich alles so, die Daten auch auftauchen, wie sie auftauchen sollen, weil das ja meistens dann doch irgendwie nicht ganz so funktioniert. Ja, nachmittags ist es immer etwas unterschiedlich, was gerade so ansteht.

I: Okay, also dann projektabhängig?

D: Genau.

I: Und diese Organismen-Markierung läuft einfach nebenbei, weil das jeder so machen muss? D: Genau, das ist immer, wenn so ein bisschen Zeit ist, dann setzt man sich mal daran. Man muss dazu sagen, ich finde es schwierig, dass stundenlang zu machen. Also auch in meiner Masterarbeit war es so, ich musste ja viel auswerten und ich habe auch viel ausgewertet, aber man kann nicht den ganzen Tag nur Bildauswertung machen, da wird man echt bekloppt, wenn man da auf den Bildschirm starrt und irgendwelche Tiere sucht, sag ich mal. Also eine Viertelstunde am Stück und dann zwei, drei Minuten Pause macht, das geht schon eine Weile, man kann das auch über mehrere Stunden machen, aber irgendwann braucht man eine längere Pause.

I: Kann ich verstehen. Es ist ja doch schon recht monoton und man muss viele Details erkennen.

D: Ja man muss sich ja schon konzentrieren und immer das ganze Bild so abscannen und

monoton ist es auch.

I: Okay, und bei Macro Scheduler, das hast du jetzt gestern zuletzt benutzt, richtig? **D**: Ja.

I: Okay, und wie oft benutzt du den so bisher?

D: Na ja, wenn ich mich an die Bilder dran setze, muss ich das jedes Mal benutzen logischerweise. Im Moment, wenn du nicht wärst, würde ich kaum damit arbeiten. Nein also ich versuche mich jetzt mal jeden Tag immer so ein bisschen dran zu setzen und wie gesagt, ich arbeite jetzt auch noch nicht wieder lange hier. Und so den ersten Monat habe ich dann auch andere Sachen gemacht, aber jetzt vor zwei oder drei Wochen, da hieß es halt, dass Person C auch viele andere Sachen zu tun hat und dass ich das dann so ein bisschen mit übernehmen soll, auch so ein bisschen die Bildauswertung und dann halt auch dir Informationen zukommen zu lassen und in dem Zuge habe ich dann angefangen. Ich werde jetzt gucken, dass ich mich irgendwie ein Mal täglich mich da ein bisschen dran setze und eine Zeit lang nur die Bilder angucke.

I: Und wie lange benötigst du dann für eine Sitzung, sage ich mal?

D: Zwei Stunden oder so, würde ich mal sagen.

I: Das wäre dann die Auswertung für einen Tag, also 48 Bilder.

D: Also in zwei Stunden schafft man schon mehr als einen Tag. Ich müsste mal gucken, wie lange ich für einen Tag brauche, aber das ist auch immer sehr unterschiedlich je nach Qualität der Bilder und auch was drauf ist. Wenn die Qualität oder die Trübung so schlecht ist, dass man nichts sieht, dann ist man da schnell durch, weil dann klickt man immer nur high turbidity und es ist nichts zu sehen, was nicht heißt, dass nichts drauf ist. Man sieht einfach nichts. Wenn die Qualität gut ist und auch viel drauf ist, dann dauert das natürlich länger. Also ich hatte auch schon Monate in meiner Masterarbeit, wie im September, wo Unmengen an Fischen drauf waren und dann dauert das halt länger.

I: Oh ja, das glaube ich. Und wie ist dann der generelle Arbeitsablauf? Also wenn du jetzt das Programm geöffnet hast, was muss man da erstmal machen?

D: Also wenn man das Programm geöffnet hat, muss man natürlich erstmal diese ganzen Rahmen, also er fragt dich danach wo er die Bilder hernehmen soll, wo er sie abspeichern soll. Dann ist da so ein ini-file hinterlegt, das hat dir Person C bestimmt auch gezeigt, wo die ganzen Informationen dann auch nochmal drin sind. Das muss man als aller erstes eingeben, das muss man dann nur einmal machen für diese 48 Bilder. Und dann wird das eigentliche Programm geöffnet. Und dann muss man als ersten den Bildschirm kalibrieren. Also links oben in der Ecke den Cursor positionieren, eins klicken, und unten rechts dann zwei drücken. Dann zeigt er dir die Pixelzahl an, das muss man dann mit dem Bild vergleichen. Das muss man aber auch nur einmal machen und dann fängt die eigentliche Auswertung an. Das ist bei jedem Bild immer das gleiche. Er fragt dich als erstes danach, wie die Trübung ist und man kann das Bild korrigieren. Also man kann, wenn man will, den Kontrast oder die Helligkeit hoch oder runter machen, also heller oder dunkler das Bild. Und es gibt so image equalize, da wird der Kontrast ganz anders, das Bild wird von den Farben her ein bisschen verfälscht, das sieht ein bisschen komisch aus. Aber dadurch, dass der Kontrast viel stärker ist, kann man manchmal ein bisschen besser was sehen auf dem Bild. Also das heißt, man kann diese Korrektur vornehmen und dann im Prinzip sagt man, welche Trübung vorliegt, und dann kann man sagen, ob da ein Fisch oder was auch immer zu sehen ist und klickt da halt jeweils an, was man sieht.

I: Okay, dann klickt man einfach auf den Fisch und dann sagt man das ist ein Fisch?

D: Genau, also da, wo der Fisch zu sehen ist, da klickt man den halt an. Dann taucht rechts diese Eingabemaske auf, wo man dann halt Buttons hat, ist das ein Fisch, ist das eine Benthische Crustacea, ist es eine Pelagische, ist es eine Qualle.

I: Also dann erstmal so die grobe Einordnung. Und wenn du dann fertig mit einem Bild bist, klickst du dann was?

D: Wenn ich jetzt gar kein Tier drauf habe, sage ich ja direkt no object on image, dann verschwindet das Bild und das nächste kommt. Wenn ich jetzt sage, ich habe eine Qualle, dann könnte ich jetzt noch drei Fische eingeben, also dann klicke ich einmal alle Organismen auf dem Bild durch und am Ende muss ich einmal auf den Button drücken all objects counted. Dann weiß er, ich bin fertig damit und dann kommt auch das nächste Bild. Und beim nächsten Bild geht es dann wieder von vorne los, also dass man halt sagen muss, welche Trübung, wie man das Bild korrigieren könnte, wenn man möchte, und dann halt wieder die Organismen anklicken. Und das wiederholt sich dann, bis man alle 48 Bilder durch hat. Und dann hört der automatisch auf und wenn man jetzt weiter machen wollen würde, also den nächsten Tag zum Beispiel. Dann müsste man das Programm nochmal starten und dann wieder am Anfang das neu eingeben. Das ist halt ein bisschen nervig, dass man jedes Mal dem Programm sagen muss, da und da nimm deine Bilder her, da und da speichere sie ab. Also es gibt ja dieses ini-file, wo auch bestimmte Informationen hinterlegt sind. Und genau, da steht dann auch drin, wo er z.B. abspeichern soll. Eigentlich, wenn es im ini-file alles korrekt angegeben ist, wo er sie hernehmen und abspeichern soll, ist das, wenn man das Programm öffnet, alles schon voreingegeben. Und man muss nur kontrollieren, stimmt das. Er fragt halt immer nochmal nach, also man gibt halt ein z.B. nimm die Bilder da und da her bzw. Er schlägt es dir vor, man kontrolliert, ist das jetzt richtig, sagt ok, und dann kommt nochmal so ein Fenster, wo steht: Willst du wirklich diesen Ordner, wo deine Bilder herkommen? Das finde ich persönlich ein bisschen nervig, weil ich beim ersten Mal schon nachgucke, ob es das richtige ist. Es wahrscheinlich eine Kontrolle für Leute, die so flüchtig dadurch gehen und sagen, ah ja passt schon. Das war wahrscheinlich mal so gedacht. Ich finde es persönlich ein bisschen doppelt gemoppelt.

I: Das könnte natürlich auch nochmal dazu verleiten, nochmal schneller durchklicken. So ja ja ja, passt schon alles. Auch für die Leute, die halt nur flüchtig drüber gucken, die klicken vielleicht auch da weiter. Und was frustriert dich noch so an dem Prozess? Also außer, dass man die ganzen Sachen wieder neu eingeben muss?

D: Also, was im Moment ein bisschen blöd ist, ist, dass wenn man das Bild öffnet, muss man irgendwie drei Sekunden warten oder so, bis man letztendlich sagen kann, es ist die und die Trübung. Vorher kann man das nicht anklicken. Das ist aber eine Einstellungssache. Also Person E, die das Programm ja auch so ein bisschen mitentwickelt, also geschrieben hat, diesen Macro Scheduler, hat es halt so eingebaut. Also rein theoretisch kann man das auch wieder raus nehmen. Das ist ein bisschen an der falschen Stelle. Also Person E hat es eingebaut, damit man halt sich mindestens drei Sekunden Zeit nehmen muss, das Bild anzugucken und nicht einfach durchklicken kann. Aber ich finde, das kommt an der falschen Stelle. Weil eigentlich müsste man, ich gucke mir das Bild an, sage ok das ist die Trübung. Da brauche ich keine drei Sekunden für und danach müsste diese Zeit runterlaufen, dass man sich das Bild anguckt. Ich weiß nicht, ob das unbedingt sein muss, dass man so eine aufgezwungene Zeit bekommt, die man warten muss. Das ist wahrscheinlich auch ein bisschen Geschmackssache oder auch ... Person E, der letztlich auch die Verantworung hat und den Leuten sagt, setz dich da dran, hat das vielleicht aus gutem Grund gemacht. Aber ich finde, es ist an der falschen Stelle, man müsste es ein bisschen nach hinten schieben, weil wie gesagt für den ersten Eindruck, für diese Trübung, braucht man keine drei Sekunden. Es geht ja darum, dass ich mir das Bild in Ruhe angucke und nicht, weil ich keine Lust habe, so ah komm, das passt schon, da ist nichts drauf, so ungefähr.

I: Ah klar, okay. Fällt dir noch etwas ein, was nicht so gut an dem Ablauf ist? Also was ich z.B. noch hatte, ist, dass man die Markierungen nicht sieht. Also wenn du auf einen Organismus drauf klickst, dann siehst du ja nicht, wo du hingeklickt hast.

D: Ja stimmt, das ist auf jeden Fall nervig. Manchmal ist es halt so, also ich versuche mir das immer zu merken, wo ich welches Tier schon angeklickt habe. Aber manchmal ist es halt schon so, dass wenn jetzt viele drauf sind, also was weiß ich, 15 Tiere, dann weißt du dann ja nicht mehr, hab ich das schon angeklickt oder nicht. Wobei im Zweifel klickt man dann lieber doppelt, weil im zweiten Schritt, wenn man das Tier vermisst, geht man alle Bilder, wo was drauf war, nochmal durch und dann sieht man nämlich, wo markiert wurde und dann kann man, wenn doppelt markiert wurde, wieder raus nehmen. Es wäre natürlich schöner, wenn man das gleich beim ersten Mal sieht, dass man sieht, aha das hat man angeklickt. Reicht ja ein Punkt oder irgendwas oder ein Kreuz oder keine Ahnung, dass man sieht, aha das ist schon markiert worden.

I: Genau, okay. Was findest du an dem Ablauf dann gut? Kannst du dazu was sagen?

D: Also grundsätzlich ist das Programm vom Aufbau her schon ganz gut, also dass man einfach nur klicken braucht und man auch diese vorgefertigten Buttons hat. Also es ist ja nicht so, dass man da jetzt irgendwelche Pixel aufschreiben müsste, wo man dann halt das Tier gesehen hat oder so. Das wäre ja viel umständlicher. Im Prinzip ist das Programm vom Grundsatz her ja nicht verkehrt, dass man halt einfach aufs Bild klickt, da ist der Fisch, dann drückt man auf den Button, das ist der Fisch, und wenn es 15 Fische sind, sind halt 15 da. Das ist ja vom Grundansatz schon ziemlich gut, dass das so funktioniert.

