

# Semi-Automatic Image Annotation Tool

**Tilda Hylander and Miranda Alvenkrona**

Master of Science Thesis in Media Technology  
**Semi-Automatic Image Annotation Tool**  
Tilda Hylander and Miranda Alvenkrona  
LiTH-ISY-EX-23/5598-SE

Supervisor: **Magnus Malmström**  
ISY, Linköpings universitet

Examiner: **Fredrik Gustafsson**  
ISY, Linköpings universitet

*Division of Automatic Control  
Department of Electrical Engineering  
Linköping University  
SE-581 83 Linköping, Sweden*

Copyright © 2023 Tilda Hylander and Miranda Alvenkrona

## Abstract

Annotation is essential in machine learning. Building an accurate object detection model requires a large, diverse dataset, which poses challenges due to the time-consuming nature of manual annotation. This thesis was made in collaboration with Project Ngulia, which aims at developing technical solutions to protect and monitor wild animals. A contribution of this work was to integrate an efficient semi-automatic image annotation tool within the Ngulia system, with the aim of streamlining the annotation process and improving the employed object detection models. Through research into available annotation tools, a custom tool was deemed the most cost-effective and flexible option. It utilizes object detection model predictions as annotation suggestions, improving the efficiency of the annotation process. The efficiency was evaluated through a user test, with participants achieving an average reduction of approximately 2 seconds in annotation speed when utilizing suggestions. This reduction was supported as statistically significant through a one-way ANOVA test.

Additionally, it was investigated which images should be prioritized for annotation in order to obtain the the most accurate predictions. Different sampling methods were investigated and compared. The performance of the obtained models remained relatively consistent, although with the even distribution method at top. This indicate that the choice of sampling method may not substantially impact the accuracy of the model, as the performance of the methods was relatively comparable. Moreover, different methods of selecting training data in the re-training process was compared. The different in performance were considerably small, likely due to the limited and balanced data pool. The experiments did however indicate that incorporating previously seen data with unseen data could be beneficial, and that a reduced dataset can be sufficient. However, further investigation is required to fully understand the extent of these benefits.



## Acknowledgments

We would like to express our gratitude and appreciation to our supervisor Magnus Malmström, and examiner Fredrik Gustafsson for their helpful advise and valuable input. A special thanks to Martin Stenmarck for appreciated advise and helping with the technical related matters. Thanks to the people at HiQ for lending us an office workplace in Norrköping. And a big thanks to the other thesis students that have contributed to the work of Project Ngulia.

*Norrköping, June 2023  
Tilda Hylander and Miranda Alvenkrona*



---

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.2	Motivation . . . . .	2
1.3	Aim . . . . .	2
1.4	Research questions . . . . .	2
1.5	Delimitations . . . . .	3
<b>2</b>	<b>Theory</b>	<b>5</b>
2.1	Image Annotation . . . . .	5
2.2	Image Annotation in Object Detection . . . . .	5
2.3	Semi-Automatic Image Annotation . . . . .	6
2.4	Iterative Bounding Box Annotation Tool . . . . .	6
2.5	Active Machine Learning . . . . .	6
2.5.1	Selective Annotation . . . . .	7
2.5.2	Uncertainty Sampling . . . . .	8
2.5.3	Representative Sampling . . . . .	10
2.6	Evaluation . . . . .	10
2.6.1	Object Detection Model Evaluation . . . . .	11
2.6.2	ANOVA Test . . . . .	12
<b>3</b>	<b>User Interface</b>	<b>15</b>
3.1	Research . . . . .	15
3.1.1	Other Annotation Tools . . . . .	15
3.1.2	Implementing a Custom Annotation Tool . . . . .	16
3.1.3	Target Audience . . . . .	17
3.2	Prototype . . . . .	17
3.3	First Version . . . . .	18
3.4	Dataset . . . . .	18
3.4.1	Base model . . . . .	19
3.5	User Test . . . . .	19
3.5.1	Test Procedure . . . . .	19
3.5.2	Image Data And Collected Meta Data . . . . .	19
3.5.3	Test Steps . . . . .	20

3.5.4	Results . . . . .	20
3.5.5	Observations . . . . .	22
3.5.6	Survey . . . . .	23
3.6	Analysis . . . . .	25
3.6.1	Effect map . . . . .	25
3.6.2	Changes . . . . .	26
<b>4</b>	<b>Implementation</b>	<b>29</b>
4.1	System Overview . . . . .	29
4.2	Front End . . . . .	30
4.2.1	Overview . . . . .	30
4.2.2	Header . . . . .	32
4.2.3	Class List . . . . .	33
4.2.4	Instance List . . . . .	34
4.2.5	Toolbar . . . . .	34
4.2.6	Shortcuts . . . . .	35
4.2.7	Canvas . . . . .	36
4.3	Database . . . . .	38
4.4	Server . . . . .	42
<b>5</b>	<b>Active Learning</b>	<b>43</b>
5.1	Data Pool . . . . .	43
5.2	Selective Annotation . . . . .	43
5.2.1	Image Pool . . . . .	44
5.2.2	Uncertainty Sampling . . . . .	44
5.2.3	Representative Sampling . . . . .	45
5.2.4	Other Samplings . . . . .	46
5.2.5	Results . . . . .	46
5.3	Re-training of the Model . . . . .	49
5.3.1	Image Pool . . . . .	49
5.3.2	Unseen Data . . . . .	49
5.3.3	Maximizing Data: Selecting All Unseen and Seen Data . . .	49
5.3.4	Maintaining Data Balance: Achieving an Even Balance of Seen and Randomly Selected Unseen Data . . . . .	50
5.3.5	Preserving Class Balance: Achieving an Even Distribution of Unseen and Seen Data . . . . .	50
5.3.6	Results . . . . .	51
<b>6</b>	<b>Discussion</b>	<b>53</b>
6.1	Results . . . . .	53
6.1.1	User Test . . . . .	53
6.1.2	Selective Annotation . . . . .	55
6.1.3	Re-training of the Model . . . . .	55
6.2	Method . . . . .	56
6.2.1	Dataset . . . . .	56
6.2.2	User Test . . . . .	56



---

6.2.3	Selective Annotation . . . . .	57
6.2.4	Re-training of the Model . . . . .	57
6.3	The work in a wider context . . . . .	57
<b>7</b>	<b>Conclusion</b>	<b>59</b>
7.1	Research questions . . . . .	59
7.2	Future work . . . . .	60
7.2.1	Annotation tool . . . . .	60
7.2.2	Selective Annotation and Re-training . . . . .	61
<b>A</b>	<b>User Test Questions</b>	<b>63</b>
<b>B</b>	<b>User Test Annotation Times</b>	<b>65</b>
	<b>Bibliography</b>	<b>69</b>



# 1

---

## Introduction

In the field of machine learning, annotation is the process of labeling data. Image annotation is performed by labeling an image to show the data features that you want your model to recognize and learn. A common use in computer vision is object detection, where the model learns to detect and classify objects in an image. The label is often in the form of a bounding box that bounds an object, thus displaying the position and category of the detected object.

In machine learning (ML), training a good model starts from the data that is fed into it. A huge amount of annotated data is the key to successfully train a machine learning model. Annotation is a very repetitive task when done manually. By automating parts of the process, the average annotation time would likely be reduced. Image labeling can be performed by fully automatic algorithms; however this often leads to label noise, errors, and missing labels. To produce quality annotations the labeling is often performed manually, which is highly time consuming. A compromise between the two would be to introduce a semi-automatic annotation tool where a model can pre-annotate an image and then manual correction can be performed by a human annotator.

### 1.1 Background

A semi-automatic annotation tool uses an object detection model to predict the placement of the bounding boxes in addition to the corresponding class labels in an image. Active learning is achieved by re-training with the newly annotated samples added to the training data, to improve the accuracy of the predictions.

In the literature, multiple creations of semi-automatic annotation tools have been made. The tools proposed include computer vision and machine learning methods that support humans to produce more efficient annotations, while some promote the use of crowd sourcing to divide the workload and improve the qual-

ity of the annotations. Our contribution is to employ an efficient annotation tool for iterative refinement of the existing object detection models used in the Ngulia Project, which aims to protect wild animals. It will be investigated which images should be annotated to produce the most accurate predictions, and also how the new annotated images should be included in the training data in re-training process.

## 1.2 Motivation

The black rhino is a rhinoceros species that is still considered critically endangered today. Ngulia is a rhino sanctuary in Kenya which aims to protect the rhinos against poaching and black-market trafficking of rhino horns. These wildlife crimes continue to threaten the survival of the species. Project Ngulia is a collaboration between different companies and organizations, among them are Linköping University, HiQ, Kolmården Zoo, and Kenya Wildlife Service. The aim is to develop technical solutions to protect and monitor wild animals, in particular the rhinos in Ngulia sanctuary to help the park rangers.

From earlier work within the Ngulia Project [1–3] a large collection of camera trap images of big animals has been gathered. In order to use these images to improve the already existing machine learning models for object tracking and object detection in the project, annotation is required.

The thesis will focus on implementing an efficient semi-automatic annotation tool for iterative refinement of the existing object detection model. Annotated images are important in the process of continuous improvement of a machine learning model by introducing a larger set of training data. The tool should provide ML-assisted annotation suggestions as well as manual correction of image annotations.

## 1.3 Aim

This thesis aims to integrate an efficient semi-automatic image annotation tool within the Ngulia system. The annotated images should be integrated with existing training data of the model, thus allowing iterative learning. It will be investigated which images should be annotated to produce the most accurate predictions, and also how the new annotated images should be included in the training data in re-training process.

## 1.4 Research questions

The following research questions will be answered in this thesis:

- Is ML-supported annotation with suggestions more efficient than manual annotation in terms of time spent on each annotation? If so, what is the extent of this efficiency gain?

- What is the impact of different sampling methods, such as even distribution and prioritizing uncertain images, on prediction accuracy when selecting a limited number of annotated images?
- How should newly annotated images be included in model re-training? Should the focus be on maintaining an equal proportion of seen and unseen images, preserving class balance, solely using unseen data, or simply maximizing data by using all available data?

## 1.5 Delimitations

This thesis will use an already existing object detection model [1] for the ML-assisted suggestions in the annotation tool. The focus will lie on the construction of the tool rather than changing the model. The improvement of the model will solely come from producing more annotated training data and contentious training the model on that data. The dataset provided for testing and experiments contains the Swedish carnivores, but the annotation tool is intended to be used for the Ngulia Project.



# 2

---

## Theory

This chapter is divided into sections that cover different parts of the theory related to this thesis: image annotation, some methods to automate the annotation process, active learning, different methods for selective annotation, evaluation for object detection models, and how to test for statistical significance.

### 2.1 Image Annotation

In machine learning, image annotation is the process of labeling images in order to communicate to the model what features to recognize and associate with certain classes. The images can then be used to train a model using supervised learning. Supervised learning is defined by its use of labeled datasets to train models to classify data correctly. Once the model is trained, it should be able to identify those features in new unlabeled images and classify them correctly [4].

Image features are mostly low-level since they are extracted directly from signal information of image data. In contrast, the human cognitive perception of an image is based on high-level concepts that are obtained from those low-level features. Automatic concept recognition from visual features of images is challenging due to the semantic gap that exist between low level visual features and high level concepts [5].

### 2.2 Image Annotation in Object Detection

One challenge in supervised object detection is collecting large, high-quality labeled datasets. This since the performance of supervised machine learning models relies heavily on the amount and quality of annotated training data. The repetitive and hard work of manual annotation for larger datasets is often solved

by crowdsourcing, but this is not always a feasible option. For example, when the dataset is small, when the data is confidential in nature, or when annotation resources are limited. It is described to be a demand for “resource-efficient, user-friendly annotation tool” to assemble labelled training data. [6]. In the literature, there are different approaches to visual object annotation, however bounding box annotation is by far the most common task in practical and industrial applications [7].

## 2.3 Semi-Automatic Image Annotation

A large amount of labelled training data is the key to successfully train a machine learning model. Labelling data in a manual manor is tedious and time consuming. Semi-automatic annotation tools aim to relieve the user from this burden of manual annotation as much as possible. The literature gives examples of semi-automatic annotation approaches that aims to speed up the annotation process, using automatic generation of annotation proposals. There are many examples of such tools, for example V7 [8], SuperAnnotate [9], LabelBox [10], and MakeSense [11].

The semi-automatic approach is usually divided into a two-stage process. First a trained classification model performs a preliminary (but possible incorrect) annotation of the images, then a human annotator reviews the annotation produced by the automated annotation process and performs manual corrections to retain quality annotations [12].