I: Gut, also hauptsächlich dieses drei Sekunden Warten, das immer wieder eingeben, wenn man das Programm neu startet und dass man keine Markierungen hat, das wäre halt zu verbessern. Sonst ist der generelle Ablauf ganz gut, z.B. dass man nicht mit Pixeln arbeitet, sondern direkt auf dem Bild. Was mir noch aufgefallen war, ist, du kannst nicht zurückgehen. D.h. wenn du irgendwas markiert hast, kannst du es nicht rückgängig machen.

D: Genau, man kann es nicht löschen.

I: Da rennst du praktisch die ganze Zeit vorwärts, also du markierst und markierst und dann wählst du das aus, aber du kannst nicht sagen, ok oh jetzt habe ich mich da vertan, das ist ja gar kein Fisch, das ist eine Qualle gewesen. **D**: Ja, also wie gesagt, in dem zweiten Schritt sieht man dann immer, was geklickt wurde. Und ich hatte das auch schon ein paar Mal in meiner Masterarbeit, dass ich dann irgendwas angeklickt habe, wo ich schon im ersten Schritt gemerkt habe, oh das war jetzt aber falsch, es aber nicht korrigieren konnte. Dann muss man dann hoffen, dass man später im zweiten Schritt, also entweder macht es die gleiche Person oder wenn eine andere Person das macht, darauf hoffen, dass sie das dann sieht, oh das war aber kein Fisch, sondern eine Qualle und dann kann man das in dem Schritt nochmal ändern. Aber es ist ein bisschen umständlich. Es wäre schöner, wenn man sieht, was man angeklickt hat und auch als was man das deklariert hat und dass man sagen kann, ne da habe ich mich jetzt vertan, das ist doch das und das. I: Du hast is auch noch Zeit zwischen den beiden Programmen oder? Du startest is nicht

I: Du hast ja auch noch Zeit zwischen den beiden Programmen oder? Du startest ja nicht Macro Scheduler, machst die 48 Bilder und gehst dann zu Stereo Marker oder?

D: Nein, also normalerweise ist das so, dass man, also das hängt imme so ein bisschen davon ab, wer gerade die Zeit hat bzw. manchmal gibt es halt bzw. bei meiner Masterarbeit gab es ja eine konkrete Fragestellung. Da habe ich halt alle Monate durchweg erst den Schritt eins gemacht, dann habe ich Schritt zwei gemacht. D.h. zwischen dem ersten Bild in der Auswertung von Schritt eins und dann dann wieder dem ersten Bild angucken in Schritt zwei, war dann eine erhebliche Zeitspanne gewesen. Dann ist das schon ewig her und manchmal weiß man dann gar nicht mehr. Dann sitzt man da und denkt, was hast du denn da für einen Quatsch geklickt, so ungefähr.

I: Dann weiß man natürlich auch nicht mehr, welche Fehler man damals schon gemacht hat. D: Vielleicht wäre es da nochmal sinnvoll, wenn man da irgendwie, ja wie so ein Kommentarfenster hätte, dass wenn einem irgendetwas besonderes auffällt, dass man das dahin schreiben kann.

I: Zu den Organismen selber?

D: Also ich hatte das irgendwann mal, da war eine Robbe vor der Linse, da hat man auch nichts gesehen. Das könnte man als Kommentar dann vermerken. Weil man klickt ja dann nur no object on image, aber es liegt ja nicht daran, dass da nichts drauf ist, sondern dass da etwas vor der Linse ist. Oder wir hatten das mal, früher gab es mal eine Phase. Also die Kamera hat ja so Scheibenwischer vor der Linse, weil die Linse sonst bewächst und dann sieht man ja auch nichts. Dieser Scheibenwischer kann aber auch schonmal kaputt gehen, dann ist halt auch Bewuchs auf der Linse. Ich meine, Spitzbergen kommt man ja nicht einfach mal eben hin. Wenn der kaputt geht, dauert das meist mehrere Monate, bis man das reparieren kann, wenn man sowieso hoch fährt. Da ist mal so eine Phase, wo das so zugewachsen ist, dass man kaum noch etwas sieht. Gut, man kann dann überlegen, ob man die Bilder überhaupt auswertet, weil die Qualität halt... Das sind dann halt so Momente wo ich denke, vielleicht wäre das so ein Kommentarfeld gar nicht so schlecht, dass man nochmal was bemerken kann. Weil wenn man sich dann hintereher an die Auswertung macht, kann man ja immer noch entscheiden. Dann sieht man halt Bilder von Datum x bis Datum y sind total bewachsen. Dann kann man sich überlegen, die Bilder nehme ich raus für meine Fragestellung oder sage ich okay, ich lasse sie mal noch drin. Also sowas vielleicht.

I: Das könnte man natürlich bei den Bildnotizen mit reinbringen, also da wo man sagt, wie trüb das Bild ist.

D: Ja irgendwie sowas. Man muss das halt nicht ausfüllen. Wenn da nichts ist, lässt man es halt offen.

I: Okay gut, dann zu Stereo Marker, dem zweiten Programm. Wann hast du das das letzte Mal benutzt?

D: Das ist schon ewig her. Also ich weiß so ungefähr, was man da so tut, aber die ganzen Details kann ich dir nicht sagen. Da müsste ich mich die Tage nochmal hinsetzten und mir das nochmal angucken. Also das schwierigste an diesem Programm ist halt, du musst halt. Also du vermisst ja die Organismen die du gefunden hast, d.h. Du hast dann das linke und das rechts Bild gleichzeitig offen und dann siehst du halt, was da geklickt wurde und als was es deklariert wurde. Dann musst du es einmal, wenn es denn geht, näher spezifizieren, also was genau ist denn das für ein Fisch und du musst sie dann halt vermessen. Und wenn sehr viele Fische auf diesem Bild sind, ist es halt manchmal ein bisschen schwierig herauszufinden, welcher linke Fisch. Also dieser eine Fisch ist ja einmal auf dem linken und einmal auf dem rechten Bild vorhanden und dann musst du die zuordnen. Das ist manchmal nicht einfach zu überlegen, weil die Kamera ja sozusagen aus zwei unterschiedlichen Perspektiven auf diesen Fisch drauf guckt. Und wenn der Fisch jetzt etwas gekrümmt ist, dann ist der Blickwinkel von der einen Seite anders als von der anderen Seite. Und dann muss man immer gucken, ok die Krümmung von dem Fisch auf Fisch auf dem Bild könnte zu dem Fisch mit der Krümmung auf dem anderen Bild passen. Weil das Bild ist auch ein bisschen versetzt logischerweise, weil die Kameras hängen ja so nebeneinander und da ist der Bildausschnitt links und rechts ja nicht exakt der gleiche, sondern es verschiebt sich ja so ein mü. Das Programm schlägt einem immer schon vor, man kann einen Fisch markieren, also vorne und hinten, dann wird da so ein Strich durchgezogen. Dann schlägt das Programm dir schon vor auf der anderen Seite, das könnte der und der Fisch sein. Das stimmt aber auch nicht immer und was auch schwierig ist, ist die Längenvermessung, wenn die Fische gekrümmt sind und nicht gerade, dann ist es natürlich schwierig mit der Länge.

I: Ja, was macht ihr in dem Fall? Wenn der sich jetzt ein bisschen schlängelt, der Fisch?

D: Den abschätzen und, also wenn der jetzt so nach rechts gekrümmt ist, versucht man dann halt trotzdem den Strich gerade zu machen und abzuschätzen, wie lang er denn wäre, wenn er nicht gekrümmt wäre. Aber das ist glaube ich recht ungenau.

I: Okay, das heißt ihr markiert Kopf und Schwanz, dann zieht ihr die Linie da durch und dann könnt ihr irgendwie separat in einem Feld die Länge nochmal korrigieren. Oder wie funktioniert das?

D: Also du markierst normalerweise den Kopf und den Schwanz, aber der macht immer eine gerade Linie, d.h. Wenn ich jetzt einen gekrümmten Fisch habe und ich markiere den Kopf und den Schwanz, dann nimmt der ja die kürzeste Strecke sozusagen und dann ist ja die Länge viel kürzer, als wenn der Fisch jetzt gerade wäre. Was man dann machen kann, ist, dass man den Kopf markiert und dann den zweiten Punkt nicht am Schwanz setzt, sondern ein bisschen weiter nach links oder rechts, je nachdem wohin er halt gebogen ist. Das setzt man da, wo man denken würde, wenn der Fisch gerade wäre, verstehst du?

I: Ja, ich habe es verstanden.

D: Das ist ehrlich gesagt ein bisschen, ich glaube, dass es recht ungenau ist. Also Person

E, der jetzt Jahrzenten mit Fischen arbeitet, der kann das vielleicht ganz gut abschätzen, aber er wertet auch die Bilder eigentlich nie aus, weil Person E keine Zeit dafür hat. Aber alle anderen Leute, die ... Also ich glaube man kann das als nicht total Erfahrener schlecht abschätzen.

I: Ja okay, da wären vielleicht so Linien ganz sinnvoll, die man unterbrechen kann, dass man mehrere kleine Linien hat.

D: Genau, oder es ist wie so ein Freihand-Tool. Also bei so Zeichenprogrammen gibt es immer diese Freihand-Linien, was man dann einfach mit einer Kurve machen kann. Also das Programm kann ja nur im Moment gerade Striche und dass man dann in der Lage ist, auch Bögen zu machen. Das wäre vielleicht ganz gut, also eine Form, die halt der echten entspricht. Oder, wie du sagst, man macht halt gerade Linien, aber man darf halt mehrmals unterbrechen und muss dann sagen, ich fange jetzt an, unterbreche ein paar Mal und muss dann natürlich irgendwann erkennen, dass da das Ende ist.

I: Ja genau, okay, das ist schonmal ein Problem bei dem Programm, dass man halt diese Linie nicht richtig der Form anpassen kann.

D: Also das ist nicht nur bei den Fischen ein Problem, das ist auch ein Problem z.B. bei Tieren, die er nur halb sieht. Also, es gibt ja auch Baumstämme, Algen und sonst was Unterwasser. Manchmal hat man halt Krebse, die nunmal halb drauf sind, z.B.. Und entweder sagt man dann, okay, ich lasse das mit der Vermessung, man kann halt auch sagen, ich kann sie einfach schlichtweg nicht vermessen oder man muss dann halt auch wieder abschätzen.

I: Ja, das könnte schwierig sein. Man könnte vielleicht eine Standard-Länge von dem Tier dann nehmen. Man hat ja immer so eine Durchschnittslänge.

D: Genau, und man misst ja auch zwei Längen, also man misst ja z.B. einmal die Länge und die Breite, sozusagen. Und es gibt da bestimmt auch Literatur-Erfahrungswerte, müsste ich mal nachgucken, dass wenn man einen Wert komplett messen kann, z.B. beim Krebs die Breite vielleicht, da gibts bestimmt so ein Verhältnis, das könnte man bestimmt ergänzen.

I: Ja, das hatte Person E schonmal erzählt, dass dieses Verhältnis im Program mit aufgenommen werden soll.

D: Bis jetzt war das dann halt immer so, entweder ich habe dann halt beide Längen messen können oder ich habe halt gesagt, okay, ich kann es gar nicht vermessen. Und es wäre halt cool, wenn man halt sagen kann, okay ich messe die eine Breite, die ich sehe, und die andere weiß ich nicht. Da kann man ja den Anfang markieren und sagen, das Ende sehe ich nicht. Und vielleicht kann man da dann irgendwelche Daten hinterlegen, so Verhältnisse, da kann der das dann vielleicht ausrechenen. Das ist dann natürlich nicht der echte Wert. Das kann man ja dann irgendwie markieren, dass man da irgendwie irgendwas hinterschreibt, dass man weiß, dass ist gerechnet und nicht gemessen.

I: Ja, aber ich glaube, das würde den Rahmen schon ein bisschen sprengen.

D: Ja bei dir bestimmt.

I: Und generell, der Arbeitsablauf bei diesem Programm, wie sieht das da ungefähr aus? Also am Anfang musst du z.B. auch irgendwie die Daten laden.

D: Ja das weiß ich jetzt gerade nicht mehr. Was man auf jeden Fall dazwischen noch machen muss, nach dem ersten Schritt, bevor du die Bilder dann eignetlich in dem zweiten Schritt

vermessen kannst, musst du die Bilder nochmal rektifizieren. Also du musst im Prinzip, die Kameralinse hat ja eine Krümmung und diese Linsenkrümmung, die wird rausgerechnet und du musst halt, bevor man diesen Stereo Marker überhaupt benutzen kann, muss man die erstmal rektifizieren. Das macht man mit Matlab. Und das muss vorab einmal gemacht werden. Das dauert ziemlich lange, aber das kann man abends anstoßen, dann läuft das über Nacht und dann ist man da morgens mit fertig. Also das ist jetzt nicht unbedingt das Problem. Und dann, wenn man Stereo Marker öffnet. Ich weiß es jetzt ehrlich gesagt nicht mehr so ganz genau. Also man muss dem auf jeden Fall natürlich sagen, wo er die Bilder hernehmen soll.

I: Ich habe gesehen, dass man da auf jeden Fall noch die Tabelle mit sieht, dass die da angezeigt wird. Ich gucke nochmal eben nach. Hast du die benutzt? Hast du also in die Tabelle rein geguckt? (Zeige Screenshot von Stereo Marker Oberfläche) Okay, siehst du etwas? Okay, das war das Controller Window. Also es gibt ja irgendwie zwei Fenster bei Stereo Marker und das ist das eine und das andere war das hier, was du gesagt hattest, wo man linkes und rechtes Bild sieht, wo man dann die Spezies eingeben kann, irgendwie noch eine Description.

D: Genau und dann hat man dann noch so Beispielbilder von den Organismen, die man sieht. Weil man die ja genau spezifizieren soll.

I: Ist das hilfreich?

D: Ja auf jeden Fall. Einige Tiere kennt man natürlich, aber gerade so verschiedene Quallenarten waren mir jetzt nicht so geläufig, als ich damit angefangen habe und ich habe das auch ergänzt mit ein paar Bildern damals. Es ist auf jeden Fall hilfreich, gerade für Leute, die damit neu arbeiten, dass man weiß ungefähr, das könnte das oder das oder das sein. Also klar, die wiederholen sich auch, aber für den Anfang, wenn man sich noch nicht auskennt, ist das natürlich hilfreich.

I: Das macht Sinn. Das hier war das zweite Fenster, da zeigt der hier unten irgendwie so die Tabelle an.

D: Genau, links steht dann immer rectified. Also das ist dann immer dieses Bild, das dann schon rektifiziert wurde, dann stehen da die Koordinaten von den Tieren drauf, die angeklickt wurden. Und wenn man das jetzt vermessen hat, dann schreibt der in diese Tabelle dann auch die Länge rein.

I: Guckt man sich das in der Tabelle an? Also benutzt man das?

D: Also ich habe das schon öfter reingeguckt, weil man halt daran sieht, ob der, wenn du die Länge vermessen hattest, das auch wirklich da rein schreibt. Oder rein theoretisch kann man da auch Zeilen rauslöschen oder neu einfügen. Also ich hatte da z.B. auch schon, da war ein Tier markiert, wo ich gedacht habe, ne da ist doch gar keins. Da kann man die Zeile dann z.B. rauslöschen aus der Tabelle. Oder wenn jetzt drei Fische angeklickt wurden und es sind aber vier, z.B., dann kann ich auch eine Zeile einfügen. Also man arbeitet schon auch mit dieser Tabelle.

I: Okay, d.h. du fügst alles, also wenn du ein neues Tier hast, fügst du das in der Tabelle ein und nicht auf dem Bild.

D: Genau, ich meine, man muss das in dieser Tabelle einfügen, weil auf dem Bild selber

kann man das nicht. Also man kann auf dem Bild nur vermessen im Prinzip und halt das Bild sehen. Aber wenn du Veränderungen vornehmen willst, muss du das in der Tabelle machen. Also man kann halt wie gesagt Zeilen rauslöschen, einfügen, die Länge wird dann da angegeben. Ich weiß gar nicht, ob da in der Tabelle auch ein leeres oder Bewertungsfeld ist, wo man auch was reinschreiben kann. Auf jeden Fall, wenn man Veränderungen vornehmen will, muss man das in dieser Tabelle machen. Letztendlich ist die Tabelle, die man da sieht, auch das, was er dann wegschreibt sozusagen, was als Ergebnis dann rauskommt.

I: Aber in dem ersten Programm markiere ich doch nur. Also wenn ich einen Fisch habe, mache ich ja nur einen Punkt für diesen Fisch. Egal wo auf dem Fisch. Und jetzt in dem Programm muss ich doch wissen, wo Kopf und Schwanz sind.

D: Ja genau, also das Bild taucht halt auf, das linke und das rechte und dann hat man auf dem Bild ganz viele Punkte, wo Tiere markiert wurden. Das siehst du jetzt nicht in deinem, da ist ja gar kein Bild. Und da kannst du in der Tabelle auf eine Zeile klicken und wenn du eine Zeile markiert hast, dann wird dir im Bild der Punkt farbig hervorgehoben. Also dann weißt du, welche Zeile zu welchem markierten Tier gehört. Die Kreuze sind vorher glaube ich alle grau und wenn ich jetzt eine Zeile in der Tabelle anklicke, dann wird das Kreuz meinetwegen rot. Dann weiß ich, aha, die Zeile, wo ein Fisch drin steht, ist der Fisch, der da markiert wurde. Dann kann ich sagen, okay, den möchte ich vermessen.

I: Okay und die Vermessung machst du dann wie genau?

D: Der Fisch ist markiert worden so grob. Und dann tauchen da halt, wenn du diese Zeile angeklickt hast, tauchen da zwei Kreuze auf, ein rotes und ein blaues. Und dann muss man halt das erste Kreuz an die Spitze von dem Fisch machen und das zweite Kreuz hinten an die Schwanzflosse. Also die Kreuze so auseinanderziehen. Die beiden Kreuze sind halt am Anfang um diesen Punkt drumherum, den ich ursprünglich markiert habe, weil er nicht weiß wie lang der Fisch ist. Und dann muss ich diese beiden Kreuze so positionieren, dass ich den Anfang und das Ende des Fisches habe. Und dann muss ich halt auf dem rechten Bild gucken, dass das der richtige Fisch ist, der dazugehört. Also wenn ich die Kreuze auf dem linken Bild setze, wird auf dem rechten das Pendant dazu sozusagen gesetzt. Aber manchmal passt das nicht genau, dann muss man das nochmal verschieben.

I: Okay, d.h. man hat zwei Bildschirme und dann hat man links die Bilder offen und rechts hat man dann den Controller offen, also die Tabelle?

D: Man hat links ein Bild, rechts ein Bild, also wir haben immer zwei Bildschirme, auf dem linken ein Bild und auf dem rechten das andere Bild und die Bilder, also das was du da jetzt hast, da ist das eine Bild, ne warte. Erzähle ich jetzt Blödsinn?

I: Also das, was ich geöffnet hatte, war hier so untereinander.

D: Ja, kann auch sein. Ich überlege gerade, was dann auf dem anderen Bildschirm ist, weil eigentlich braucht man dafür immer zwei Bildschirme.

I: Man hat noch die Tabelle offen, oder? Da muss man doch dann auf jeden Eintrag draufklicken, damit man dieses Tier vermessen kann. D.h. man hat auf der einen Seite die Tabelle offen, man klickt dann auf den ersten Eintrag und auf dem linken Bildschirm vermisst man das dann, dann geht man wieder auf den rechten Bildschirm, klickt auf den nächsten Eintrag und dann auf den linken Bildschirm und vermisst das dann. D: Ja, im Prinzip ja.

I: Und was findest du an dem Programm schlecht? Fällt dir dazu noch etwas ein?

D: Ja wie gesagt, ich habe das schon lange nicht mehr gemacht. Daher kann ich das jetzt ehrlich gesagt nicht so ganz genau sagen. Aber, wie gesagt, was ich schwierig fand teilweise, ist dieses Rausfinden, welcher Fisch gehört zu welchem auf den beiden Bildern bzw. das mit der Krümmung, worüber wir schon gesprochen haben. Ja die Identifizierung der Tiere ist halt auch manchmal schwierig, weil die Qualität der Bilder oft bisschen zu Wünschen übrig lässt, aber das hat ja was mit der Kamera, nichts hier mit dem Programm zu tun. Ansonsten... ich weiß es nicht, ich müsste es nochmal machen.

I: Ja, das ist ja kein Problem. Ist ja einfach mal, um einen ersten Eindruck zu kriegen. Aber ich glaube, damit hätte ich meine Fragen soweit beantwortet. Vielen Dank!

B. Hierarchical Task Analyses

B.1. Macro Scheduler

0. Roughly annotate organisms on multiple images

- 1. Prepare session
- 2. Load image
- 3. Screen calibration
- 4. Edit current image
- 5. Open next image
- 6. Exit the program

Plan 0: 1, 2 - 5 as long as there is another image, 3 only for the first image, then 6

1. Prepare session

- 1.1 Correct ini file
- 1.2 Confirm ini file
- 1.3 Correct year
- 1.4 Confirm year
- 1.5 Correct month
- 1.6 Confirm month
- 1.7 Correct result file
- 1.8 Confirm result file
- 1.9 Correct image input directory
- 1.10 Confirm image input directory
- 1.11 Enter experiment ID
- 1.12 Enter image prefix
- 1.13 Enter user ID
- 1.14 Confirm search string for files
- 1.15 Confirm number of images to be processed

Plan 1: 1.1 if needed, 1.2, 1.3 if needed, 1.4, 1.5 if needed, 1.6, 1.7 if needed, 1.8, 1.9 if needed, 1.10, 1.11 - 1.14, 1.15 (if number of images to be processed is not zero)

3. Screen Calibration

- 3.1 Press upper left corner of target image, then 1
- 3.2 Press lower right corner of target image, then 2
- 3.3 Correct image resolution in x dimension
- 3.4 Confirm image resolution in x dimension
- 3.5 Correct image resolution in y dimension

3.6 Confirm image resolution in y dimension Plan 3: 3.1, 3.2, 3.3 if needed, 3.4, 3.5 if needed, 3.6

4. Edit current image

- 4.1 Select an image filter
- 4.2 Add an image remark
- 4.3 Add organism
 - 4.3.1 Annotate organism position (by clicking on it in the image)
 - 4.3.2 Select organism group from a list
- 4.4 Click beside the image and select "All objects counted"

Plan 4: 4.1 - 4.2 as long as another filter is desired, 4.3 as long as there is another undetected organism on the image, 4.4

Plan 4.3: 4.3.1, 4.3.2

B.2. Stereo Marker

0. Measure organisms on multiple images

- 1. Select image path
- 2. Set camera configuration
- 3. Edit image pairs
- 4. Exit the program

Plan 0: 1 - 3, 4 if user wants to finish

3. Edit image pairs

3.1 Select one row in the data table, i.e. one organism (controller window)

3.2 Drag picture detail rectangle to organism position (controller window)

- 3.3 Click button "delete record" (controller window)
- 3.4 Click button "edit record" (controller window)

3.5 Adapt image scrolling synchronization between left and right image (stereoview window)

- 3.6 Choose image display mode (left, right or both) (stereoview window)
- 3.7 Adapt contrast and/or brightness of left image (stereoview window)
- 3.8 Adapt contrast and/or brightness of right image (stereoview window)
- 3.9 Edit current organism (stereoview window)

3.10 Go to next organism (stereoview window)

3.11 Save table (controller window)

3.12 Export table to CSV (controller window)

Plan 3: 3.1 - 3.2, 3.3 - 3.8 if desired. in an arbitrary sequence at any time, 3.9 - 3.10 as long as another organism exists, 3.11 if desired, 3.12 if desired

3.9 Edit current organism (stereoview window)

3.9.1 Click on button for organism head on right image

3.9.2 Click on organism head on right image

3.9.3 Click on button for organism tail on right image

3.9.4 Click on organism tail on right image

3.9.5 Click on button for organism head on left image

3.9.6 Click on organism head on left image

3.9.7 Click on button for organism tail on left image

3.9.8 Click on organism tail on left image

3.9.9 Drag points to correct positions

3.9.10 Select organism species

3.9.11 Click button "last value" to select previous species

3.9.12 Add description

Plan 3.9: 3.9.1 optional, 3.9.2, 3.9.3 optional, 3.9.4, 3.9.5 optional, 3.9.6, 3.9.7 optional, 3.9.8, 3.9.9 if positions are not correct, 3.9.10 or 3.9.11, 3.9.12 optional

3.10 Go to next organism (stereoview window)

3.10.1 Click button "save & next"

3.10.2 Click button "clear/new"

3.10.3 Click button "as new"

3.10.4 Click button "save/update"

Plan 3.10: 3.10.1 if current organism is finished or 3.10.2 to reset current positioning or 3.10.3 to copy current organism and add it as new entry, 3.10.4 optionally to update table

C. Prototype

C.1. Slideshow

Paper Prototype

Somewhat different...