## 2.4 Iterative Bounding Box Annotation Tool

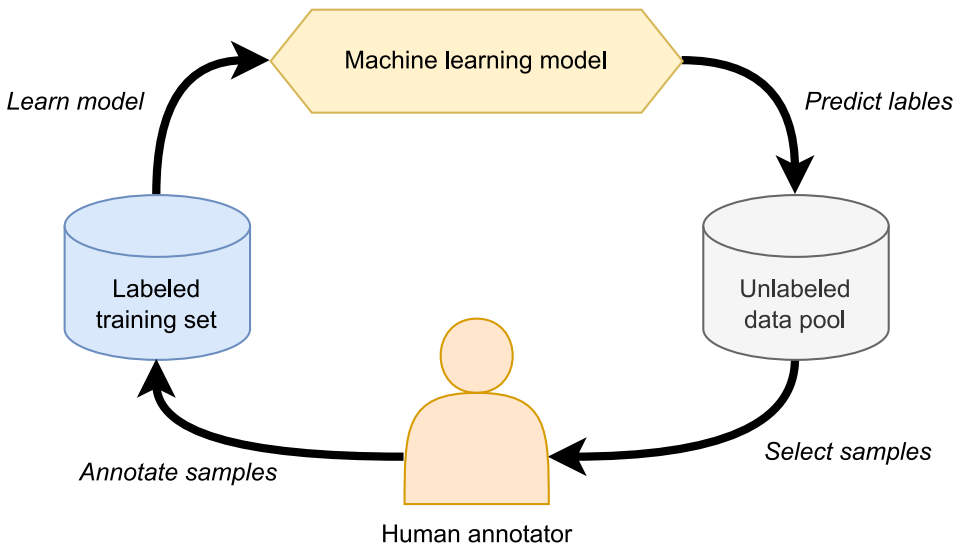
In the paper *Iterative Bounding Box Annotation for Object Detection* [6] an iterative annotation approach is presented, which takes advantage of a trained classification model to pre-annotate a batch of unlabeled images, leaving the annotator only for correction work. Their study also investigates different strategies for determining the order in which the images are presented to the annotator. Another study is conducted in the paper *Faster Bounding Box Annotation for Object Detection in Indoor Scenes* [7], where the tedious work of annotation is divided into a two-stage process. This process involves an initial stage of manual annotation, where the training data is used to train an object detector. In the second stage, the trained object detector is employed to generate proposal annotations, which are refined and corrected by a human annotator.

## 2.5 Active Machine Learning

Active machine learning is an iterative approach in machine learning that improves a model’s performance by determining the optimal data for human annotators to annotate. The goal of active machine learning is to enhance the efficiency of the annotation process by prioritizing which data to be annotated.



There are several studies in natural language processing using active learning that have shown a reduce of effort in the annotation process [13–16]. Active learning has also been used in image classification tasks, where Support Vector Machines (SVMs) have been utilized as a sampling method which have proven to reduce manual annotation effort [17–19]. The active learning process is illustrated in Figure 2.1. Active machine learning begins by annotating a smaller set of data to obtain an initial version of the model. The trained model can be utilized to predict the labels of additional samples, which can be used to prioritize useful samples for annotation. The annotated data is accumulated over time and utilized to retrain the model. This process continues, enabling the model to continually improve its performance through iterative training and annotation[20].



**Figure 2.1:** The active learning process.

### 2.5.1 Selective Annotation

Selective annotation is a technique utilized in active machine learning to reduce the amount of data required to be annotated. The technique involves utilizing an algorithm to identify the most informative instances from an unlabeled dataset and annotating only those selected instances. The primary idea behind retraining the model with the most informative instances is that it will likely improve the model's performance. The most informative instances refer to those that the model is most uncertain about or those that best represent the data distribution [21]. Several methods exist that can be used to select the most informative instances, including uncertainty sampling, representative sampling, diversity sampling and other techniques. These methods help to optimize the selection process, reduce labeling costs, and improve model performance, leading to a more efficient retraining of a model.

### 2.5.2 Uncertainty Sampling

Uncertainty sampling is one method to select instances from an unlabeled dataset. There are several types of uncertainty sampling, including least confidence, margin of confidence, ratio of confidence, and entropy. In this thesis, the methods of least confidence and entropy are implemented and evaluated.

The uncertainty sampling methods are centered on identifying instances where the model exhibits lower confidence, expecting it to lead to more accurate predictions after retraining. However, there are some adverse aspects to beware of when utilizing these sampling methods. The methods are reliant on the accuracy of the model's predictions, which could be problematic if the model is overly confident in an incorrect prediction. This may result in the exclusion of instances that contain the valuable information. Thus, it is crucial to ensure that the model is reasonably accurate before implementing these sampling methods.

#### Least Confidence

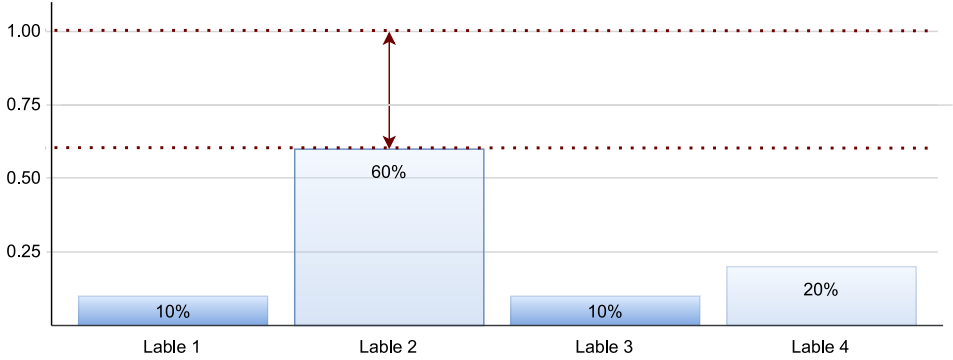
One of the simplest strategies of uncertainty sampling is least confidence. The strategy aims to select instances with the lowest confidence, as these instances are likely to be the most informative. For a probability distribution over a set of labels  $y$  for the item  $x$ , the confidence score denoted as  $\phi_{LC}(x)$  is calculated according to (2.1) with the probability of the highest confidence of the label given as  $\hat{y}$ . This measurement is illustrated in Figure 2.2.

$$\phi_{LC}(x) = 1 - P(\hat{y}|x) \quad (2.1)$$

By selecting the instances the model is least certain about, it can provide valuable information for improving the model's performance. The confident score can be normalized according to (2.2) for easier detection, where  $n$  is number of labels. When normalized the score is in a 0-1 range where 0 is the most certain score and 1 is the most uncertain.

$$\phi_{LC}(x) = (1 - P(\hat{y}|x)) * \frac{n}{n - 1} \quad (2.2)$$

There is a risk that the least confident instances are more ambiguous rather than informative. The method is also only sensitive to the predicted label, and do not consider uncertainty between the other labels. This can lead to problems where the model has difficulties distinguishing between similar labels. Another negative consequence that can arise is that there is a bias risk which can result in limited diversity. There is a possibility that too many similar instances from the same region in the feature space are selected, resulting in lack of diversity and can lead to overfitting [22].



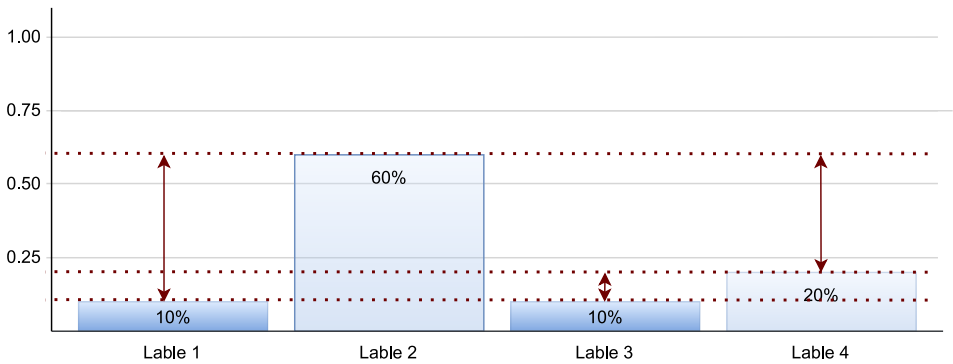
**Figure 2.2:** Least confidence score is the measure between the most confident prediction and 1.

## Entropy

Entropy, as a method of uncertainty sampling, is a measure of the amount of uncertainty within the probability distribution over a set of labels, see Figure 2.3. The entropy is calculated according to (2.3) and the normalized entropy as (2.4). Similar to the normalization of least confidence, the entropy is also normalized for the same underlying reasons.

$$\phi_{ENT}(x) = - \sum_y P(y|x) \log_2 P(y|x) \quad (2.3)$$

$$\phi_{ENT}(x) = \frac{- \sum_y P(y|x) \log_2 P(y|x)}{\log_2(n)} \quad (2.4)$$



**Figure 2.3:** Entropy is the measurement between all predictions.

The entropy sampling selects instances where the model indicates the highest level of uncertainty regarding the labels. When the model's predictions are evenly distributed among different labels, as illustrated in Figure 2.4, the entropy

value is high. Conversely, when the model’s predictions are concentrated on one or a few specific labels, as illustrated in Figure 2.5, the entropy is low.

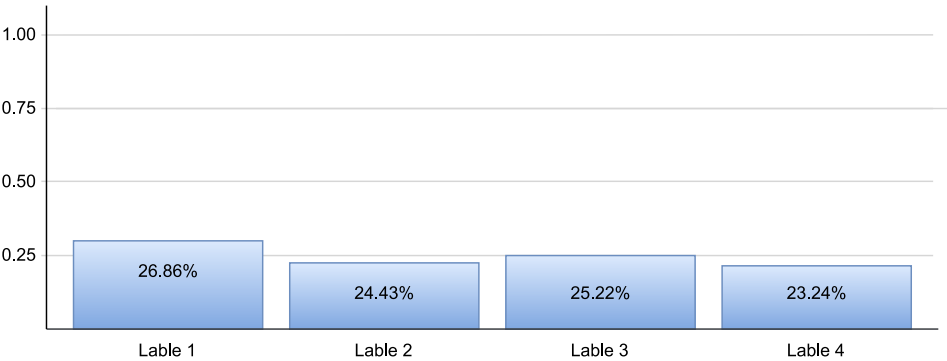


Figure 2.4: Example of high entropy.

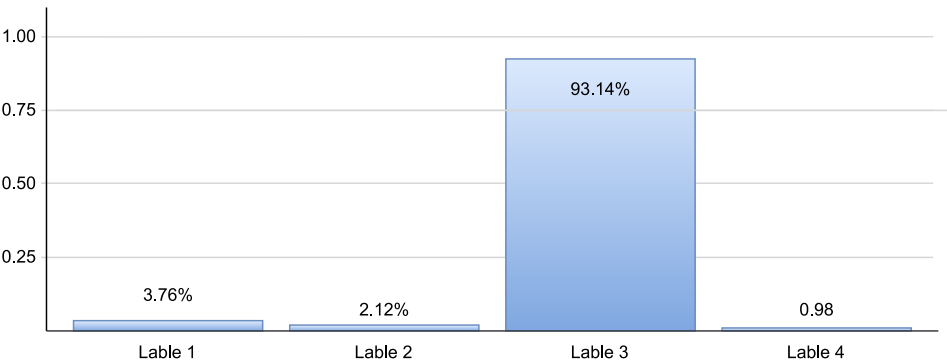


Figure 2.5: Example of low entropy.

2.5.3 Representative Sampling

Representative sampling is a method used to select a subset of data that accurately represents the entire dataset. It achieves this by categorizing the data based on a specific criterion, such as the distribution across classes or similarity to other data points in the dataset [23].

2.6 Evaluation

The retrained object detection models in this study will be evaluated using precision, recall, and F1-score using an intersection over union (IoU) threshold. The analysis of variance (ANOVA) test is used to assess the statistical significance between means of different groups.

### 2.6.1 Object Detection Model Evaluation

In addition to commonly used metrics such as precision, recall, and F1-score, object detection tasks in computer vision benefit from additional metrics such as IoU for evaluating predictions using bounding boxes.

#### Intersection over Union

The IoU is used as a threshold for determining whether a predicted outcome is a true positive or a false positive. The IoU is computed by dividing the intersecting area of the predicted bounding box and the ground truth bounding box by the total combined area encompassed by both bounding boxes (e.g. an IoU of 0.5 means that the areas are overlapping with 50%).

#### Precision

Precision measures the proportion of correctly predicted bounding boxes (true positives) out of all predicted bounding boxes, and ranges for 0 to 1. Precision is typically calculated by considering a specific IoU threshold. A predicted bounding box is considered a true positive if the IoU between the prediction and the ground truth bounding box exceeds the threshold. Precision is calculated according to

$$Precision = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Positives\ (FP)} \quad (2.5)$$

where TP represents the number of correctly predicted bounding boxes and FP represents the number of incorrectly predicted bounding boxes. A high precision indicates the model's ability to minimize false positives.

#### Recall

Recall measures the proportion of the correctly predicted bounding boxes (true positives) out of all the ground truth bounding boxes, and ranges for 0 to 1. Recall is also typically calculated using a specific IoU threshold, where a predicted bounding box is considered a true positive if it exceeds the threshold. Recall is calculated according to

$$Recall = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Negatives\ (FN)} \quad (2.6)$$

where TP represents the number of correctly predicted bounding boxes and FN represents the number of missed ground truth bounding boxes. High recall indicates a model's effectiveness in detecting most of the positive instances.

#### F1-score

The F1-score is a combined metric that considers both precision and recall. It provides a single value to assess the overall performance of the object detection

model, where the maximum value is 1 and the minimum value is 0. A high F1-score indicates high precision and high recall. The F1-score is commonly calculated as the harmonic mean of precision and recall according to 2.7, using the same IoU threshold.

$$F\text{-score} = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (2.7)$$

## 2.6.2 ANOVA Test

The one-way ANOVA is a statistical method used to evaluate whether there are statistically significant differences between the means of two or more independent (unrelated) groups. It is also possible to conduct a T-test when comparing only two groups. The main purpose of ANOVA is to determine if the means of the compared groups differ significantly from one another, or if any observed differences are simply due to random chance [24]. ANOVA uses a null hypothesis  $H_0$  (the means of the two groups are equal) and an alternative hypothesis  $H_a$  (the means of the two group are not equal). F-test Statistics (F-value) and Probability Value (P-value) are statistical measures used to determine the significance of the differences observed between the means of compared groups. In combination, they provide insights into whether the null hypothesis should be accepted or rejected.