1

Some Personal Information

Name:

Gender:

Nationality:

Age:

Employment (work or course of study):

Regularity of image annotations:
How does it work?

- Similar to a paper prototype
 - You get a task
 - You try to fulfill the task by performing actions with the program
 - I execute your actions
- But different
 - Online (actions might be delayed)
 - Click is visualized by red circle
 - When you triggered an action, I will say "action"
 - When I am done executing the action, I will say "ok" (and you can continue)
 - My components: on side of slides
- Data processing (sign privacy statement)
 - I will record screen and voice
 - Only for the scope of my thesis

... Let's start!





Welcome Hello and welcome to MarOMarker (Marine Organism Marker). Hease fill in the following fields. You can change them later in "Settings". Inter your personal ID. Your ID Choose a camera configuration file. Image termate Image termate Image termate Velcome Image termate Imag	Welcome Hello and welcome to MarOMarker (Marine Organism Marker). Please fill in the following fields. You can change them later in "Settings". Inter your personal ID. Your ID Choose a camera configuration file. path Browee K	•	<u>^</u>	1 – ¬×
Welcome X Hello and welcome to MarOMarker (Marine Organism Marker). Please fil in the following fields. You can change them later in "Settings". Inter your personal ID. Your ID Choose a camera configuration file. Path K	Welcome X Hello and welcome to MarOMarker (Marine Organism Marker). Please fill in the following fields. You can change them later in "Settings". Enter your personal ID. Vour ID Choose a camera configuration file. path Browse	L T Image remark 💌	ডি+ → ≡ ৩	=
			Welcome × Hello and welcome to MarOMarker (Marine Organism Marker). Please fill in the following fields. You can change them later in "Settings". Enter your personal ID. Your ID Choose a camera configuration file. path Browse	





Task 2: Load Data

To start, you need to load the image data of the 01.06.2019 and apply the neural network to it. Get comfortable with the application, look around. Tell me, what you are thinking and doing.

9

























	\$		1	>			
Settings							
o Camera Settings		User					
Config C:/Users/Desktop/cameraConfig/ Y-offset Distance between cameras Distance between chip and lense	Browse Save I 0 pixel 156 mm 4355 pixel	D	mm				





References

- Images
 - Fish (first slide): <u>http://clipart-library.com/clipart/kTMRjijTj.htm</u> (last access: 13.08.2020)

C.2. Test Protocol Proband A

Task One

Clicks on MarOMarker icon.
A: Okay, I don't have an ID.
I: It is just the first letter of your first name plus the first letter of your second name.
A: Okay, so XY?
I: Yes. Cou can just enter it there.
A: I can do this?
I: Yes.
A: Okay.
Enters ID in ID-field. Clicks on browse-button.
A: Okay, I don't know what to do with this. So I click on this thing.
Clicks on config_file.txt.
I: I am sorry, the explorer is in German. You can just click "Öffnen" to open the file.

Task Two

A: Hm. Okay.

Clicks on menu and navigates to data-page. Clicks on browse-button to browse for an image directory.

A: I have to think now where this file would be. I guess in "Documents" or "Downloads", or? Clicks on "Documents" in file explorer.

I: I actually preselected the folder for you.

A: Ah, okay. I didn't understand that.

Clicks ok-button. Clicks run-neural-network-button.

I: Error: Please select a result file before running the neural network.

A: Ah, okay.

Clicks on browse-button to browse for a result file. Clicks on "result_2019_06.txt" in the explorer and clicks "Öffnen" to load the file.

A: Do I need to type anything for this "Image prefix" and "Experiment ID"?

I: No, these are just optional.

A: Okay.

Clicks on run-neural-network-button.

Task Three

Clicks on correct-predictions-button.

A: I suppose that these circles should be where the head is. I want to correct that. Drags the circle for fish 1 to the correct place.

A: And then the tail is correct here.

Drags blue circle of the false positive to the fish head that was not found (fish 3).

A: I am putting an X on the fish's tail.

Drags blue cross of the false positive to the fish tail that was not found (fish 3).

Clicks on fish 1, then on the group-field in animal-specifications-widget of the current fish and selects group "Fish". Selects "Mackerel" in species-field. Clicks on fish 3 and selects species "Mackerel".

A: I guess they don't need a remark.

Clicks on fish 2 and selects group "Fish".

A: So that would turn the X and the O blue, right?

I: Correct.

Selects species "Mackerel" for fish 2.

Clicks on the undetected crustacean.

I: No action.

A: I have to figure out how to put an X there.

Clicks on menu.

A: Hm. I don't know anymore. What is this button?

Clicks on image-switch-button (switches to right image).

A: Hm okay, is this a new picture?

I: No it is the second picture of the scene, the right image.

A: Oh, let's go back.

Clicks on image-switch-button (I switched to left image instead of LR view - my mistake). Clicks on menu.

A: Wait, maybe I can use this thing.

Clicks on add-animal-button and clicks on the crustacean.

A: Oh no, we need the X, too.

Clicks on add-animal-button.

I: Error: Please select the tail before switching off the add mode.

Clicks on the tail of the crustacean. (Fig. C.1)

Clicks on image-switch-button (switches to right image).

A: I want to correct this fellow.

Clicks on fish 3. Changes group from "Jellyfish" to "Fish". Selects species "Mackerel".

A: So I need to put the X of it (fish 1) here (where the tail of fish 1 is) and the O over here (where the head of fish 1 is). Where is the O? Come here.

Drags circle from tail of fish 1 to its head and drags X from head of fish 1 to its tail.

Clicks on annotation where no animal is.

A: I choose backspace to get rid of this.

I: No action. (Fig. C.2)

A: Hm, is there a subtract on here?

Clicks on remove-button.

A: Oh, but it got rid of this thing?

I: No.

A: But... Maybe I have to drag it over here?

Drags the circle to the bin.

I: That is not possible.

Clicks on the animal.

C. Prototype



Figure C.1.: Proband A - Adding Crustacean



Figure C.2.: Proband A - Removing Annotation

A: Ah, so now the yellow things are deleted?

I: Yes.

Clicks on crustacean. Drags X to tail and O to head (the labels were switched). Selects species "Benthic Crustacea".

I: Okay, you have corrected all the animals!

Task Four

Clicks on fish 1.

A: I thought at first that I would click "Length" (in the animal specifications window) to do that, but it doesn't work.

Navigates to settings-page. Clicks on X to leave the application.

I: Do you want to close the application?

A: Oh no, I just wanted to go back out of this thing (settings page).

Clicks on save-button (for saving the camera config file). Clicks on top left corner (icon of application).

I: No action.

Navigates to home-page.

A: Hm. I mean I guess I have to label them, maybe. I don't know.

I: What do you mean by labelling?

A: Like "1" and "123", like that. But I don't know how to do that.

I: I am sorry, that was my fault, I did not switch the images correctly. With the image switch button you can switch between the left, right and both images.

A: Ah, okay! Okay, now I understand.

Drags bounding box from upper image to lower image.

I: That is not possible.

Clicks on fish 1 on right image (match is correct). Clicks on fish 2 on right image (matches fish 1 on left image and fish 2 on right image). Clicks on fish 2 on left image (this changes selection on right image to fish 1).

A: What? (Fig. C.3)

Clicks on fish 2 on right image (match is correct). Clicks on fish 3 on left image (match with fish 1 on right image). Clicks fish 3 on right image (match is correct). Clicks on crustacean on right image (matches fish 3 on left image with crustacean on right image).

A: What? But I just corrected this.

Clicks crustacean on left image (match with fish 3 on right image). Clicks crustacean on right image (correct match).

A: Okay, I am done.

C.3. Test Protocol Proband B

Task One

B: Ich sehe jetzt diesen Bildschirm mit diesem wellenartigen, ozean-mäßigen Muster, sehe

C. Prototype



Figure C.3.: Proband A - Matching Animals

diesen Fisch mit MarOMarker und gehe jetzt zu diesem Fisch hin und möchte gerne drauf klicken. Ich habe jetzt darauf geklickt.

Klickt auf MarOMarker Icon auf dem Desktop.

B: Jetzt sehe ich ein großes Menü-Fenster, wo in der Mitte ein Fenster aufgepoppt ist mit "Hello und welcome to MarOMarker" und "Enter your personal ID". Das ist ein bisschen klein geschrieben und irgendein file soll ich öffnen. Also ich gehe jetzt mal davon aus, dass in der normalen Welt ich jetzt eine ID hätte, die würde ich da jetzt eintragen.

I: Genau, da nimmst du einfach den ersten Buchstaben deines ersten Namens und den ersten deines Nachnamens.

B: Okay, ich bin jetzt auf das Feld gegangen, habe es angeklickt und gebe jetzt XY ein. Enter. Gibt ID in das ID-Feld ein und bestätigt mit der Enter-Taste.

B: Jetzt gehe ich auf das nächste Feld. Das ist wirklich sehr klein, ich kann das kaum lesen. I: Da steht: "Choose a camera configuration file".

B: Da wüsste ich jetzt nicht genau, was ich machen soll.

I: Darunter ist ein Feld, wo "path" drin steht und daneben steht ein Browse-Button.

B: Ja, das stimmt. Also ich würde jetzt einfach etwas herumklicken. Ich würde jetzt erstmal mit der Maus auf "path" gehen und mal gucken, was da passiert und klicken.

Klickt auf das Feld, das den Kamerakonfigurationspfad anzeigt.

I: Keine Aktion.

B: Dann gehe ich hin zu "Browse" und klicke da drauf.

Klickt auf den Browse-Button.

B: Ein anderes Fenster hat sich jetzt geöffnet, aber das ist jetzt noch kleiner.

I: Moment, ich mache es mal etwas größer.

B: Jetzt kommt etwas. Also ich denke mir mal, dass ich da drauf klicke, auf das gehighlightete camera_config, ich klicke drauf und enter.

Klickt auf die vorausgewählte Kamerakonfigurationsdatei.

B: Das habe ich jetzt anscheinend ausgewählt, dann würde ich auf "OK" gehen, klicke darauf. Bestätigt die Datei durch einen Klick auf den Ok-Button.

I: Sehr gut, damit hast du Aufgabe 1 erfolgreich abgeschlossen.

Task Two

B: Ich soll das jetzt erstmal öffnen. Dann würde ich oben auf Menü gehen und klicken. Klickt auf das Menü.

B: Jetzt hat sich dieses "Home, Data, Settings" geöffnet. Ich möchte ja etwas laden. Ich bin mir jetzt nicht ganz sicher, wo ich es finde, würde vielleicht aber unter "Data" gehen, weil ich mir denke, dass da Daten drin zu finden sind. Also gehe ich jetzt auf "Data". Navigiert zur Data-Seite.

B: Okay, jetzt sehe ich halt auf der linken Seite einen Kalender und auf der rechten Seite einen "Check". Hmm. Da steht noch "Run Neural Network", d.h. das muss ich ja gleich auch noch machen. Aber das wäre vielleicht der Schritt, der danach kommt. Input directory, filter options, not checked. Ich bin mir jetzt nicht sicher, wie ich die Daten hochgeladen kriege, weil input directory, da muss ich ja schon irgendetwas für wissen. Ich muss ja schon irgendwie eine Koordinate wissen. Das scheint schon etwas zu sein, wo ich etwas schon hinterlegt habe. Irgendetwas ist ja schon gemacht worden dann, irgendwo ist ja etwas hinterlegt worden. Result file, das ist ja dann schon eine Ergebnisdatei, also irgendwas ist ja davor schon passiert. Also würde ich da jetzt erstmal wieder raus gehen. Dann würde ich erstmal wieder auf "Home" gehen, also auf diese drei Stiche und würde jetzt auf "Home" gehen. Navigiert zur Home-Seite.

B: Jetzt würde ich mich etwas ärgern, weil ich denke, da war ich ja eben schon. Hmm. Die Aufgabe sagt ja, ich soll etwas laden und darauf das neuronale Netzwerk anwenden. Ich soll Daten laden vom 01.06.2019. Aber ich weiß nicht, wie ich daran komme. Lass nochmal gucken. Wenn ich jetzt das Plus drücke, macht es ja den Bildschirm größer, nehme ich an.

Klickt auf den Tier-hinzufügen-Button. I: Keine Aktion (da kein Bild geladen).

I: Keine Aktion (da kein bild gelade

B: Ich gehe mal auf diese Lupe. Klickt auf den Zoom-Button.

I: Keine Aktion.

B: Ich gehe mal auf den Pfeil.

Klickt auf den Vorheriges-Tier-Button.

I: Keine Aktion.

B: Hm, ich gehe mal auf das Zeichen "L".

Klickt auf den Bildwechsel-Button.

I: Keine Aktion.

B: Und hier drauf.

Klickt auf den Filter-Button.

I: Keine Aktion.

B: Okay, dann gehe ich wieder zurück auf diese drei Striche. Dann gehe ich auf Settings. Navigiert zur Settings-Seite.

B: Camera-Settings. Da haben wir doch eben von gesprochen. Das sind ja die zwei Kameras. Aber damit kann ich ja jetzt eigentlich nur berechnen, wie groß so ein Fisch ist. Aber die ist ja geladen, das ist ja schonmal ganz gut. Aber hier kann ich die Daten ja immer noch nicht laden. Also gehe ich jetzt wieder zurück. Home hat mir nichts gebracht, Settings hat mir auch nichts gebracht. Also muss ich nochmal auf Data gehen.

Navigiert zur Data-Seite.

B: Verdammt, input directory. Vielleicht... ah moment. Ich gehe jetzt hier auf den Kalender und gehe hier auf den ersten Juni 2019 und klicke.

Wählt den 01.06.2019 im Kalender aus.

B: Und jetzt "Run Neural Network", d.h. ich würde jetzt hier unten auf diese Leiste gehen und Enter drücken.

Klickt auf den Button, um das neuronale Netz zu starten.

B: Aufgabe beendet.

Task Three

B: Jetzt brauche ich ja aber das Bild. Ich muss ja irgendein Bild haben von irgendwelchen Kameras, wo ich etwas korrigieren kann. Also, dann irgendetwas korrigieren wäre jetzt für mich hier unten "Correct Predictions" und da klicke ich jetzt drauf.

Klickt auf den Korrigiere-Vorhersagen-Button.

B: Also, ich sehe jetzt einen grauen Bildschirm mit Fischen und einer Krabbe, Fischen in verschiedenen Größen. Ich sehe Group, Species, Remark und Length als Eingabefelder und ich sehe Kreuze und Kreise in Blau und in Orange. Ich sehe, dass ein orangenes Kreuz mit Kopf und Schwanz richtig positioniert ist und mehrere falsch oder gar nicht erfasst sind. Da ist jetzt der große Fisch in der Mitte, der ist auch so eingekreist mit einem rechteckigen Rahmen. Der ist nur als Schwanz markiert und jetzt möchte ich... Ich gehe davon aus, weil da ein Rahmen drum ist, dass ich dieses Tier jetzt bearbeiten kann und sage ja, dass ist die Spezies Fisch. Jetzt würde ich gerne diesen Kreis hochziehen zum Kopf.

I: Das kannst du einfach so machen.

B: Ich nehme mir den jetzt, zack meine linke Maustaste und lasse los. Das passt.

Zieht den Kreis von Fisch 1 an die korrekte Position.

B: Jetzt möchte ich gerne diesen Fisch, den kleinen, den würde ich noch erfassen. Da würde ich jetzt auf den Fisch einmal drauf klicken. Und hoffe, dass sich dieser weiße Rahmen über den Fisch setzt.

Klickt auf Fisch 3 (nicht erkannt vom Netzwerk).

I: Keine Aktion.

B: Hm, okay. Dann gehe ich mal hier runter und nehme das blaue Kreuz und ziehe das blaue Kreuz hier an den Schwanz und lasse los.

Zieht das blaue Kreuz (vom false positive Fisch) zum Schwanz von Fisch 3. (Fig. C.4)



B: Dann nehme ich mir den blauen Kreis und ziehe den bis zu dem Kopf und lasse los. Das

Figure C.4.: Proband B - Dragging Circle to Head

ist nicht ganz gerade, ich würde das jetzt nochmal nehmen und noch ein Stückchen höher setzen, so. So finde ich es besser.

Zieht den blauen Kreis (vom false positive Fisch) zum Kopf von Fisch 3.

B: Jetzt stellt sich für mich die Frage, warum ich zwei verschiedene Farben an Kreuzen habe, der obere Fisch hat ein gelbes Kreuz, die unteren Fische haben blaue Kreuze. Da bin ich mir etwas unsicher, was das bedeutet. Vielleicht ist das aufgrund einer anderen Spezies. Jetzt habe ich hier unten noch diesen Krebs. Ich habe aber keine Xe und keine Kreise mehr. Ich weiß jetzt also nicht, wie ich den markieren kann. Ich klicke einmal darauf auf den Krebs. Klickt auf das unmarkierte Krustentier.

I: Keine Aktion.

B: Ich würde ja gerne eine Markierung dazu setzten. Vielleicht gehe ich mal hier auf Spezies auf diesen Pfeil und klicke da mal drauf.

Klickt auf das Species-Feld im Tierspezifikations-Feld.

B: Dann habe ich da diese Krabbe gefunden. Aber oh ich gehe nochmal zurück. Ich gehe mal auf Fisch und klicke da mal.

Klickt auf das Group-Feld im Tierspezifikations-Feld.

B: So okay. Da nehme ich jetzt erstmal diese Krabbe, das rote Feld und klicke.

Wählt "Crustacea" als Tiergruppe aus.

B: Was habe ich getan? Das war schlecht. Okay, ich muss das Ganze rückgängig machen. Ich klicke jetzt hier oben auf diesen Return-Pfeil. Ich hoffe, dass es nicht aktualisieren ist, sondern rückgängig machen. Da klicke ich jetzt drauf.

Klickt auf den Rückgängig-Button.

B: Ich freue mich. Ich habe nämlich herausgefunden, dass man etwas rückgängig machen kann. Was heißt denn Remark (im Spezifikations-Feld)?

I: Das ist eine Bemerkung.

B: Aber hmm wo kann ich denn ein Tier, vielleicht unter Plus?

Klickt auf dem Zoom-Button.

I: Das zoomt das Bild heran.

B: Und Plus?

Klickt auf den Tier-hinzufügen-Button.

B: Was hat er getan? Was ist passiert? Das Plus ist blau. Okay. Dann gehe ich jetzt auf die Krabbe und klicke da.

Klickt auf das unmarkierte Krustentier.

B: Hey, ich habe den Kopf markiert. Jetzt müsste ich ja auch noch den Schwanz irgendwie markieren. Das würde ich natürlich jetzt wieder mit dem Plus machen.

Klickt auf den Tier-hinzufügen-Button.

I: Fehler: Bitte wähle den Schwanz vom Tier aus, bevor du den Add-Mode deaktivierst. (Fig. C.5)

B: Ah, damit würde ich das deaktivieren. Okay, gut. Dann, ehm, ich habe ja jetzt keine



Figure C.5.: Proband B - Adding Crustacean

Möglichkeit... Außer ich gehe jetzt hier an die Krabbe, unten, wo der Schwanz sein sollte und klicke hier.

Klickt auf den Schwanz des Krustentiers.

B: Okay. Ich habe eine Krabbe markiert. Jetzt drücke ich hier auf Fisch, sage es ist eine Krabbe

und klicke Krabbe.

Wählt "Crustacea" im Group-Feld im Tierspezifikations-Feld aus. **B**: Jetzt habe ich die Aufgabe gelöst. Ja da oben ist natürlich noch, ich weiß jetzt nicht, ob ich den Fisch ändern muss, die Spezies ändern muss.

Klickt auf Fisch 2 (als Qualle markiert).

B: Ja, das ist doch keine Qualle! Das ist doch ein Fisch! Ich gehe hier drauf und sage das ist ein Fisch.

Wählt "Fish" im Group-Feld des Tierspezifikations-Felds aus.

B: Jetzt bin ich zufrieden. Ich habe drei Fische erkannt und einen Krebs.

Task Four

B: Also das Bild habe ich ja gerad bearbeitet, habe gesagt, wo welcher Fisch zu sehen ist, wo Schwanz und Kopf sind und dass ich einen Krebs erkannt habe. Das habe ich jetzt alles auf der linken Kamera gemacht. Jetzt klicke ich hier oben drauf.

Klickt auf den Bildwechsel-Button.

B: So, ich sehe jetzt, dass dieser Depp wieder zweimal eine Qualle erkannt hat, obwohl da gar keine ist. Es hat sich auch ein bisschen verändert. Ich bin mir jetzt nicht sicher, ob der kleine Fisch kleiner Fisch war und ob der große Fisch großer Fisch war. Ich müsste nochmal zurück gehen, um mir sicher zu sein, dass ich jetzt auch den richtigen Fisch markiere.

Klickt auf den Bildwechsel-Button (wechselt zur Ansicht mit beiden Bildern).

B: Jetzt sehe ich, im oberen Bild ist der große Fisch identisch mit dem großen Fisch auf der anderen Kamera. Und der Fisch, der nach oben weg schwimmt, ist identisch mit dem hier. Und der kleine Fisch im unteren Bild, der als Qualle markiert ist, ist identisch mit dem hier oben, der vorher auch als QUalle markiert war. Jetzt bin ich sicher, was ich zu tun habe. Hier sehe ich jetzt, dass ich beide Bildschirme gleichzeitig aufgerufen habe, das finde ich sehr gut. Ich gehe jetzt auf dieses Bild und klicke drauf.

Klickt auf Fisch 3 im rechten Bild. (Fig. C.6)

B: Also, dieses Tier ist jetzt markiert. Ich habe es als Fisch erkannt, aber der Computer sagt, es ist eine Qualle. Also gehe ich jetzt hier auf den Pfeil und wähle den blauen Fisch und klicke darauf.

Wählt "Fish" im Group-Feld des Tierspezifikations-Felds von Fisch 2 aus (rechtes Bild).