### F-test Statistics

ANOVA uses the F-test statistic, which measures the ratio of the between-group variability to the within-group variability. The F-test statistic is calculated according to (2.8) where  $MS_{bg}$  is the between-group mean square and  $MS_{wg}$  is the within-group mean square.

$$F\text{-value} = \frac{MS_{bg}}{MS_{wg}} \quad (2.8)$$

The calculated F-value is compared to the critical F-value obtained from a statistical table. The critical F-value is retrieved from the table by calculating the degrees of freedom between groups (DFB) (2.9) and the degrees of freedom within groups (DFW) (2.10), where  $k$  is the number of groups and  $N$  is the number of samples [25].

$$DFB = k - 1 \quad (2.9)$$

$$DFW = N - k \quad (2.10)$$

If the calculated F-value exceeds the critical F-value, the null hypothesis is rejected, indicating a significant difference between the means of the two groups. Conversely, if the calculated F-value does not exceed the critical F-value, the null hypothesis is not rejected, indicating insufficient evidence to suggest a significant difference between the means.

**Probability Value**

The probability value (P-value) is a measure of the evidence against the null hypothesis. It indicates the probability of obtaining the observed F-value, or a more extreme value, assuming that the null hypothesis is true. Essentially, the P-value provides insight into how likely the data would be if there were no actual differences between the means of the groups. A P-value below the significance level rejects the null hypothesis, indicating significant differences between group means. A P-value above the significance level fails to reject the null hypothesis, suggesting that any observed differences between the group means could be due to random chance.





# 3

---

## User Interface

This chapter describes the process of developing and evaluating the annotation tool interface. The process includes initial research, prototype creation, user testing, and evaluation of the annotation efficiency.

### 3.1 Research

During the research phase, different annotation tools available on the market were tested. The key aspects considered were the suitability of the tools for the specific purpose of the study and the potential for workflow optimization. Additionally, the benefits and drawbacks of developing an annotation tool from scratch were carefully evaluated.

#### 3.1.1 Other Annotation Tools

The annotation tools that were considered was V7 labs [8], Superannotate [9], Labelbox [10], and Make Sense [11]. The idea was to try out these tools to get a grip of the workflow and experienced efficiency, but only the Make Sense tool was available free of charge. Therefore, the analysis of the tools was conducted by reviewing available resources, including videos on YouTube, relevant documentation, and the tools' official websites.

The aspects that were analyzed was mainly price and collaboration opportunities, which is illustrated in Table 3.1. Another aspect considered was if the tools provide ML-assisted labeling, which all the tools offered to some extent.

Tool	Price	Allows collaboration
V7 Labs	Undisclosed	Yes
Superannotate	Starts at \$6000 annually	Yes
Labelbox	Undisclosed	Yes
Make Sense	Free	No

*Table 3.1: Summary of the Annotations Tools.*

### 3.1.2 Implementing a Custom Annotation Tool

Creating an annotation tool from scratch would provide larger flexibility and could be used entirely free of charge for the Ngulia Project. This would also allow integration with the existing Ngulia system which would give direct access to images taken by cameras. These images are already classified by object detection models running on a server. Utilizing these classifications would remove the need for running a separate model for annotation suggestions. A custom tool also allows for a custom user management system, where users can be provided with different privileges, thus some functionality can be restricted to higher level users. Additionally, a tool from scratch can be customized to meet specific needs and preferences of a target audience.

After investigating the option of creating a custom annotation tool as well as checking out available options on the market, it was decided to create one from scratch. Opting to develop an annotation tool from scratch presented a more cost-effective solution, in addition to offering advantages such as enhanced flexibility. A custom tool can also allow for optimization of functionality and design with intuitiveness and efficiency as a central focus. By exploring other available annotation tools, it was possible to gain valuable insights regarding desirable functionality and design. The desirable functionality observed in the different tools encompasses the ability to manipulate images within the canvas, such as zooming and panning, to obtain a beneficial view. Additionally, the illustration of vertical and horizontal help lines for the mouse cursor, aids in achieving greater accuracy when drawing bounding boxes. The ability to adjust the size of the bounding box using markers positioned on the sides and corners is also advantageous.

A list of desirable keyboard shortcuts observed in the tools is presented below:

- Use the keys 1,2,3...,9 to change object class.
- Use the keys A, D or W, S to switch between the object classes.
- Press key 0 to reset the zoom, thus returning the image to its original size.
- Mouse scroll to zoom.
- Use the keys W, A, S, and D to move around in the image. Press W and S to go up and down, and press A and D to go left and right.
- Press the key . (dot) to continue to the next image.
- Press the key , (comma) to go back to the previous image.
- Press the key B to activate bounding box mode or click on symbol in the toolbar. To draw a bounding box, simply click, hold, drag, and then release.
- Use keys ctrl + C and keys ctrl + V to copy and paste bounding boxes.

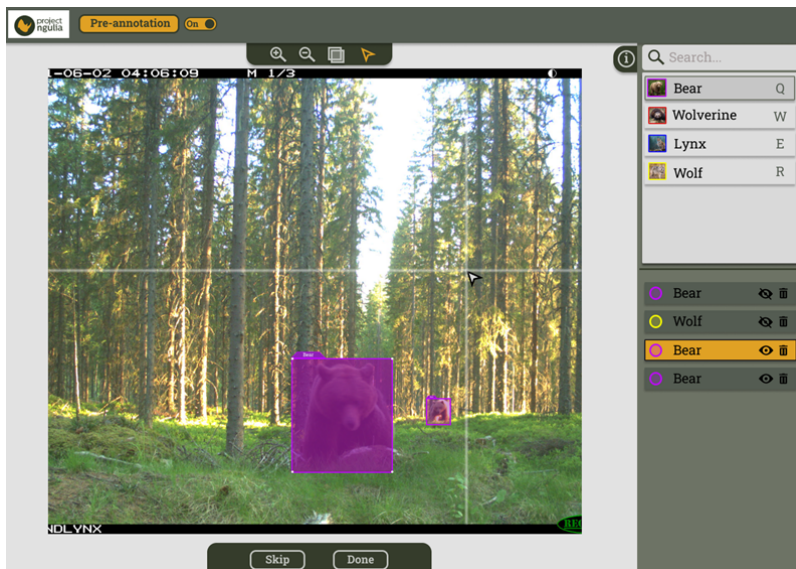
### 3.1.3 Target Audience

The tool is designed for people involved in the Ngulia project, such as developers and rangers as well as individuals with some technical understanding. The target audience includes a broad variety of users with different needs and knowledge. Therefore, the tool needs to have a simple and user-friendly interface and functionality that is not too complex. It should also support users in the annotation process by using image classification data to provide helpful suggestions for annotations.

## 3.2 Prototype

In order to simulate the real product, a prototype was created with the identified target audience in mind. Inspiration for the appearance and functionality came from the initial research of other available annotation tools. The design was created following the Ngulia brand, incorporating the designated colors and fonts specific to the brand.

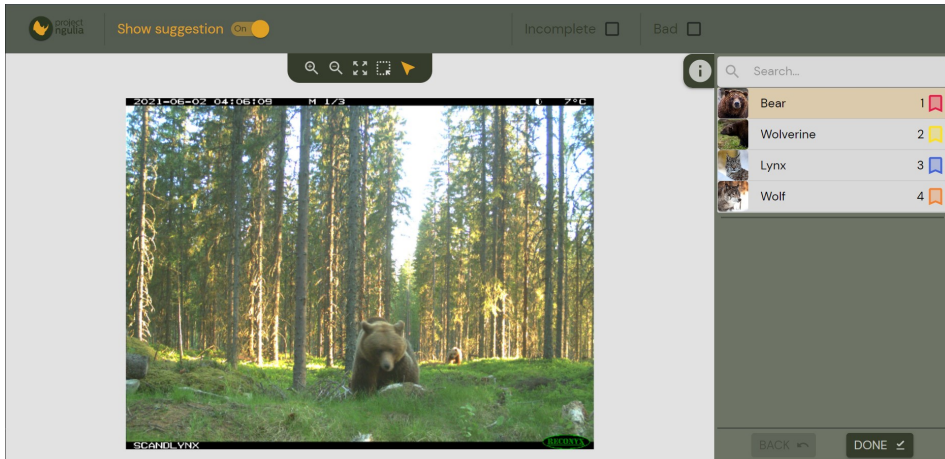
The prototype was created in Figma [26]. The objective was to create a straightforward and user-friendly layout. The main components consist of the header, toolbar, image canvas, class list, instance list, and shortcut information tab. The central focus of the application revolves around the image canvas, where the current image is presented, and users can effortlessly draw bounding boxes around animals. The chosen color scheme encompasses a blend of green, grey, and yellow hues. The prototype is displayed in Figure 3.1.



**Figure 3.1:** The prototype of the annotation tool created in Figma

### 3.3 First Version

A first version of the annotation tool was implemented in accordance with the prototype, with the exceptions of some minor changes to the appearance, see Figure 3.2. The tool communicates with a web server and a SQL-database where classifications and annotations are stored. The tool provides an information page with a general guide of important concepts such as *Incomplete* and *Bad*. An image can be marked as *Incomplete*, which means that the image contains animals that has not been annotated. This can be used in the case of missing classes and uncertainty of the annotator regarding which class an object should belong to. An image can also be marked as *Bad* when there is a fault in the image caused by the camera, which means that the image is destroyed or broken. There are several keyboard shortcuts available to enhance the efficiency of the annotation process.



*Figure 3.2: First implemented version of the annotation tool.*

### 3.4 Dataset

The image dataset available for development and testing was the NINA database, which contains images of the four largest Swedish carnivores: bear, wolverine, lynx, and wolf. Upon retrieval, the images were organized into individual folders, each representing a specific class. However, they were not accompanied by bounding box labels. The dataset has a distribution of 441 images of bears, 2,305 images of wolverines, 6,258 images of lynxes, and 7,387 images of wolves. This dataset was expanded with some images captured at Järvsö Zoo, containing 100 images of bears. It is important to mention that some of these images may be empty as they were captured by a motion-detection camera, which sometimes triggers without any actual animals present. A total of 2024 images from this dataset was classified by the base object detection model.

### 3.4.1 Base model

The base model used for classification was the CenterNet object detection model trained by Olsson & Linder [1] for the four Swedish carnivores. CenterNet is a deep neural network architecture designed for object detection that utilizes a keypoint triplet to represent each object within an image [27]. The triplet consists of a center point, a top-left and a bottom-right corner points, which is used to generate bounding boxes. The model was pre-trained on the Common Object in Context, COCO, 2017 dataset [28] and then trained using data from the NINA database, with a total of 356 images. The distribution between classes is shown in Table 3.2.

Class	Images
Bear	85
Wolverine	89
Lynx	91
Wolf	91

*Table 3.2: Data Distribution for the Base Model*

## 3.5 User Test

A user test was conducted to assess the efficiency of the annotation tool, and to identify any potential challenges encountered by users when using the tool. Additionally, the test aimed to compare manual annotation with ML-assisted annotation.

### 3.5.1 Test Procedure

A total of 10 user tests was performed, involving individuals with varying ages and level of computer skills. Each test was estimated to take around 30 minutes to complete. During the test, participants were provided with a set of 80 images to annotate at their own pace. Out of these, 40 images were accompanied by suggestions for bounding boxes with corresponding labels, while the remaining 40 images had no such suggestions. For the first subject, the suggestions were presented for the initial half of the images, and subsequently, for every other test, the order of suggestion was swapped. The test was performed on a laptop and the subject had a wireless mouse connected to the laptop to use if needed.

### 3.5.2 Image Data And Collected Meta Data

The image data used for the user test was obtained from the NINA database. A total of 800 images was used, where 400 of these images were classified by the base model.

The time for each annotation was measured to enable further analysis of user patterns and to assess the efficiency of the annotation tool. To gain insights from

the qualitative results of the user test, two ANOVA tests were conducted. The first test compared the annotation time across all 80 images, while the second test compared the time between manual annotation and suggestion-based annotation.

### 3.5.3 Test Steps

The test steps for the user test are listed below.

1. Initially, the subject is provided with a brief introduction explaining the purpose of Project Ngulia, along with an overview of the annotation process and the importance of gathering high-quality training data.
2. Secondly, the subject receives a comprehensive explanation of how to accurately annotate and when to mark an image as incomplete or bad. The subject is encouraged to ask questions during this part to ensure their understanding of the annotation tool.
3. Following the explanation of the annotation tool, the test procedure is described to the subject.
4. Next, the subject is given access to the annotation tool and a test image, in order to try out the annotation tool. The subject is asked to try every shortcut in the shortcut information tab at least once.
5. When the subject having achieved an acceptable level of comfort with the functionality of the annotation tool, the test begins. Throughout the test, the test supervisor closely observes the subject's behavior and takes notes regarding any relevant user patterns or observations.
6. After completion of the test, the subject is asked to answer a short survey, see Appendix A.

### 3.5.4 Results

The measured annotation times and the annotation speed for each participant across the two sections of the test is presented in Table 3.3. The annotation time for each participant per image can be found in Appendix B. The measured annotation times for each section of 20 images in the user test is presented in 3.4. The table illustrates the distribution of time across a total of 80 images, divided into four sections of 20.

The results of the ANOVA tests are displayed in Table 3.5. Two tests were conducted: one comparing the annotation time across the 80 images, and the other comparing the times measured for images with suggestions against the times measured for images with manual annotation. Each ANOVA test utilized a total of 800 samples. Both tests were conducted using a significance level of  $\alpha = 0.05$ .