B: Das finde ich schon sehr gut. Jetzt hat er hier etwas erkannt, was gar nicht da ist. Ich gehe wieder in die Mitte zwischen Kopf und Schwanz ungefähr und klicke.

Klickt auf das false positive Tier (rechtes Bild).

B: Das ist gar nichts, ich möchte das löschen. Dann gehe ich oben zum Papierkorb und klicke darauf.

Klickt auf den Tier-löschen-Button.

B: Ja weg damit, da machst du nur den Papierkorb auf? Okay, hm, was mache ich denn da jetzt? Ich klicke das hier mal an und schiebe das hier oben in den Papierkorb rein.

Klickt auf den Kreis des Tiers und versucht diesen zum Mülleimer-Icon zu ziehen.

I: Das Anklicken hat das Tier schon gelöscht.

B: Ah, okay. Jetzt bin ich davon überzeugt, dass die linke und rechte Kamera die gleiche

C. Prototype



Figure C.6.: Proband B - Corrections on Right Image

Anzahl an Fischen und Krebsen haben. Damit habe ich die Aufgabe doch gelöst oder? I: Noch nicht ganz, es fehlt noch der Schritt, dass du dem Computer sagst, welcher Fisch links zu welchem rechts gehört.

B: Ah ja stimmt, die Zuordnung fehlt. Wie sage ich dem Computer, dass, der Fisch... Hovert über Fisch 1 (linkes Bild).

B: ...gleich der Fisch ...

Hovert über Fisch 1 (rechtes Bild). Klickt auf Fisch 1 (linkes Bild).

B: ...ist? Ich habe den Fisch angeklickt und wenn ich den hier unten anklicke.

Klickt auf Fisch 1 (rechtes Bild).

B: So, okay, jetzt ist also auf beiden Bildschirmen der Fisch markiert. Das ist schonmal viel Wert. So, ehm, da ist die Länge. Ich könnte vielleicht, um dem Computer mitzuteilen, dass es der gleiche Fisch ist, könnte ich unter Remark, also eine Bemerkung setzen, "Fisch 1". Also ich könnte hier auf "Remark" gehen.

Gibt "Fisch 1" im Remark-Feld des Tierspezifikations-Felds ein (linkes Bild).

B: Und das gleiche kann ich doch auch hier unten machen. Da sage ich auch "Fisch 1".

Gibt "Fisch 1" im Remark-Feld des Tierspezifikations-Felds ein (rechtes Bild).

B: Jetzt gehe ich auf den zweiten Fisch, klicke ihn an und ich klicke den Fisch hier unten an, den ich als zweiten Fisch haben möchte.

Klickt auf Fisch 2 (linkes Bild) und klickt auf Fisch 2 (rechtes Bild).

B: Jetzt gehe ich wieder zur "Remark" und mache daraus jetzt ein "Fisch 2".

Gibt "Fisch 2" im Remark-Feld des Tierspezifikations-Felds ein (linkes, dann rechtes Bild).

Klickt auf Fisch 3 (linkes, dann rechtes Bild). Gibt "Fisch 3" im Remark-Feld des Tierspezifikations-

Felds ein (linkes, dann rechtes Bild).

Klickt auf das Krustentier (linkes, dann rechtes Bild). Gibt "Crustacea 1" im Remark-Feld des Tierspezifikations-Felds ein (linkes, dann rechtes Bild).

B: Aufgabe erfolgreich erledigt.

C.4. Test Protocol Proband C

Task One

C: Dann klicke ich einfach mal hier drauf.

Klickt auf das MarOMarker Icon auf dem Desktop.

C: Ah, okay. Jetzt muss ich meine ID eintragen. Habe ich schon eine ID?

I: Das ist einfach nur deine normale ID, also XY.

C: Okay, das kann ich jetzt machen.

Gibt ID in das ID-Feld ein.

C: ID und dann... dann klicke ich jetzt auf "Browse".

Klickt auf den Browse-Button.

C: Dann sehe ich eine congig file. Kann ich das einfach öffnen oder muss ich das noch bearbeiten?

I: Du kannst es einfach so benutzen, es ist schon vorbereitet.

C: Dann klicke ich auf "Öffnen". Hakt das bei dir auch so?

Klickt auf "Öffnen". Klickt auf den Ok-Button.

I: Ja, heute ist es schlimm.

Task Two

C: Ich möchte jetzt das image laden. D.h. ich muss ja erstmal mein image finden. Dann klicke ich jetzt hier drauf, auf diese drei Balken.

Klickt auf das Menü.

C: Ich würde auf "Data" gehen und da jetzt meine Bilder finden?

Klickt auf "Data".

C: Okay, es hat sich nichts geändert. Ah jetzt.

(Übertragungsverzögerung). Ich kann jetzt ein Datum auswählen, ich möchte welches Datum nochmal auswählen?

I: Das war der 01.06.2019.

Wählt den 01.06.2019 im Kalender aus.

C: Ah, da ist schon mein image directory und ich habe auch schon meine result file. Verstehe. Ich würde nochmal nachgucken, ob das input directory und das result file richtig sind. Dann würde ich einmal "Browse" klicken.

Klickt auf den Browse-Button, um das Bildverzeichnis zu überprüfen.

C: Okay, hier steht "here you find all time-normized images that you need". Die sind also alle time-normized. Ich kann aber keine Datei auswählen.

I: Genau, hier kannst du nichts auswählen.

C: Okay, d.h. ich lasse das einfach so. Dann klicke ich "OK".

Klickt auf den Ok-Button.

C: Hmm, und dann gucke ich jetzt nochmal, ob das result file richtig ist.

Klickt auf den Browse-Button, um die Ergebnisdatei zu überprüfen.

C: Result 2019 null sechs hört sich richtig an.

Wählt result_2019_06.txt im Dateiexplorer aus.

C: Und dann klicke ich auf "OK".

Klickt auf den Ok-Button.

C: Gut. Image prefix, ich weiß jetzt nicht, was das ist... Brauche ich ein image prefix? **I**: Nein, das ist optional.

C: Okay, dann würde ich jetzt einfach auf "Run Neural Network" klicken.

Klickt auf den Button, um das neuronale Netz zu starten.

C: Okay, ein Bild bearbeitet.

Task Three

C: Kann ich da bei Filter-Options etwas auswählen? Unter dem Kalender.

I: Du kannst gerne ausprobieren.

C: Oho, wenn ich das jetzt auch mache...

Klickt auf die Combobox für die Datenfilter.

C: Ich würde einfach mal auf "All" gehen. Dann klicke ich auf "Correct predictions".

Wählt "Alle" in der Combobox für Datenfilter aus und klickt auf den Korrigiere-Vorhersagen-Button.

C: Links oben steht nur das Datum. Da steht nur 2017, aber ich sollte ja 2019 machen.

I: Da hast du recht, das ist mir gar nicht aufgefallen.

C: Okay, das heißt, ich habe das linke Bild angezeigt. Ein Fisch ist schon ausgewählt, ein anderer ist gelb. Den würde ich jetzt erstmal rüberziehen.

Zieht den Kreis von Fisch 1 an die korrekte Stelle.

C: Und dann kann ich die Art auswählen. Muss ich nicht, aber das mache ich.

Klickt auf das Species-Feld im Tierspezifikations-Feld.

C: Eine Makrele.

Wählt "Mackerel" aus.

C: Und dann noch einmal auf die Länge.

Klickt auf "Length".

I: Keine Aktion.

C: Okay, dann würde ich zum nächsten Fisch übergehen. Da ist noch ein Fisch der nicht erkannt wurde. Da kann ich vielleicht die zwei nehmen, die da einfach herum liegen (X und O), da wurde vermutlich eins falsch erkannt. Kann ich das einfach löschen oder kann ich das hier rüber ziehen?

Zieht das Kreuz des falsch erkannten Tiers zum Schwanz des unmarkierten Tiers (Fisch 3) und den Kreis zu seinem Kopf.

C: Da ist schon "Fisch" ausgewählt. Dann kann ich hier auch noch die Art auswählen. Das ist auch eine Makrele.

Wählt "Mackerel" im Species-Feld des Tierspezifikations-Felds aus.

C: Ich würde jetzt einmal kontrollieren, ob dieser Fisch wirklich ein Fisch ist. Klickt auf Fisch 2.

C: Also das ist falsch markiert als "Jellyfish". Dann klicke ich hier nochmal auf Gruppe auswählen und wähle "Fisch" aus.

Wählt "Fish" als Gruppe für Fisch 2 aus.

C: Das finde ich richtig gut mit den Farben. Ehm und dann würde ich hier nochmal nach der Krabbe gucken.

Klickt auf das unmarkierte Krustentier.

I: Keine Aktion.

C: Dann muss ich sie ja irgendwie noch auswählen. Hmm. Dann klicke ich hier oben auf die drei Balken.

Navigiert zur Data-Seite.

C: Ah okay, dann komme ich wieder hier hin. Dann würde ich wieder zurück gehen. Navigiert zur Home-Seite.

C: Hier oben ist ein Plus. Ist das ein Plus für das Hinzufügen von einem Tier? Klickt auf den Tier-hinzufügen-Button.

C: Okay und jetzt klicke ich auf die Krabbe.

Klickt auf das Krustentier.

C: Okay, ich brauche noch das Ende von der Krabbe. Ich klicke hier nochmal.

Klickt auf den Schwanz des Krustentiers.

C: Dann klicke ich jetzt nochmal auf Art. Ah okay, benthische Crustacea.

Wählt "Benthic Crustacea" im Species-Feld des Tierspezifikations-Felds aus.

C: Sehr gut. Dann habe ich alles ausgewählt und würde weiter zum nächsten Bild gehen.

Task Four

C: Das heißt, ich muss jetzt herausfinde, wie ich zum rechten Bild komme. Hier oben ist ja ein "L". Da würde ich erstmal drauf klicken.

Klickt auf den Bildwechsel-Button (wechselt so zum rechten Bild).

C: Ah jetzt bin ich im rechten Bild. Da ist schon ein Fisch ausgewählt. Das heißt, das Program hat den schon als Fisch erkannt. Ich kann jetzt noch die Art auswählen.

Wählt "Mackerel" im Species-Feld des Tierspezifikations-Felds aus. (Fig. C.7)

C: Jetzt würde ich den hier gerne zum Fisch auf dem anderen Bild vergleichen. Was passiert denn, wenn ich hier drauf klicke?

Klickt auf den Filter-Button.

I: Hier könntest du einen Filter für das Bild auswählen.

Klickt auf Fisch 1 und wählt "Mackerel" im Species-Feld des Tierspezifikations-Felds aus.

C: Was passiert denn jetzt, wenn ich jetzt darauf klicke?

Klickt auf "Length".

I: Keine Aktion.

C: Okay, hmm, dann bearbeite ich erstmal den Fisch, der falsch als "Jellyfish" markiert wurde. Klickt auf Fisch 3 und wählt "Fish" im Group-Feld des Tierspezifikations-Felds aus. Wählt



Figure C.7.: Proband C - Species Specification on Right Image

"Mackerel" im Species-Feld.

C: Und dann habe ich hier noch etwas Gelbes. Da sehe ich nichts. Also lösche ich das.

Klickt auf den Tier-löschen-Button.

C: Und dann klicke ich hier hin.

Klickt auf die false positive Markierung (wird gelöscht).

C: Dann wähle ich einmal die Krabbe aus.

Klickt auf das Krustentier.

C: Das stimmt so. Was ist denn das hier rechts unten in der Ecke?

Klickt auf den Button, um das Bild in einem separaten Fenster zu öffnen.

I: Aktion ist noch nicht implementiert. Das würde das Bild in einem neuen Fenster öffnen.

C: Achso. Hm. Jetzt würde ich die gerne miteinander vergleichen. Dann klicke ich mal auf das "R" drauf.