Subject	Time (min)			Speed (s/annotation)		
	Suggestions	Manual	Total	Suggestions	Manual	Average
1	6.295	5.69	11.985	9.443	8.535	8.987
2	4.50	8.01	12.51	6.75	12.015	9.383
3	4.734	3.898	8.632	7.101	5.847	6.474
4	4.263	8.481	12.744	6.394	12.722	9.558
5	5.679	7.512	13.191	8.519	11.268	9.894
6	4.155	6.575	10.73	6.232	9.863	8.048
7	10.07	9.323	19.393	15.106	13.984	14.545
8	4.555	6.825	11.38	6.833	10.237	8.535
9	8.846	7.129	15.975	13.269	10.693	11.981
10	5.152	7.801	12.981	7.729	11.701	9.715
Average	5.825	7.124	12.952	8.737	10.687	9.714

**Table 3.3:** The measured annotation time and speed from the user test. Annotation time is presented in minutes, while speed is presented in seconds per annotation. Both the suggestions and manual sections of the test consisted of 40 images each.

Subject	Image 1-20	Image 21-40	Image 41-60	Image 61-80
	<b>ML-assisted Annotation</b>		<b>Manual Annotation</b>	
1	3.624	2.672	3.095	2.594
3	2.967	1.767	1.836	2.062
5	3.571	2.108	3.882	3.630
7	5.304	4.766	4.770	4.553
9	5.265	3.580	3.815	3.314
Average	4.146	2.979	3.480	3.231
	<b>Manual Annotation</b>		<b>ML-assisted Annotation</b>	
2	5.154	2.856	2.387	2.113
4	5.676	2.805	1.950	2.312
6	3.745	2.830	2.894	1.260
8	3.855	2.970	3.011	1.545
10	4.959	2.842	2.633	2.520
Average	4.678	2.861	2.575	1.950

**Table 3.4:** Annotation times in sets of 20 images for the subject. Subjects 1, 3, 5, 7, and 9 began with ML-assisted annotation followed by manual annotation. Subjects 2, 4, 6, 8, and 10 began with manual annotation followed by ML-assisted annotation.

Test	P-value	F-value	DFB	DFW	Critical F-value
Suggestion vs Manual	0.00218897	9.4453	79	720	1.296
Time Variation	7.422e-9	2.323	1	720	3.8544

**Table 3.5:** The *P*-values and *F*-values obtained from the two ANOVA tests in addition to the degrees of freedom between groups (*DFB*), degrees of freedom within groups (*DFW*), and the critical *F*-value.

### 3.5.5 Observations

The following observations were made by the test supervisor.

- Some subjects made mistakes during annotation that went unnoticed, such as failing to assign the correct class. This occurrence was slightly more frequent than anticipated. A potential solution to this issue could be to implement a review process whereby users review their annotation before submission, and/or have the annotations reviewed by another annotator.
- Many images depict only a partial view of an animal, requiring the subject to draw a bounding box along the edge of the image. The annotation tool automatically completes the box upon the subject's cursor leaving the image, even prior to the left mouse button being released, causing the need for readjustment of the bounding box. At times, this can result in the subject being unable to initiate drawing at the image edge.
- Images devoid of any animal presence were marked as incomplete.
- In cases where the suggestions of the annotation tool were considered uncertain, some subjects marked the image as incomplete without deleting the bounding boxes. This practice can result in flawed training data if the suggestion is incorrect. Preferred practice would be for the bounding boxes to be deleted and the image to be marked as incomplete.
- The toolbar lacks an option to move the image, with only a shortcuts available, resulting in difficulties for subjects who prefer not to use keyboard shortcuts.
- An attempt to copy and paste a bounding box was made, which led to the incomplete and bad checkbox being inadvertently marked.
- Multiple subjects failed to release the keyboard button *A* prior to releasing the left mouse button, leading to the disappearance of the bounding box being drawn. This occurrence was observed to be recurring among subjects.
- An attempt to open the shortcut information tab by pressing the keyboard key *I* was made.
- An issue arose in distinguishing the animals in nighttime images.



- The bad checkbox caused confusion among the subjects, as they perceived empty, dark, or blurry images to be categorized as 'bad' images.
- The incomplete checkbox, bad checkbox and show suggestion switch suffer from poor contrast when in the 'off' state, leading to issues with detection.
- Confusion arose among some subjects regarding the shortcut information tab, as they experienced difficulty in locating the specific command they were seeking. Some attempts were made to interact with the keyboard icons within the tab, such as marking an image as incomplete.
- Some confusion arose regarding the usage of shortcuts Q and W during the annotation process.
- Some subjects mistakenly attempted to select a bounding box by clicking on the label tag.

### 3.5.6 Survey

The results from the ranking questions are presented in Figure 3.3. The results from the yes/no questions are presented in Figure 3.4. A summary of the answers to the remaining questions are presented below.

- **If you answered yes to the question '*Do you perceive that any part of the annotation took longer time than necessary?*', please describe which parts that took longer time than necessary.**

The parts identified as problematic during the test included the positioning of the shortcut for advancing to the next image and the position of the incomplete checkbox. In addition, the need for smoother zooming, moving, and drawing functionality was also highlighted.

- **If you answered yes to the question '*Do you find any of the keyboard shortcuts cumbersome to use?*', please explain why it was cumbersome.**

The keyboard shortcuts that were expressed as cumbersome included most shortcuts located at the right side of the keyboard as they were not easily accessible by the left hand. Additionally, the suggestion was made that zoom, move and draw operations should be performed without the need for holding a keyboard button. Some issues were also encountered with undo/redo.

- **What did you think was good about the tool?**

The positive feedback received highlighted the ease of use of the annotation tool, with the shortcuts improving efficiency. The interface was described as simple, lacking any unnecessary features. The intuitive sectioning of animals was appreciated with the use of colors contributing to easy comprehension.

- **What did you think was bad about the tool?**

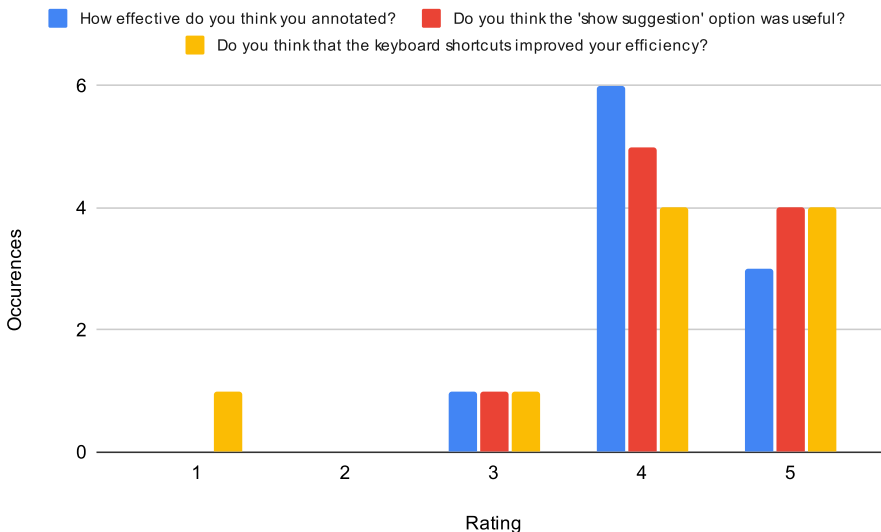
The constructive feedback addressed the challenge of drawing near the image border, leading to manual adjustment of bounding boxes. Additionally, suggestions were made for better placement of check boxes, and there was also confusion regarding the shortcuts for moving up and down in the animal list.

- **Was there anything that you would like to change, regarding functionality or design?**

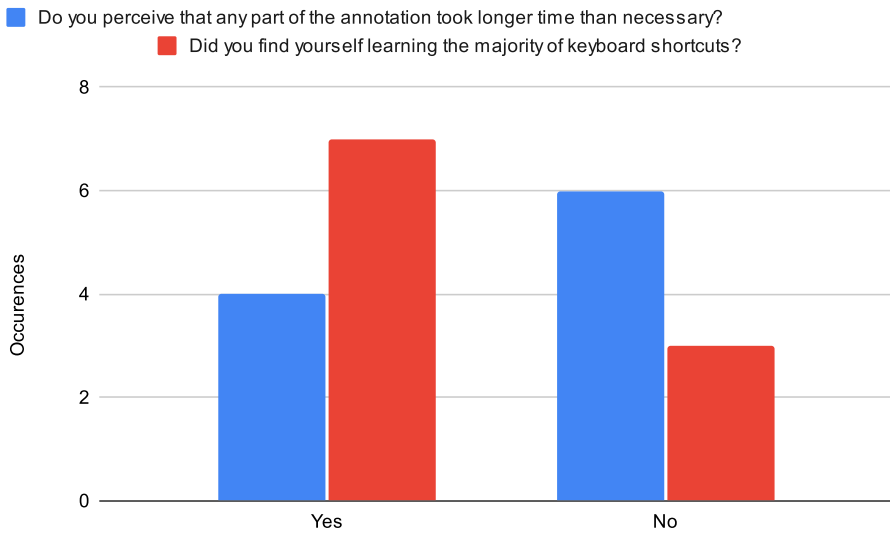
The feedback included suggestions for enhancing the information page, utilizing toggles for keyboard shortcuts, and ensuring easy accessibility of shortcuts for the left hand on the keyboard. There was also confusion regarding the numbers accompanying the classes and the shortcut instructions was perceived as using the shift key.

- **Was there anything that made you insecure or that you thought was unclear?**

The challenges of differentiating between incomplete annotations and images without animals were noted. Additionally, the suggestion was made to introduce new categories, such as identifying animals, inability to identify animals, no animals in the image, and broken images.



**Figure 3.3:** Results from the ranking questions from the survey.



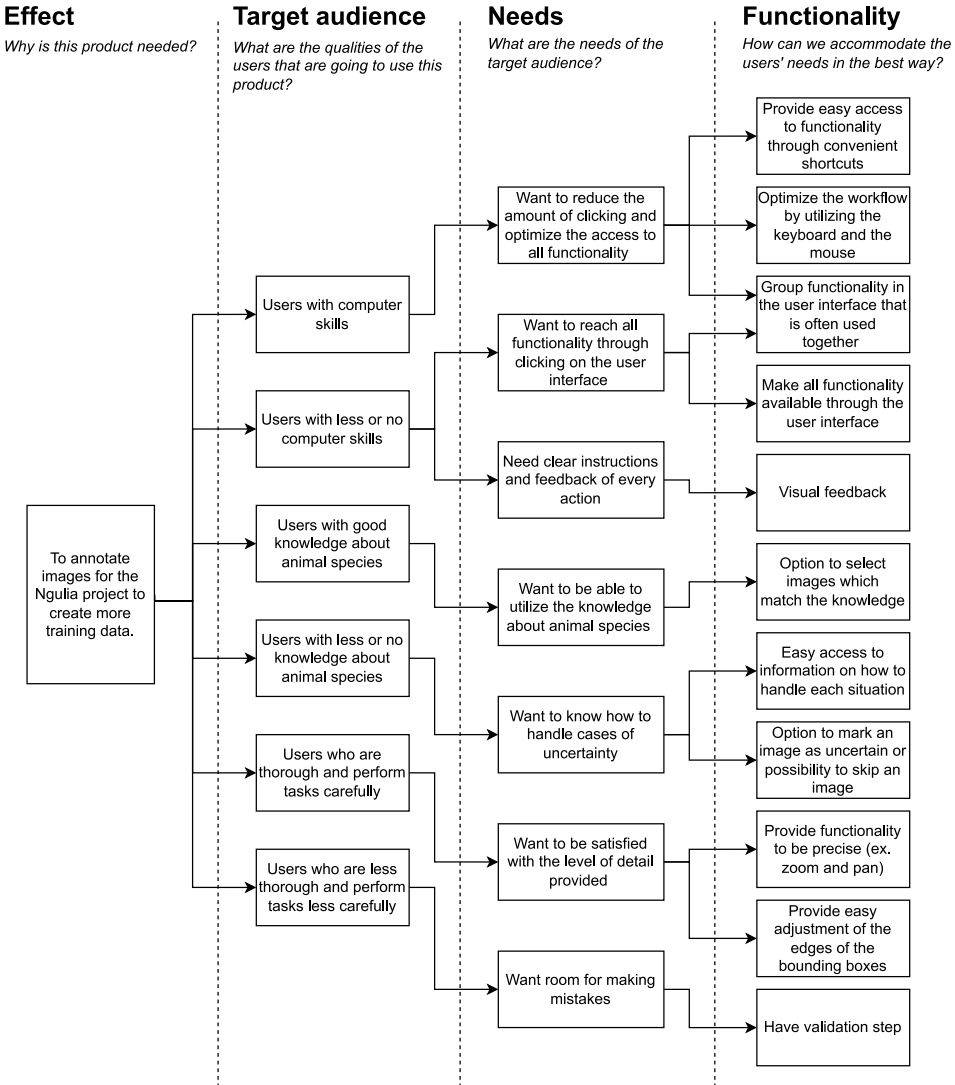
*Figure 3.4: Results from the yes/no questions from the survey.*

## 3.6 Analysis

After conducting user testing, it became evident that there were varying user needs. In order to understand how to best meet these needs, a further analysis of the target audience was performed.

### 3.6.1 Effect map

The Figure 3.5 depicts an effect map that was used to identify user profiles, their respective requirements, and the necessary functionalities that can accommodate their needs.



**Figure 3.5: Effect map.**

### 3.6.2 Changes

The effect map combined with the results from the user test, played a crucial role in identifying the key improvements that were necessary to implement. These changes specifically targeted the issues that hindered the annotation process and compromised the intuitiveness, which would result in a more streamlined and user-friendly experience. The changes that were made are listed below.

- Make it easier to draw bounding boxes at the image edge.

- Add a move tool in the toolbar.
- Make sure that undo/redo is working as anticipated.
- Make the bounding box mode a toggle.
- Change shortcuts for zoom and move.
- Change shortcuts for the keys located to the far right, in order to make them easily accessible.
- Add shortcuts for toggle bounding box visibility.
- Make changes to the label incomplete.
- Remove the previously used label bad.
- Ensure that bounding box labels remains within image boundaries, by shifting it downwards if necessary.
- Add option to skip an image.