Klickt auf den Bildwechsel-Button (wechselt zur Ansicht mit beiden Bildern).

C: Ah, dann kann ich beide Bilder sehen. Okay, dann kann ich hier den gleichen auswählen? Also ich würde jetzt hier oben auch auf die Krabbe klicken, sodass auf beiden Bildern die Krabbe ausgewählt ist.

Klickt auf das Krustentier (linkes Bild). Auf dem rechten Bild wird Fisch 3 automatisch selektiert.

C: Hm, okay. Kann ich das hier unten bewegen? (bounding box) **I**: Nein.

C: Der hat mir die nur falsch zusammengeführt. Dann würde ich das gerne ändern. Ist nur die Frage, wie. Wenn ich auf die Krabbe unten klicke, was passiert dann?

Klickt auf das Krustentier (rechtes Bild).

C: Ah jetzt sind beide ausgewählt. Jetzt möchte ich gerne die Länge wissen.

I: Die wird gerade berechnet.

Die berechnete Länge ist negativ.

C: Okay. Bei der oberen Krabbe sind Kopf und Fuß vertauscht.

Tauscht Kopf- und Schwanzmarkierungen des Krustentiers (linkes Bild), sodass das Kreuz auf dem Kopf liegt und der Kreis auf dem Schwanz.

C: Dann stimmt die Krabbe jetzt. Dann klicke ich jetzt hier auf diesen kleinen Fisch. Klickt auf Fisch 3 (linkes Bild).

C: Und dann klicke ich auf den kleinen Fisch hier.

Klickt auf Fisch 3 (rechtes Bild).

C: Der ist 20cm, groß, das ist schön. Jetzt klicke ich auf den Fisch da oben.

Klickt auf Fisch 2 (linkes Bild).

C: Der hat auch 20cm. Gut, dann klicke ich auf den anderen Fisch. Klickt auf Fisch 2 (rechtes Bild).

C: Dann klicke ich auf den Fisch hier.

Klicjt auf Fisch 1 (linkes Bild, dann rechtes Bild). Die berechnete Länge ist negativ.

C: Der untere ist falsch markiert, also tausche ich die Markierungen wieder.

Tauscht Kopf- und Schwanzmarkierungen von Fisch 1 (rechtes Bild), sodass der Kreis auf dem Kopf und das Kreuz auf dem Schwanz liegt.

C: Kann ich jetzt bestätigen, dass ich fertig bin oder gehe ich einfach zum nächsten Bild? Ich würde einfach auf weiter klicken.

Klickt auf den Nächstes-Bild-Button.

D. Usability Questionnaire

2.

3.

4.

5.

1. Ich kann mir sehr gut vorstellen, das System regelmäßig zu nutzen.

	1	2	3	4	5	
Ich stimme nicht zu	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	Ich stimme zu
mpfinde das System	als un	nötig ko	omplex			
	1	2	3	4	5	
Ich stimme nicht zu	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	Ich stimme zu
mpfinde das System	als ein	fach zu 2	ı nutzer 3	ı. 4	5	
npfinde das System Ich stimme nicht zu	als ein	fach zu 2	a nutzer 3	1. 4	5	Ich stimme zu
mpfinde das System Ich stimme nicht zu enke, dass ich techn	als ein 1 	fach zu 2 Suppor 2	a nutzer 3 	n. 4 hen wü 4	5 irde, ur 5	lch stimme zu n das System z

Ich stimme nicht zu	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	Ich stimme zu

6. Ich finde, dass es im System zu viele Inkonsistenzen gibt.



7. Ich kann mir vorstellen, dass die meisten Leute das System schnell zu beherrschen lernen.

	1	2	3	4	5	
Ich stimme nicht zu	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	Ich stimme zu

8. Ich empfinde die Bedienung als sehr umständlich.

	1	2	3	4	5	
Ich stimme nicht zu	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	Ich stimme zu

9. Ich habe mich bei der Nutzung des Systems sehr sicher gefühlt.

	1	2	3	4	5	
Ich stimme nicht zu	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	Ich stimme zu

10. Ich musste eine Menge Dinge lernen, bevor ich mir dem System arbeiten konnte.



11. Ich bin der Meinung, dass eine Qualitätsverbesserung bei der Markierung und Vermessung der Tiere mithilfe der Anwendung erfolgt. (Nur falls du Erfahrung mit Macro Scheduler und/oder Stereo Marker hast).

 \bigcirc Ja

 \bigcirc Nein

12. Weil

13. Weitere Bemerkungen

E. Tables

	60 epochs	110 epochs
Minimal CNN	0.001529645	0.001722515
No layers	0.009095004	0.005477736
Last layers	0.001100117	0.001099558
All layer	0.000862427	0.000746301
All layers delayed	0.001002867	0.000744527

Table E.1.: Test MAE for the minimal CNN and the transfer learning variations

	TPs	TNs	FPs	FNs	Precision	Recall
Fish head	97	800302	283	86	25.53%	53.01%
Fish tail	88	800547	38	95	69.84%	48.09%
Crustacea head	107	800453	124	84	46.32%	56.02%
Crustacea tail	101	800466	111	90	47.64%	52.88%
Chaetognatha head	0	800757	0	11	0%	0%
Chaetognatha tail	0	800757	0	11	0%	0%
Jellyfish head	0	800758	0	9	0%	0%
Jellyfish tail	0	800759	0	9	0%	0%
Unidentified head	0	800763	0	5	0%	0%
Unidentified tail	0	800763	0	5	0%	0%

Table E.2.: Classification metrics for coordinate evaluation, unweighted model, 60 epochs

	TPs	TNs	FPs	FNs	Precision	Recall
Fish head	109	800420	165	74	39.78%	59.56%
Fish tail	109	800378	207	74	34.49%	59.56%
Crustacea head	110	800380	197	81	35.83%	57.59%
Crustacea tail	102	800417	160	89	38.93%	53.40%
Chaetognatha head	0	800757	0	11	0%	0%
Chaetognatha tail	0	800757	0	11	0%	0%
Jellyfish head	0	800759	0	9	0%	0%
Jellyfish tail	0	800759	0	9	0%	0%
Unidentified head	0	800763	0	5	0%	0%
Unidentified tail	0	800763	0	5	0%	0%

Table E.3.: Classification metrics for coordinate evaluation, unweighted model, 110 epochs

	TPs	TNs	FPs	FNs	Precision	Recall
Fish head	82	800479	106	101	43.62%	44.811%
Fish tail	0	800585	0	183	04%	0%
Crustacea head	66	800507	70	125	48.53%	34.55%
Crustacea tail	87	800474	103	104	45.79%	45.55%
Chaetognatha head	3	800712	45	8	6.25%	27.27%
Chaetognatha tail	3	800742	15	8	16.67%	27.27%
Jellyfish head	3	800754	5	6	37.50%	33.33%
Jellyfish tail	2	800749	10	7	16.67%	22.22%
Unidentified head	1	800751	12	4	7.69%	20.00%
Unidentified tail	1	800753	10	4	9.09%	20.00%

Table E.4.: Classification metrics for coordinate evaluation, weighted model, 60 epochs

	TPs	TNs	FPs	FNs	Precision	Recall
Fish head	95	800509	76	88	55.56%	51.91%
Fish tail	94	800509	76	89	55.29%	51.37%
Crustacea head	108	800454	123	83	46.75%	56.54%
Crustacea tail	87	800497	80	104	52.10%	45.55%
Chaetognatha head	3	800739	18	8	14.29%	27.27%
Chaetognatha tail	3	800737	20	8	13.04%	27.27%
Jellyfish head	3	800751	8	6	27.27%	33.33%
Jellyfish tail	2	800755	4	7	33.33%	22.22%
Unidentified head	0	800742	21	5	0%	0%
Unidentified tail	0	800741	22	5	0%	0%

Table E.5.: Classification metrics for coordinate evaluation, weighted model, 110 epochs

L. 100005	Ε.	Tables
-----------	----	--------

	TPs	TNs	FPs	FNs	Precision	Recall
Fish	76	800547	38	107	66.67%	41.53%
Crustacea	97	800473	104	94	48.26%	50.79%
Chaetognatha	0	800757	0	11	0%	0%
Jellyfish	0	800763	0	9	0%	0%
Unidentified	0	800759	0	5	0%	0%

Table E.6.: Classification metrics for closest match evaluation, unweighted model, 60 epochs

	TPs	TNs	FPs	FNs	Precision	Recall
Fish	91	800464	121	92	42.92%	49.73%
Crustacea	97	800422	155	94	38.49%	50.79%
Chaetognatha	0	800757	0	11	0%	0%
Jellyfish	0	800763	0	9	0%	0%
Unidentified	0	800759	0	5	0%	0%

Table E.7.: Classification metrics for closest match evaluation, unweighted model, 110 epochs

	TPs	TNs	FPs	FNs	Precision	Recall
Fish	0	800585	0	183	0%	0%
Crustacea	63	800508	69	128	47.73%	32.98%
Chaetognatha	1	800745	12	10	7.69%	9.09%
Jellyfish	1	800753	7	8	12.5%	11.11%
Unidentified	1	800752	10	4	9.09%	20.00%

Table E.8.: Classification metrics for closest match evaluation, weighted model, 60 epochs

	TPs	TNs	FPs	FNs	Precision	Recall
Fish	75	800526	59	108	55.97%	40.98%
Crustacea	87	800498	79	104	52.41%	45.55%
Chaetognatha	2	800745	12	9	14.29%	18.18%
Jellyfish	1	800743	5	8	16.67%	11.11%
Unidentified	0	800754	20	5	0%	0%

Table E.9.: Classification metrics for closest match evaluation, weighted model, 110 epochs

List of Figures

 2.1. 2.2. 2.3. 2.4. 2.5. 	Supervised Learning (adapted from [9])Perceptron (adapted from [10])VGG16 Architecture (based on [15])MobileNetV2 Architecture (adapted from [9])MobileNetV2 blocks	4 6 7 7
3.1.	GT encodings of test image 60 for fish (red points/blue crosses show head/tail	
	coordinates, vector fields only show non-zero vectors)	22
3.2.	Architecture of the maturalistic CNN	23
3.3. 2.1	Architecture of the networks for fish heads using transfer learning	24 26
3. 4 . 3.5.	Visualization of coordinate extraction from fish head heatmaps on test image 94 (white areas are heatmap, blue crosses are the GT coordinates, red points	20
	are the coordinates extracted from the heatmaps)	30
3.6.	Training MAE of minimal CNN and transfer learning variations	31
3.7.	Fish head heatmaps predicted by different 110 epochs models overlayed on test	
	image 39 (red points show GT head coordinates)	33
3.8.	MAE for all classes networks from selected training phases	34
3.9.	MAE for all classes networks in testing phase	35
3.10.	Fish head heatmaps predicted by all classes models overlayed on test image 51	
	(red points show GT head coordinates)	36
3.11.	Confusion matrices for networks with all classes. The background detections,	
	shown in the first row, are not counted and displayed as zero	38
3.12.	Provisional segmentation of fish class in test image 51 (red points are GT fish	
	heads, blue points are GT fish tails)	40
3.13.	Vector field predictions on test image 60 (red points show GT head coordinates,	10
	blue points show GT tail coordinates)	42
4.1.	Personas	53
4.2.	First slide of task two: empty home page	59
4.3.	Task three: correct positions and groups	60
4.4.	Task four: correct positions and groups (right image) and perform matching .	61
4.5.	Probands for prototype	62
4.6.	MarOMarker initial data page. Two functionality groups are highlighted by	
	thick red boxes: process guide (left) and initially disabled options (right)	66

List of Figures

4.7.	MarOMarker home page. Two functionality groups are highlighted by thick red boxes: general image manipulation options (left) and options to work
	within the image (right). Different animal groups have different colours 67
4.8.	Additional probands for the usability test
C_{1}	Drohand A Adding Crustaneon 114
C.1.	Froband A - Adding Crustacean
C.2.	Proband A - Removing Annotation
C.3.	Proband A - Matching Animals
C.4.	Proband B - Dragging Circle to Head 119
C.5.	Proband B - Adding Crustacean 120
C.6.	Proband B - Corrections on Right Image
C.7.	Proband C - Species Specification on Right Image

List of Tables

2.1.	Confusion matrix for binary classification (from [18])
3.1.	Test data statistics (rounded to two digits) 19
3.2.	Training data statistics (rounded to two digits)
3.3.	Validation data statistics (rounded to two digits)
3.4.	Class weights (rounded to two digits)
3.5.	Coordinate evaluation for network with all classes on test data 37
3.6.	Closest match evaluation for network with all classes on test data
4.1.	Keyboard shortcuts on home page
E.1.	Test MAE for the minimal CNN and the transfer learning variations 131
E.2.	Classification metrics for coordinate evaluation, unweighted model, 60 epochs 131
E.3.	Classification metrics for coordinate evaluation, unweighted model, 110 epochs 132
E.4.	Classification metrics for coordinate evaluation, weighted model, 60 epochs 132
E.5.	Classification metrics for coordinate evaluation, weighted model, 110 epochs . 132
E.6.	Classification metrics for closest match evaluation, unweighted model, 60 epochs133
E.7.	Classification metrics for closest match evaluation, unweighted model, 110 epochs133
E.8.	Classification metrics for closest match evaluation, weighted model, 60 epochs 133
E.9.	Classification metrics for closest match evaluation, weighted model, 110 epochs 133

Acronyms

- AWI Alfred-Wegener-Institute, Helmholtz-Centre for Polar and Marine Research. 1, 2, 14, 16, 17, 47, 48, 52, 65, 72
- BN Batch Normalization. 6, 21
- CNN Convolutional Neural Network. 1, 5, 14, 20, 21, 74
- COSYNA Coastal Observing System for Northern and Arctic Seas. 1, 17
- **FN** False Negative. 9, 35–37, 41
- **FP** False Positive. 9, 30, 34, 35, 37, 41, 45, 72, 74
- GT Ground Truth. 9, 21, 22, 27, 28, 30, 33, 35, 36, 40–42, 134
- GUI Graphical User Interface. 46, 47
- HCI Human-Computer Interaction. 11
- HTA Hierarchical Task Analysis. 11, 46–49, 55, 57, 65
- IRes Inverted Residual. 6, 24
- MAE Mean Absolute Error. 8, 9, 31, 32, 34, 35, 38-40
- mAP mean Average Precision. 14
- MarOMarker Marine Organism Marker. 2, 46, 47, 73
- MSE Mean Squared Error. 8, 9, 28, 72
- ReLU Rectified Linear Unit. 6, 21
- RemOS1 Remote Optical System 1. 17, 18
- SUS System Usability Scale. 46, 68–70, 72
- TA Task Analysis. 11, 13
- **TN** True Negative. 9, 35, 37, 41
- **TP** True Positive. 9, 30, 35–37, 41
- UCD User-Centred Design. 2, 3, 10, 13, 47
- **UI** User Interface. 1, 2, 12, 13, 47, 64, 74, 75

Bibliography

- R. Danovaro, L. Carugati, M. Berzano, A. E. Cahill, S. Carvalho, A. Chenuil, C. Corinaldesi, S. Cristina, R. David, A. Dell'Anno, N. Dzhembekova, E. Garcés, J. M. Gasol, P. Goela, J.-P. Féral, I. Ferrera, R. M. Forster, A. A. Kurekin, E. Rastelli, V. Marinova, P. I. Miller, S. Moncheva, A. Newton, J. K. Pearman, S. G. Pitois, A. Reñé, N. Rodríguez-Ezpeleta, V. Saggiomo, S. G. H. Simis, K. Stefanova, C. Wilson, M. Lo Martire, S. Greco, S. K. J. Cochrane, O. Mangoni, and A. Borja. "Implementing and Innovating Marine Monitoring Approaches for Assessing Marine Environmental Status". In: *Frontiers in Marine Science* 3 (2016).
- [2] J. Carstensen. "Need for monitoring and maintaining sustainable marine ecosystem services". In: *Frontiers in Marine Science* 1 (2014), p. 33.
- [3] COSYNA Coastal Observing System for Northern and Arctic Seas. URL: https://www.hzg. de/institutes_platforms/cosyna/ (visited on 10/13/2020).
- [4] H. Lu, Y. Li, Y. Zhang, M. Chen, S. Serikawa, and H. Kim. "Underwater Optical Image Processing: a Comprehensive Review". In: *Mobile Networks and Applications* 22.6 (Dec. 2017), pp. 1204–1211.
- [5] J. Walsh, N. O' Mahony, S. Campbell, A. Carvalho, L. Krpalkova, G. Velasco-Hernandez, S. Harapanahalli, and D. Riordan. "Deep Learning vs. Traditional Computer Vision". In: *Advances in Computer Vision*. Vol. 2. Springer, Cham, Apr. 2019.
- [6] T. Jasper. "Erkennen und Vermessen von Meereslebewesen auf Stereo-Kamerabildern mit einem neuonalen Netz". B.S. Thesis. Universität Bremen, 2019.
- [7] T. M. Mitchell. *Machine Learning*. 1st ed. USA: McGraw-Hill, Inc., 1997.
- [8] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. http://www.deeplearningbook. org. MIT Press, 2016.
- [9] U. Frese. *Anwendungen der Bildverarbeitung*. University Lecture. Universität Bremen. 2020.
- [10] C. C. Aggarwal. Neural Networks and Deep Learning. Springer, Cham, 2018.
- [11] Q. Yang, Y. Zhang, W. Dai, and S. J. Pan. *Transfer Learning*. Cambridge University Press, 2020.
- [12] L. Torrey and J. Shavlik. "Transfer Learning". In: *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods and Techniques*. IGI Global, 2009.
- [13] Keras. URL: https://keras.io/ (visited on 10/13/2020).
- [14] ImageNet. URL: http://image-net.org/ (visited on 10/13/2020).

- [15] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. 2014.
- [16] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen. "Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation". In: *CoRR* (2018).
- [17] J. Bhattacharjee. *Practical Machine Learning with Rust: Creating Intelligent Applications in Rust.* Berkeley, CA: Apress, 2020.
- [18] A. Tharwat. "Classification assessment methods". In: *Applied Computing and Informatics* (2018).
- [19] D. Deuff and M. Cosquer. "Introduction to the Methods Employed". In: *User-Centered Agile Method*. John Wiley & Sons, Ltd, 2013. Chap. 2, pp. 5–19.
- [20] International Standardization Organization. "Part 210: Human-centered design for interactive systems". In: *ISO 9241-210, Ergonomics of human-system interaction*. Mar. 2010.
- [21] C. Wilson. Interview Techniques for UX Practitioners. Boston: Morgan Kaufmann, 2014.
- [22] A. Cooper, R. Reimann, and D. Cronin. *About Face 3: The Essentials of Interaction Design*. Wiley Publishing, Inc., 2007.
- [23] F. E. Ritter, G. D. Baxter, and E. F. Churchill. "Methodology I: Task Analysis". In: Foundations for Designing User-Centered Systems: What System Designers Need to Know about People. London: Springer London, 2014, pp. 309–333.
- [24] K. McElroy. Prototyping for Designers. Sebastopol, CA: O'Reilly Media, Inc., 2017.
- [25] M. Richter and M. Flückiger. *Usability Engineering kompakt*. Springer Vieweg, Berlin, Heidelberg, 2013.
- [26] J. Nielsen. "Heuristic evaluation". In: Usability Inspection Methods. New York, NY: John Wiley & Sons, 1994.
- [27] B. Shneiderman, C. Plaisant, M. Cohen, S. Jacobs, and N. Elmqvist. *Designing the User Interface: Strategies for Effective Human-Computer Interaction.* 6th ed. Pearson, 2016.
- [28] D. Norman. *The Design of Everyday Things*. MIT Press, 1998.
- [29] A. Dix, J. Finlay, G. D. Abowd, and R. Beale. *Human-Computer Interaction*. Harlow, England: Pearson Education Limited, 2004.
- [30] J. Sauro and J. R. Lewis. "Standardized usability questionnaires". In: *Quantifying the User Experience*. 2nd ed. Camebridge, MA: Elsevier Inc., 2016.
- [31] M. Wehkamp and P. Fischer. "A practical guide to the use of consumer-level digital still cameras for precise stereogrammetric in situ assessments in aquatic environments". In: *Underwater Technology* 32.2 (2014).

- [32] M. J. Costello, Z. Basher, L. McLeod, I. Asaad, S. Claus, L. Vandepitte, M. Yasuhara, H. Gislason, M. Edwards, W. Appeltans, H. Enevoldsen, G. J. Edgar, P. Miloslavich, S. De Monte, I. S. Pinto, D. Obura, and A. E. Bates. "Methods for the Study of Marine Biodiversity". In: *The GEO Handbook on Biodiversity Observation Networks*. Springer International Publishing, 2017, pp. 129–163.
- [33] X. Li, M. Shang, H. Qin, and L. Chen. "Fast accurate fish detection and recognition of underwater images with Fast R-CNN". In: OCEANS 2015 - MTS/IEEE Washington. IEEE, 2015.
- [34] S. Villon, M. Chaumont, G. Subsol, S. Villéger, T. Claverie, and D. Mouillot. "Coral Reef Fish Detection and Recognition in Underwater Videos by Supervised Machine Learning: Comparison Between Deep Learning and HOG+SVM Methods". In: Advanced Concepts for Intelligent Vision Systems. Springer International Publishing, 2016, pp. 160–171.
- [35] P. Fischer, A. Weber, G. Heine, and H. Weber. "Habitat structure and fish: assessing the role of habitat complexity for fish using a small, semi-portable, 3D underwater observatory." In: *Limnology and Oceanography: Methods* 5 (2007), pp. 250–262.
- [36] W. Luo, Y. Li, R. Urtasun, and R. Zemel. "Understanding the Effective Receptive Field in Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc., 2016, pp. 4898–4906.
- [37] A. Ali, S. M. Shamsuddin, and A. Ralescu. "Classification with class imbalance problem: A review". In: *SOCO*. 2015.
- [38] Anaconda. URL: https://www.anaconda.com/ (visited on 10/31/2020).
- [39] *PyQt*. URL: https://riverbankcomputing.com/software/pyqt/ (visited on 10/31/2020).
- [40] *Macro Scheduler*. URL: https://www.mjtnet.com/macro-scheduler.htm?gclid= EAIaIQobChMIqr7I-azS7AIVZbR3Ch1SFQSFEAAYASAAEgLy_D_BwE (visited on 10/31/2020).
- [41] M. Wehkamp and P. Fischer. *A stereo-image analysis JAVA software with links to executable and source code*. PANGAEA. 2014.
- [42] USE Together. URL: https://www.use-together.com/ (visited on 10/13/2020).
- [43] J. Brooke. SUS: A quick and dirty usability scale. 1996.
- [44] J. Sauro. A Practical Guide to the System Usability Scale: Background, Benchmarks & Best Practices. CreateSpace Independent Publishing Platform, 2011.
- [45] R. Malaka. Mensch-Technik-Interaktion. University Lecture. Universität Bremen. 2019.
- [46] Google Forms. URL: https://www.google.com/forms/about/ (visited on 11/01/2020).
- [47] J. R. Lewis. "The System Usability Scale: Past, Present, and Future". In: *International Journal of Human–Computer Interaction* 34.7 (2018), pp. 577–590.
- [48] J. M. Johnson and T. M. Khoshgoftaar. "Survey on deep learning with class imbalance". In: *Journal of Big Data* 6.1 (Mar. 2019).
- [49] R. Girshick, J. Donahue, T. Darrell, and J. Malik. *Rich feature hierarchies for accurate object detection and semantic segmentation*. 2014.