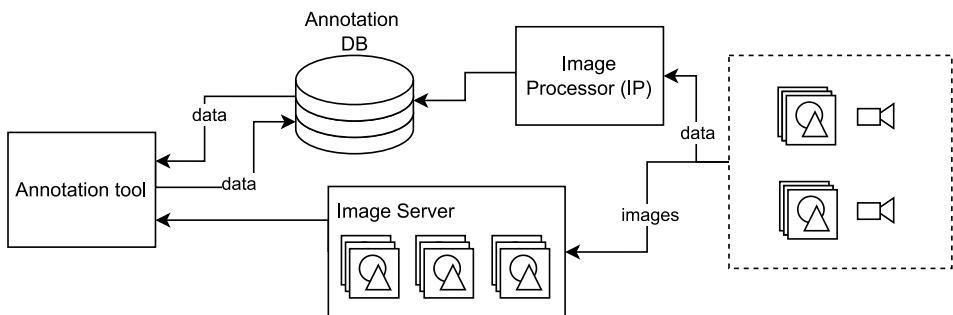
# 4

## Implementation

This chapter provides an overview of the implementation of various components of the annotation tool, including the interface, the server, and the database.

### 4.1 System Overview

The semi-automatic annotation tool will be integrated with the Ngulia system. Images captured from deployed cameras, such as those in the Ngulia sanctuary, are processed and classified by an image processor and then stored in the annotation tool database. Classified images can be retrieved in the annotation tool by human annotators, where the classification is utilized for annotation suggestions. Annotations are stored in the database for re-training of the object detection model. The system, depicted in Figure 4.1, consists of a web-based application in React, a Node.js server, and a MySQL database.



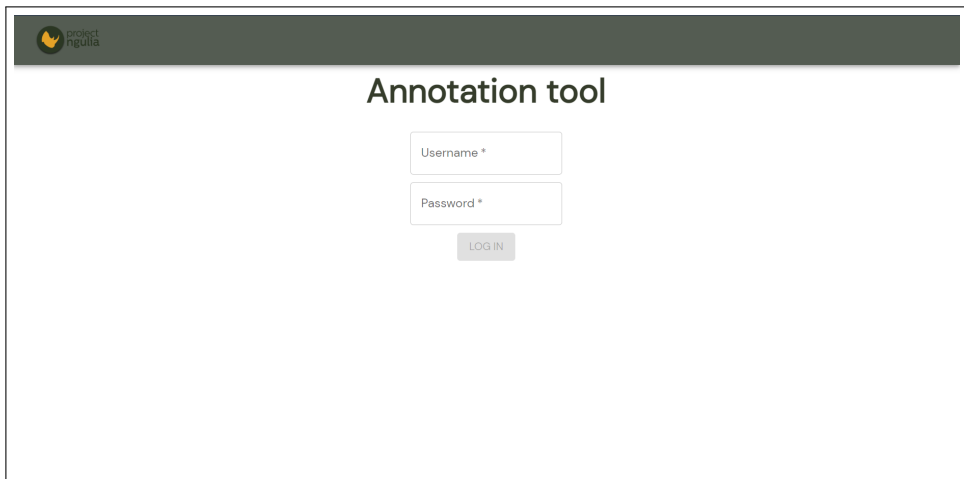
**Figure 4.1:** The system overview of the annotation tool.

## 4.2 Front End

This section contains an overview of the user interface followed by a detailed description of the different components.

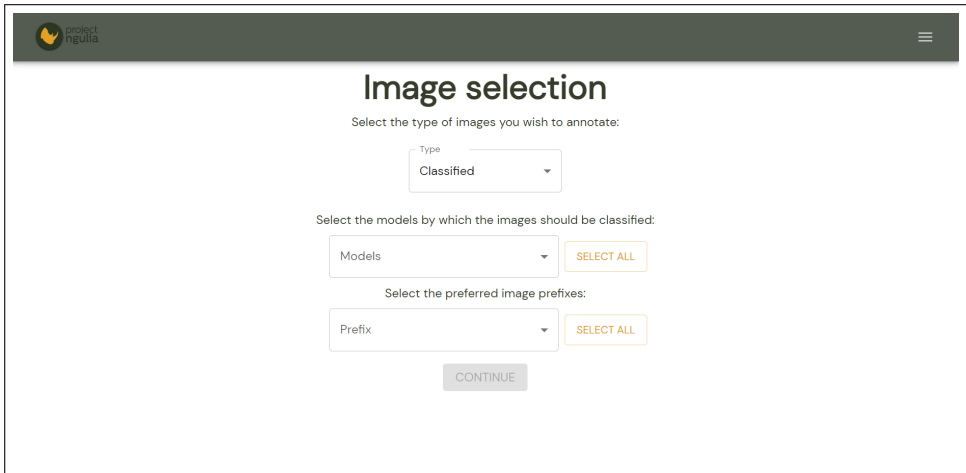
### 4.2.1 Overview

The annotation tool is built using the framework React and the library KonvaJS. The annotation tool incorporates a login system to ensure authorized access, see Figure 4.2. The system supports two user types: admin users and regular users. Both user types can annotate images using the interface displayed in Figure 4.4. Users can select the type of image to annotate using the selection page, see Figure 4.3. Admin users have additional privileges, including user creation and data extraction for model re-training. When creating a new user, the admin can assign them either to admin or regular user status, as shown in Figure 4.5. Data extraction options include CSV and JSON formats, as depicted in Figure 4.6.

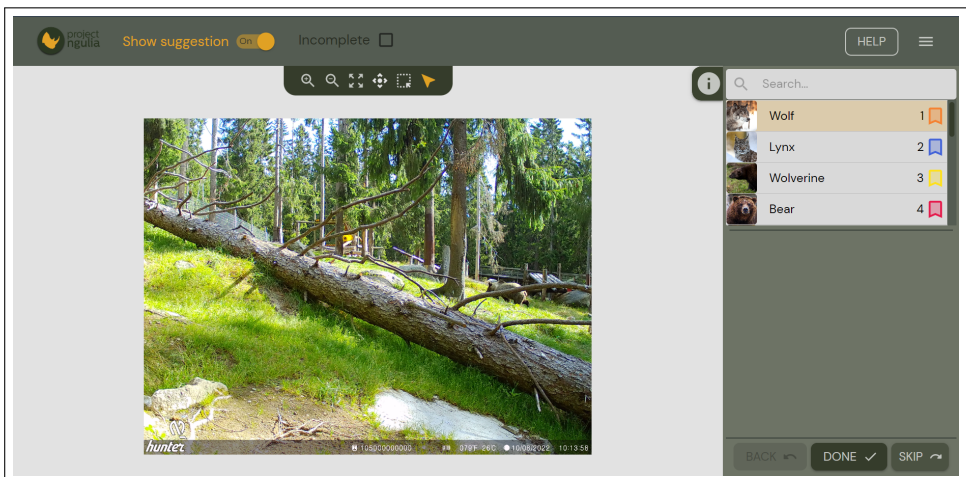
The image shows a web browser window displaying the login page of the 'Annotation tool'. At the top left, there is a logo for 'project ngulia' which consists of a stylized orange and yellow bird-like icon next to the text 'project ngulia'. The main heading 'Annotation tool' is centered in a large, bold, black font. Below the heading, there are two input fields: 'Username \*' and 'Password \*', both with light gray borders. Below these fields is a gray button with the text 'LOG IN' in white, uppercase letters. The entire page has a clean, minimalist design with a white background.

**Figure 4.2:** The interface of the login page, and the first point of entry for users. Incorrectly entered credentials will trigger the display of a warning message.

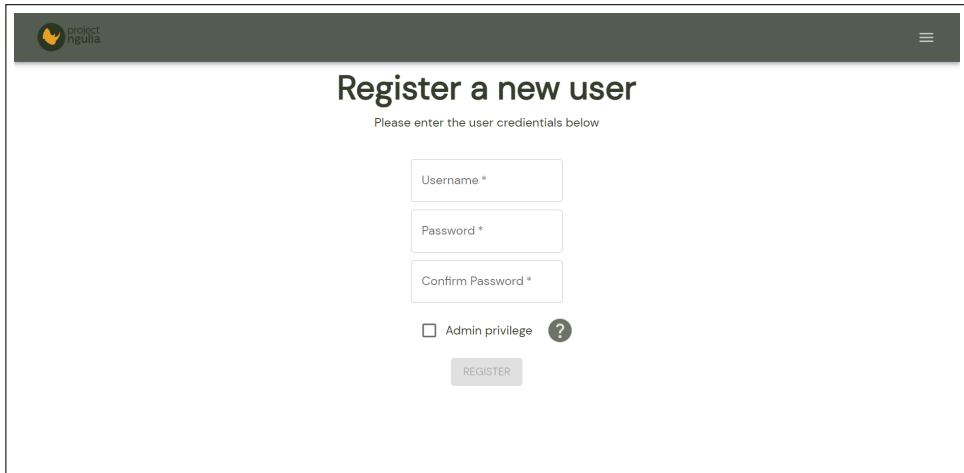




**Figure 4.3:** The image selection page. There are three image options for annotation: *Classified*, *External*, and *Incomplete*. *Classified* refers to images that have been classified by a model, allowing the tool to provide suggestions to the user based on the classification results. *External* images originate from an external source and have not undergone any classification. *Incomplete* images have been annotated but contains unidentified animals, requiring further analysis.

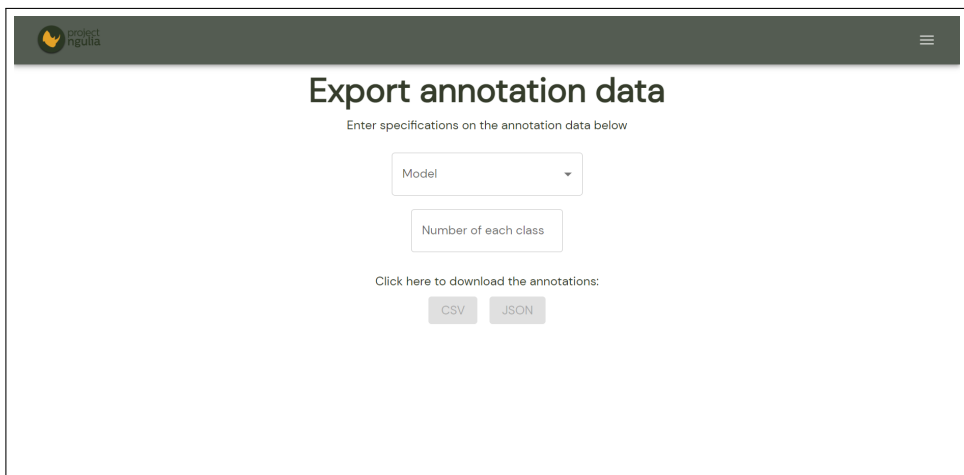


**Figure 4.4:** After selecting the desired images for annotation, users are directed to the interface of the annotation tool. This page serves as the main workspace where users can perform the annotation tasks on the images.



The screenshot shows a web interface for registering a new user. At the top, there is a dark header with the 'project ngulia' logo on the left and a hamburger menu icon on the right. The main heading is 'Register a new user' in a large, bold font. Below the heading, a subtitle reads 'Please enter the user credentials below'. The form consists of three input fields: 'Username \*', 'Password \*', and 'Confirm Password \*'. Below these fields is a checkbox labeled 'Admin privilege' with a small question mark icon to its right. At the bottom of the form is a 'REGISTER' button.

**Figure 4.5:** The user creation interface is exclusively accessible to admins and is used to add new users to the system.



The screenshot shows a web interface for exporting annotation data. At the top, there is a dark header with the 'project ngulia' logo on the left and a hamburger menu icon on the right. The main heading is 'Export annotation data' in a large, bold font. Below the heading, a subtitle reads 'Enter specifications on the annotation data below'. The form consists of two input fields: a 'Model' dropdown menu and a 'Number of each class' input field. Below these fields is a link that says 'Click here to download the annotations:'. At the bottom of the form are two buttons: 'CSV' and 'JSON'.

**Figure 4.6:** The data exporting interface is exclusive accessible to admins, providing them with the capability to choose an existing model or create a custom export with specific classes. This functionality enables efficient and tailored data exportation.

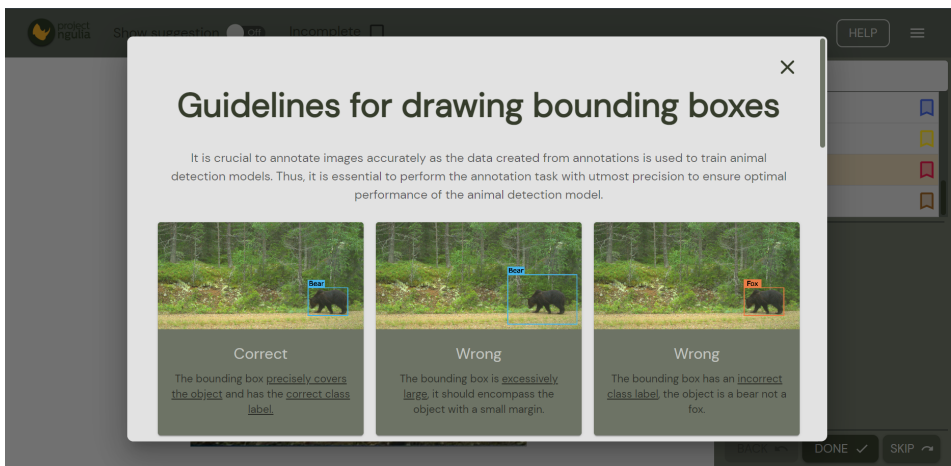
## 4.2.2 Header

The header contains a toggle switch, a checkbox, a help button and a hamburger menu, see Figure 4.7. Activating the toggle switch "Show suggestion" displays a suggestion with bounding boxes and class labels, obtained from the prediction of the model. The toggle switch "Show suggestion" is disabled for external images

without available suggestions. The checkbox "Incomplete" is used to indicate animals not in the predefined class list. The help button opens a popover with information about the annotation tool, see Figure 4.8. Tooltips provide additional guidance, and deactivated elements change from yellow to dark green.



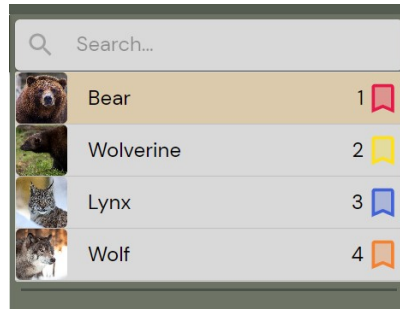
**Figure 4.7:** The header with all options turned on.



**Figure 4.8:** The information popover provides users with instructions on drawing accurate bounding boxes, handling uncertain images, using the *Incomplete* checkbox, and accessing keyboard shortcuts for efficient tool usage.

### 4.2.3 Class List

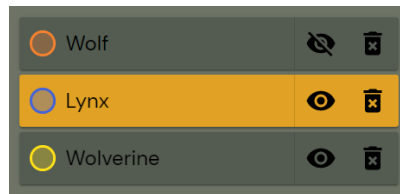
The sidebar positioned to the far right displays the available classes for labeling bounding boxes with a scrollable list, see Figure 4.9. Users can search for specific animals using the search field. Clicking on an animal image expands it for better visibility and can be closed by clicking outside the image area. The visual representation of the object classes enhances the user experience by enabling faster identification of the desired class. The first ten classes have a corresponding number indicating the shortcut key for selection. Selected classes are highlighted with a yellow background color. The color of bookmark icon represents the assigned class (e.g., red for Bear)



**Figure 4.9:** Class list containing the four Swedish carnivores.

#### 4.2.4 Instance List

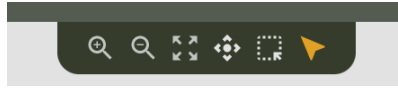
In the sidebar, below the class list is the instance list, showing the current annotated bounding boxes on the image. Each bounding box is displayed as a separate row, which provides a clear overview of all bounding box instances. Selected instances are highlighted in bright orange. The hovering over an instance will make it a darker color while the corresponding bounding box will attain a white border, assisting in easy association. Each instance offers options to toggle visibility (eye icon) and delete (trashcan icon) the bounding box.



**Figure 4.10:** The instance list, displayed with the middle instance (lynx) selected, and the visibility of the first instance (wolf) turned off.

#### 4.2.5 Toolbar

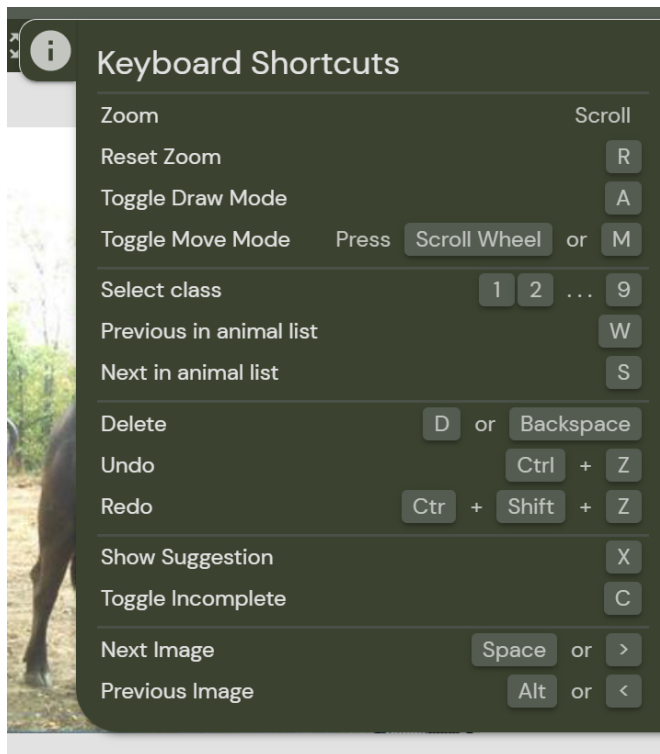
Below the header is the toolbar, see Figure 4.11, which contains a set of tools for interacting with the image. The zoom tools on the far left enables zooming in and out. The third tool from the left resets the image to its original size and position. The fourth tool lets the user navigate within the image by click and drag. The fifth tool activates the bounding box mode, where users can draw bounding boxes by clicking, dragging, and releasing the left mouse button. The bounding box is selected after release, and the mode switches back to click mode (the right most tool in the toolbar). In click mode, users can select, move, and transform bounding boxes. The selected tool is highlighted in bright orange.



**Figure 4.11:** The toolbar contains a range of tools (starting from the left): zoom in, zoom out, reset zoom, move, bounding box mode, and click mode.

## 4.2.6 Shortcuts

All actions in the annotation tool have a corresponding shortcut to make the annotation easier and reduce number of mouse clicks. There is a shortcuts information tab to make the shortcut information easy to access, see Figure 4.12.



**Figure 4.12:** Shortcuts information tab.

A more detailed explanation of the shortcuts:

- **Zoom in/out** - Scroll on the mouse wheel.
- **Reset zoom** - Press key R.
- **Turn on bounding box mode** - Press key A followed by click and drag to draw a bounding box.
- **Move around in the image** - Press the scroll mouse wheel.

- **Select class** - Press either key 1, 2, 3, ..., 9.
- **Toggle up in class list** - Press key W.
- **Toggle down in class list** - Press key S.
- **Delete a bounding box** - Press key *Delete* or *Backspace*.
- **Undo** - Press keys *Ctrl* and *Z*.
- **Redo** - Press keys *Ctrl*, *Shift*, and *Z*.
- **Show Suggestion** - Press key X.
- **Toggle Incomplete checkbox** - Press key C.
- **Next annotation (Next image)** - Press key *Right arrow* or *Space*.
- **Previous annotation (Previous image)** - Press key *Left arrow* or *Alt*.

### 4.2.7 Canvas

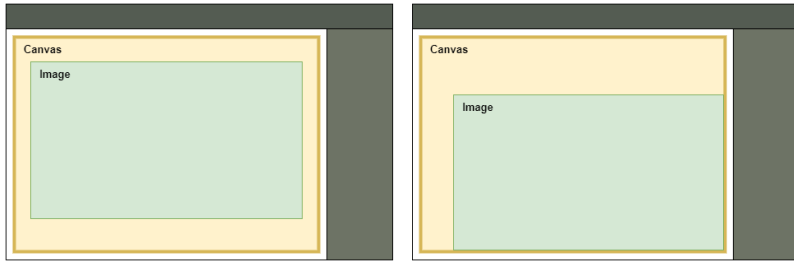
The main component of the annotation page is the canvas. The canvas allows image manipulation (e.g., zoom, move), drawing, and transforming bounding boxes. Once an image is loaded, the size is set to fully fill the canvas while keeping the image ratio and the position is set to the center of the canvas. The canvas is scalable and will automatically resize and re-position the image when the browser window is resized. The canvas contains a Konva scene which holds all graphic content (image and bounding boxes).

#### Zoom

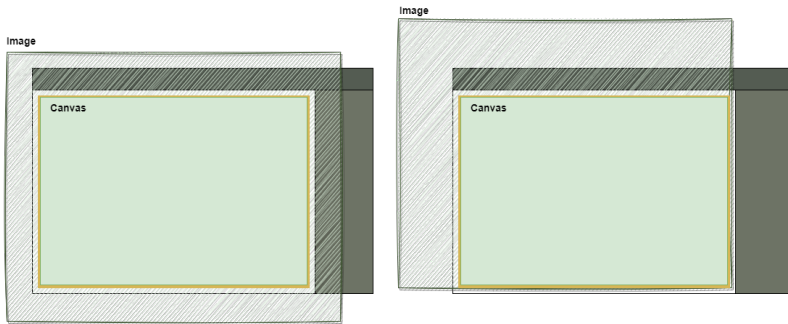
When the zoom is activated (i.e when the user scrolls on the mouse scroll wheel), the scene is transformed to fit the zoom. The amount of change in scale is calculated from the amount of scroll, as well as if the scene should increase or decrease in size. The transformation origin is based on the current mouse position. The scene is resized and positioned to represent the zoom. If the transformation representing the zoom would move the image out of the canvas, then the transformation origin is adjusted accordingly.

#### Move

When the move mode is enabled (activated by pressing the mouse scroll wheel and then clicking and dragging the canvas), the scene adjusts based on the mouse movement. Boundaries are in place to restrict the movement of the image within the canvas. When the image is smaller than the canvas the boundaries prevents the image from moving outside of the canvas borders, see Figure 4.13. When the image is larger than the canvas, the boundaries are flipped so that the image edge cannot be moved within the border of the canvas, see Figure 4.14.



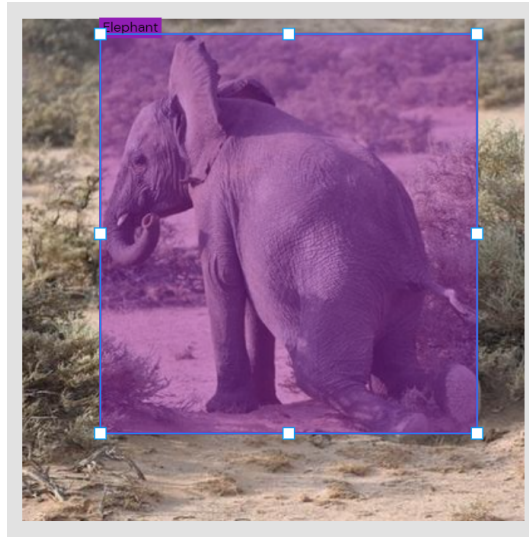
**Figure 4.13:** In the left figure, the image is smaller than the canvas. In the right figure the image is at the bottom-right canvas boundary.



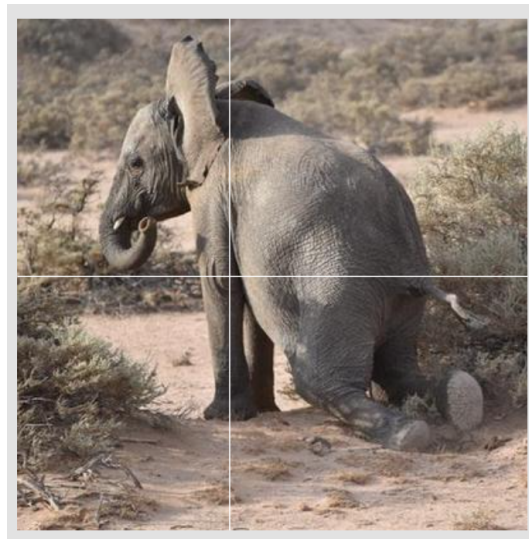
**Figure 4.14:** In the left figure, the image is larger than the canvas. In the right figure the image is at the bottom-right canvas boundary.

## Bounding boxes

When the bounding box mode is activated (either by pressing the key *A* or clicking the corresponding symbol in the toolbar), users can draw bounding boxes on the canvas. A bounding box is created by clicking and dragging, with the first corner placed upon click and the corner across the diagonal is placed upon release. A bounding box is selected once placed, which makes it possible to transform the borders by dragging the transformer-handles, see Figure 4.15. Bounding boxes can also be moved within the image boundaries. When drawing outside of the canvas, the bounding box will position itself along the border upon release. Users can change the class label of a selected bounding box. In the bounding box mode, guidelines are drawn from the edge of the image to the cursor position, see Figure 4.16. These help the user to get a better understanding of where it is suitable to draw a bounding box edge in order to fully cover an animal. This aims to reduce the number of changes needing to be made to the bounding box.



**Figure 4.15:** When a bounding box is selected it shows the transformer handles.



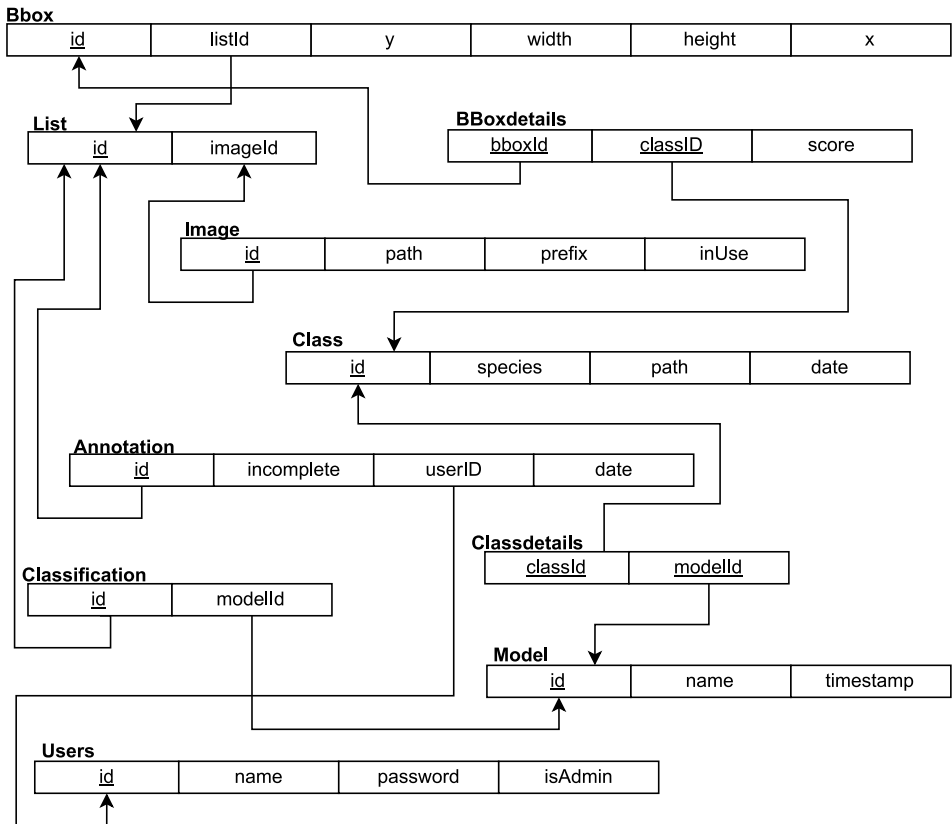
**Figure 4.16:** Guidelines from image edges to cursor position.

### 4.3 Database

To enable efficient storage of all annotations and classifications, a MySQL database was implemented. The structure of the database is illustrated in Figure 4.17 and



the relationships between the entities are illustrated in Figure 4.18 using an enhanced entity relationship model (*EER-model*).



**Figure 4.17:** The relational model of the database.

The database contains information about classified images, which is inserted from another part of the system, and annotated images. The database is structured to store meta data about bounding boxes conveniently while simultaneously reducing the amount of duplicate information (also called redundant data) because it wastes space and increases the likelihood of errors and inconsistencies. The data is divided into different database tables, see Table 4.1.

Table	Description
Image	Stores information about image path and camera prefix.
Model	Stores information about model name and date of insertion.
Class	Stores information about species, image path for identification and date of insertion.
ClassDetails	Stores information about connections between classes and models (i.e which classes belong to a certain model).
List	Stores information about the connections between annotations or classifications and images (i.e which label belong to a certain image).
Annotation	Stores information about date of insertion, the user which submitted the annotation, if the image is marked as bad or incomplete.
Classification	Stores information about which model has made the classification.
Bbox	Stores information about coordinates of the upper-left corner, the width and the height of bounding boxes and which label it belongs to (i.e which annotation or classification).
BboxDetails	Stores information about the connections between bounding boxes and classes (i.e which class and bounding box has) and the associated score.
User	Stores information about user credentials and user privilege.

**Table 4.1:** Descriptions of the database tables.

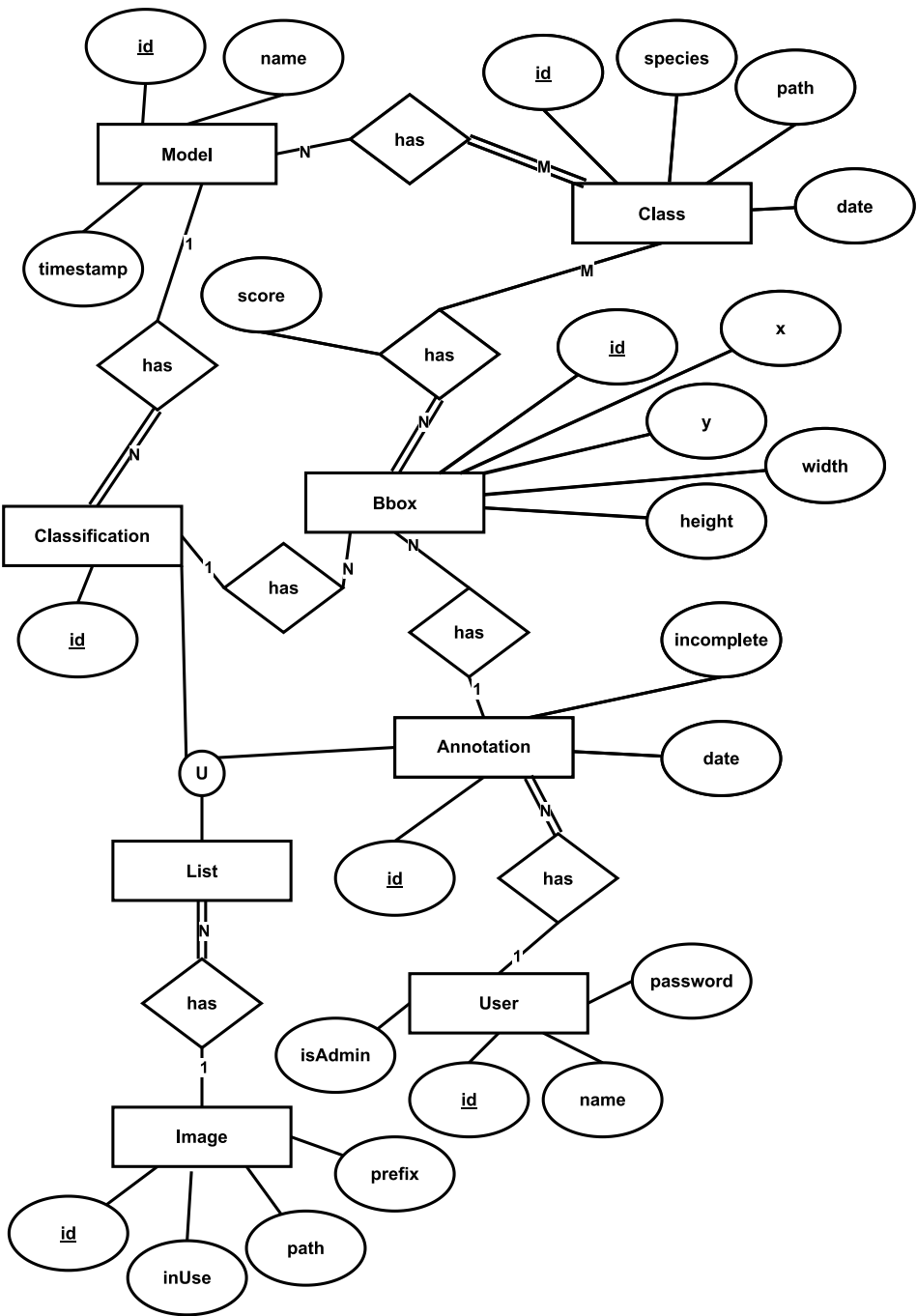


Figure 4.18: Enhanced entity relationship model of the database.

## 4.4 Server

A web server with a REST API was implemented using Node.js and the framework Express to allow communication between the user interface and the database.

The library Socket.IO [29] was used to handle simultaneous usage of the annotation tool among multiple users. It enables bidirectional and event-based communication between a client and a server. This was utilized to keep track of which images that are currently in use, hence preventing multiple users accessing and annotating the same image. An image becomes locked once it is retrieved from the database for annotation. A locked image is represented by a specific field in image table in the database (*inUse*). If the user is disconnected or inactive for too long, the image is unlocked, and the user will be re-directed to the image selection page.

A user management system was implemented by creating a database table to store user credentials and assigned privileges. User passwords are encrypted using Bcrypt [30]. JSON webtokens (*JWT*) [31] were utilized for secure transmission of information as a JSON object between parties. Upon successful login, users receive a signed JWT. Each subsequent request includes the JWT, which is verified on the server to grant access to application routes and resources. When the JWT expires, users are prompted to log in again to obtain a new token.

# 5

---

## Active Learning

This chapter describes the method and the results for the experiments conducted using active learning.

### 5.1 Data Pool

The data used to conduct the different experiments with selective annotation and re-training of the model consisted of a total of 2024 images from the NINA data set. These images were classified using the base detection model, in which a total of 974 images were classified as containing at least one animal. All images were also annotated by a human annotator, resulting in a total of 1145 images containing at least one animal.

Class	Classifications	Annotations
Bear	194	282
Wolverine	278	257
Lynx	259	344
Wolf	246	262

*Table 5.1: Data Distribution for the Selected Data Pool.*

### 5.2 Selective Annotation

In order to reduce the amount of data required to be annotated, the utilization of a technique known as selective annotation has been proposed. Selective annotation can be performed by selecting image data using different sampling algorithms, where common options include uncertainty sampling and representative

sampling. The algorithms employed for this thesis include least confidence sampling, entropy sampling, even distribution sampling, and random sampling. The model was also trained and evaluated on the entire available image pool.

The base model served as the starting point for each experiment. To obtain the training data for each experiment, the sampling algorithm was applied to the available image pool. The training process was conducted using *Google Colab* [32], *Tensorflow 2* [33], and the *TensorFlow Object Detection API* [34]. The model was trained with that data five times and the final result were obtained by averaging the outcome from each training iteration.

### 5.2.1 Image Pool

The selective annotation data pool contains a set of 974 images that have been identified as containing at least one animal by the base model. A validation dataset consisting of 160 images, with 40 images per class, was selected from the annotated dataset. A total of 849 images with classifications were allocated for training purposes. Out of these, a total of 805 images were annotated as containing at least one animal.

Approximately 72% of the data pool is selected for each sampling method, ensuring that a diverse range of data is included. By leaving 28% of the data pool unselected, we avoid training the method with excessively similar data sets. The distribution between the validation data and training data is approximately 22% and 78%, respectively, for each sampling method, except for the method that utilizes the entire available data set.

### 5.2.2 Uncertainty Sampling

By employing uncertainty sampling, it is possible to select informative and challenging samples. The implementation of uncertainty sampling follows the steps outlined below:

1. Apply the sampling algorithm to a large pool of predictions to generate an uncertainty score (based on entropy or least confidence) for each image
2. Rank the predictions by the uncertainty score
3. Select the top N most uncertain images for human review
4. Obtain labels for the top N images, retrain the model with those images, and iterate on the processes

#### Least Confidence Sampling

A total of 580 images were sampled using the least confidence sampling method (2.1), resulting in 546 corresponding annotations. The distribution of classes within the sampled images is presented in Table 5.2.

Class	Classifications	Annotations
Bear	118	132
Wolverine	174	138
Lynx	158	153
Wolf	133	123

**Table 5.2:** The class distribution in the data sampled using the least confidence method.

### Entropy Sampling

A total of 580 images were sampled using the entropy sampling method 2.3, resulting in 549 corresponding annotations. The distribution of classes within the sampled images is presented in Table 5.3.

Class	Classifications	Annotations
Bear	123	133
Wolverine	171	130
Lynx	153	152
Wolf	136	134

**Table 5.3:** The class distribution in the data sampled using the entropy method.

### 5.2.3 Representative Sampling

By employing representative sampling, it is possible to select samples that accurately represents the entire dataset. This was achieved by categorizing data based on class belonging. The implementation of even distribution sampling follows the steps outlined below.

1. Randomize the pool of predictions
2. Select N images per class for human review
3. Obtain labels for the top N images, retrain the model with those images, and iterate on the processes

A total of 580 images were sampled using the even distribution sampling method, resulting in 549 corresponding annotations. The distribution of classes within the sampled images is presented in Table 5.3.

Class	Classifications	Annotations
Bear	142	141
Wolverine	142	112
Lynx	142	155
Wolf	142	141

**Table 5.4:** *The class distribution in the data sampled using the even distribution method.*

**5.2.4 Other Samplings**

Random sampling and training on the entire available dataset were employed to enable evaluation of and comparison with the other mentioned sampling methods.

**Random Sampling**

Random sampling involves selecting data points from the dataset without considering any specific criterion or sample information. Investigating the benefits of utilizing a specific criterion for data selection compared to creating a random training set of equal size could reveal valuable insights. Random sampling runs the chance of including images that the model confidently predicts but may be incorrect, which could potentially be beneficial. A total of 580 classified images were chosen by random sampling with 557 corresponding annotations.

**Entire Image Pool**

The dataset containing the entire image pool comprises a total of 849 classifications. Out of these, a total of 805 images were annotated as containing at least one animal. The distribution of classes within the dataset is shown in Table 5.5. Including the entire available data set increases the training time for the model. Therefore, if comparable results can be achieved with less data, it might be considered a more efficient approach to model training.

Class	Classifications	Annotations
Bear	168	174
Wolverine	245	185
Lynx	215	241
Wolf	221	205

**Table 5.5:** *The class distribution of the entire data pool.*

**5.2.5 Results**

The following tables present the precision and recall achieved after re-training using five different sampling methods: Least Confident (Table 5.6), Entropy (Ta-



ble 5.7), Even Distribution (Table 5.8), Random (Table 5.9), and All Data (Table 5.10). The base model was also evaluated using the same evaluation dataset, see Table 5.11. These evaluations are performed by comparing the predicted bounding boxes of the model with the ground truth annotations. The evaluation process considers a specified number of bounding boxes (*MaxDets*) for comparison. The F1-score for each sampling method is displayed in Table 5.12.

Version	Average precision			Recall maxDets100
	IoU 0.5:0.95	IoU 0.5	IoU 0.75	
1	0.587648	0.8364	0.728359	0.71211
2	0.688589	0.912691	0.874705	0.773439
3	0.616587	0.852833	0.760344	0.731841
4	0.573501	0.781829	0.716148	0.714579
5	0.673703	0.924127	0.839851	0.753647
Average	0.6280056	0.861576	0.7838814	0.7371232

**Table 5.6:** The average precision over different IoU and recall for least confident sampling.

Version	Average precision			Recall maxDets100
	IoU 0.5:0.95	IoU 0.5	IoU 0.75	
1	0.644578	0.882886	0.776432	0.763042
2	0.689273	0.931783	0.809586	0.761150
3	0.628496	0.855054	0.760309	0.733100
4	0.584447	0.802885	0.709859	0.713168
5	0.685792	0.946063	0.847796	0.752474
Average	0.6465172	0.8837342	0.7807964	0.7445868

**Table 5.7:** The average precision over different IoU and recall for entropy sampling.

Version	Average precision			Recall maxDets100
	IoU 0.5:0.95	IoU 0.5	IoU 0.75	
1	0.682573	0.898435	0.843880	0.762989
2	0.640139	0.863818	0.785435	0.740188
3	0.675723	0.895292	0.816881	0.759009
4	0.678078	0.867724	0.831368	0.761877
5	0.738207	0.948434	0.860580	0.801618
Average	0.682944	0.8947406	0.8276288	0.7651362

**Table 5.8:** The average precision over different IoU and recall for even distribution sampling.

Version	Average precision			Recall maxDets100
	IoU 0.5:0.95	IoU 0.5	IoU 0.75	
1	0.678750	0.920118	0.831496	0.752690
2	0.540042	0.751822	0.676775	0.674429
3	0.641378	0.871800	0.765017	0.764344
4	0.624057	0.845942	0.800689	0.745702
5	0.672758	0.895205	0.822132	0.762303
Average	0.631397	0.8569774	0.7792218	0.7398936

**Table 5.9:** The average precision over different IoU and recall for random sampling.

Version	Average precision			Recall maxDets100
	IoU 0.5:0.95	IoU 0.5	IoU 0.75	
1	0.616277	0.831717	0.775243	0.728453
2	0.685654	0.915541	0.855209	0.751511
3	0.714458	0.951217	0.876178	0.781267
4	0.659312	0.882753	0.786100	0.752951
5	0.711219	0.939860	0.939860	0.767505
Average	0.677384	0.9042176	0.846518	0.7563374

**Table 5.10:** The average precision over different IoU and recall for all available data.

Average precision			Recall maxDets100
IoU 0.5:0.95	IoU 0.5	IoU 0.75	
0.552853	0.694275	0.641523	0.765704

**Table 5.11:** The average precision over different IoU and recall for the base model.

Method	F1-score
Least Confidence	0.678203
Entropy	0.692095
Even Distribution	0.721708
Random	0.681353
All Data	0.714687
Base model	0.642099

**Table 5.12:** The average F1-scores for the different sampling methods.

## 5.3 Re-training of the Model

To identify the most optimal approach for selecting data for model re-training. Four methods were explored that incorporates unseen and seen data in different ways. Training data was obtained from the available image pool for each method. The base model (described in was Section 3.4.1) then retrained five times using the respective training sets, and the results were averaged to obtain an overall performance measure.

### 5.3.1 Image Pool

The image pool consists of a total of 1145 annotated images, all of which contain at least one animal. For the validation dataset, 304 images were selected, with an equal distribution of 76 images per class. The remaining 841 images were utilized for the selection process during re-training. The base model was initially trained on 356 images, see Section 3.4.1, this training set will be referred to as the previous seen training data.

### 5.3.2 Unseen Data

This approach involves exclusively re-training the model using a dataset consisting of 841 unseen annotated images. Table 5.13 provides an overview of the class distribution within the training dataset. Unseen training data can introduce novel knowledge about specific classes. However, it is important to consider that the previously seen training data still holds valuable information that can enhance the model's accuracy. Moreover, relying solely on unseen data for training carries the risk of the model forgetting previously learned information.

Class	Data Count
Bear	205
Wolverine	181
Lynx	269
Wolf	186

**Table 5.13:** *The distribution of classes within the training dataset containing unseen data.*

### 5.3.3 Maximizing Data: Selecting All Unseen and Seen Data

This approach involves re-training the base model using all available data. This includes both the previous seen training data (356) and the new unseen data (841), resulting in a dataset containing a total of 1197 images. The class distribution within the seen data and unseen data is outlined in Table 5.14. This method allows the model to benefit from improved performance on specific knowledge

while preserving prior learning. It may seem advantageous to utilize all available data but it also presents practical challenges, particularly in terms of increased training time. As the image pool expands, training the model on the entire dataset may become impractical.

Class	Unseen Data Count	Seen Data Count	Total
Bear	205	85	290
Wolverine	181	89	270
Lynx	269	91	360
Wolf	186	91	277

**Table 5.14:** The distribution of classes within the training dataset containing unseen and seen data.

### 5.3.4 Maintaining Data Balance: Achieving an Even Balance of Seen and Randomly Selected Unseen Data

In this approach, a combination of unseen and seen data was utilized, with equal amount from each. This created a dataset with a total of 712 images, with 356 images respectively. The distribution between classes is outlined in Table 5.15. The new data was randomly selected, without any specific criteria.

Class	Unseen Data Count	Seen Data Count	Total
Bear	93	85	178
Wolverine	87	89	176
Lynx	89	91	189
Wolf	78	91	169

**Table 5.15:** The distribution of classes within the training dataset containing evenly divided seen and unseen data.

### 5.3.5 Preserving Class Balance: Achieving an Even Distribution of Unseen and Seen Data

This approach also used a dataset obtained from a combination of unseen and seen data, with equal amounts from each. This created a dataset with a total of 712 images, 356 images respectively. Instead of choosing the unseen data randomly, the class distribution was matched to the original distribution of the seen training data, as shown in Table 5.16, to maintain balance across all classes. The seen training data exhibits a relatively balanced distribution among classes, and by preserving this balanced division it is possible to prevent the model from being biased towards a specific class.

Class	Unseen Data Count	Seen Data Count	Total
Bear	85	85	170
Wolverine	89	89	178
Lynx	91	91	182
Wolf	91	91	180

**Table 5.16:** The distribution of classes within the training dataset containing unseen and seen data, evenly divided between classes.

### 5.3.6 Results

The tables below present the precision and recall obtained after re-training with four different data selection strategies: Unseen data (Table 5.17), Maximize data (Table 5.18), Maintaining Data Balance (Table 5.19), Preserving Class Balance (Table 5.20). Additionally, the F1-scores for the data selection strategies is displayed in Table 5.21.

Version	Average precision			Recall maxDets100
	IoU 0.5:0.95	IoU 0.5	IoU 0.75	
1	0.579075	0.798553	0.669292	0.701253
2	0.629001	0.924546	0.743177	0.698461
3	0.644087	0.909563	0.781598	0.721458
4	0.531904	0.792242	0.640498	0.664232
5	0.658959	0.879366	0.804606	0.739533
Average	0.608605	0.860854	0.727834	0.704987

**Table 5.17:** The average precision over different IoU and recall for unseen data.

Version	Average precision			Recall maxDets100
	IoU 0.5:0.95	IoU 0.5	IoU 0.75	
1	0.695580	0.944672	0.872873	0.758996
2	0.681112	0.907678	0.832412	0.765150
3	0.617468	0.896881	0.759403	0.713630
4	0.609789	0.866284	0.728444	0.705383
5	0.641933	0.880814	0.754330	0.716859
Average	0.649176	0.899266	0.789492	0.732004

**Table 5.18:** The average precision over different IoU and recall for unseen and seen data.

Version	Average precision			Recall maxDets100
	IoU 0.5:0.95	IoU 0.5	IoU 0.75	
1	0.628853	0.860808	0.720076	0.737140
2	0.611988	0.845383	0.717650	0.728221
3	0.662391	0.894538	0.778184	0.745962
4	0.663089	0.878902	0.787006	0.748340
5	0.658959	0.879366	0.804606	0.739533
Average	0.645056	0.871799	0.761504	0.739839

**Table 5.19:** The average precision over different IoU and recall for the method maintaining data balance.

Version	Average precision			Recall maxDets100
	IoU 0.5:0.95	IoU 0.5	IoU 0.75	
1	0.680332	0.924772	0.790755	0.747125
2	0.527496	0.730825	0.625494	0.668603
3	0.626427	0.874441	0.728781	0.721196
4	0.697491	0.910552	0.828882	0.773334
5	0.661504	0.889389	0.795886	0.747320
Average	0.63865	0.865996	0.75396	0.731516

**Table 5.20:** The average precision over different IoU and recall for the method preserving class balance.

Method	F1-score
Unseen data	0.65326
Maximize data	0.688106
Maintaining Data Balance	0.689204
Preserving Class Balance	0.681936

**Table 5.21:** The average F1-scores for the different methods.

# 6

---

## Discussion

In this chapter, the obtained results and methods of the study are discussed and analyzed.

### 6.1 Results

The process of improving an object detection model used in a real-world setting, such as the Ngulia sanctuary, relies heavily on annotated images. By integrating the annotation tool into the Ngulia system, we can leverage the existing classified images to provide annotation suggestions and optimize the annotation workflow.

#### 6.1.1 User Test

The efficiency of the annotation tool was evaluated through a user test, which provided valuable insights into the actual time spent on each annotation and the impact of annotation suggestions. The results from the survey provided valuable feedback regarding the areas of confusion and specific issues that needed to be addressed within the tool to improve the efficiency.

#### **ANOVA Test: Comparing Manual Annotation with ML-Suggestion Assisted Annotation**

The ANOVA test comparing the average annotation time between the manual annotation group and the ML-suggestion assisted annotation group revealed potential differences in the group means. The null hypothesis of this test is that the means of the two groups are equal, meaning that there is no gain from using ML-assisted suggestions for annotation. The obtained P-value of 0.00218897 strongly suggests that this result is highly unlikely to be produced by random

chance. The critical F-value for the significance level of  $\alpha = 0.05$  with  $DFB = 1$  and  $DFW = 720$  was 3.854. The calculated F-value of 9.4453 is higher than the critical F-value obtained from a statistical table, indicating sufficient evidence to support a significant difference between the means.

The statistical analysis confidently rejects the null hypothesis, which supports ML-suggested annotation as a significantly faster alternative to manual annotation. This is based on the collected data, which shows an average time difference of approximately 2 seconds per annotation between the two groups. It is important to consider that these findings are specific to the dataset used in this study, where most images contained a single animal. ML-assisted annotation could potentially contribute to even greater time savings in cases where images contain multiple animals, such as in those from the Ngulia sanctuary. Moreover, the efficiency gained from ML-assisted suggestions is dependent upon the accuracy of the underlying model. A less accurate model may require additional manual corrections, potentially diminishing the time-reducing benefits of the suggestions.

### **ANOVA Test: Comparing Annotation Time Variation Across Images**

An indication of a potential learning curve was observed when analyzing the distribution of time per set of 20 images. All subjects performed better in the last 20 images, regardless of whether they had received the suggestions first or last. To further explore the significance of these differences, a one-way ANOVA test was conducted to determine if the observed variations in average annotation times between the groups were statistically significant or simply due to random chance within each group.

The results of the ANOVA test comparing the averages among the images indicated the presence of potential differences between the group means. The obtained P-value of  $7.422e - 9$  strongly suggests that this result is highly unlikely to be produced by random chance. The obtained critical F-value for significance level  $\alpha = 0.05$  with  $DFB = 79$  and  $DFW = 720$  is 1.296. The calculated F-value (2.323) is greater than the critical F-value (1.296), suggesting a significant difference in the annotation time across the number of images. This means that the variation between the mean annotation times of the different images is greater than what would be expected due to random choice. This indicates that the number of images has a statistically significant impact on the annotation time, providing evidence for the existence of a learning curve when using the annotation tool. With more experience using the annotation tool, users will improve their annotation speed.

### **Other Findings**

Another finding was that the labeling of incomplete and bad annotations caused confusion among the test subjects. This confusion might have been a contributing factor to some subjects taking longer than others to complete their annotations. In addition to this, it was evident that individuals who were familiar with shortcuts from other applications generally annotated at a faster pace. This suggests that the familiarity with some of the functionality provided an advantage, which



could have been interesting to investigate further. Furthermore, the overall annotation time could have been impacted by the fact that each subject was assigned different images, leading to a variation in the total number of encountered empty images.

### 6.1.2 Selective Annotation

Evaluating the F1-scores of the different models made it evident that the even distribution model outperformed the other models, achieving the highest F1-score of 0.721708. The even distribution model achieved an average precision score of 0.682944 and an average recall score of 0.765136, which are the highest among all the model averages. The relatively high precision indicates the model's ability to minimize false positives, while the high recall indicates its effectiveness in detecting most of the positive instances. Therefore, the even distribution model demonstrates superior performance in terms of both precision and recall, contributing to its high F1-score.

The performance of the models is relatively consistent across the different sampling methods. Even though random sampling yielded the second lowest F1-score of 0.681353, it is still comparable to the other values. The similarity in performance among the models suggests that the advantage gained from using a sampling method to select the most informative samples is not major. Rather than being heavily influenced by a specific sampling approach, the consistent performance could depend on the relatively limited image pool. Approximately 72% of the available data was used for each sampling method, which leads to a fairly high probability of selecting similar images. A larger and more diverse data pool would likely have produced more distinct differences.

Additionally, all models perform better compared to the original base model. This improvement can be attributed to the increased amount of annotated training data, which enhances the model's ability to learn and make accurate predictions.

### 6.1.3 Re-training of the Model

The model trained using the maintaining data balance method achieved the highest F1-score of 0.689204, exceeding the performance of the other models. Comparing the F1-scores of all models reveals a very similar prediction accuracy, with only slight differences. The model trained solely on unseen data achieves the lowest F1-score of 0.65326, highlighting the importance of re-using seen data as it still holds valuable information.

Using all available training data would provide the most comprehensive dataset, in this case with a total of 1197 images. However, the maintaining data balance and preserving class balance methods achieved similar performance while using approximately 40% less images (712 images). This suggests that a more optimal approach would involve using a reduced dataset when training a new model. This becomes particularly relevant as the image pool scales up, indicating the importance of further investigation and validation of these findings.

## 6.2 Method

This section includes a detailed discussion and analysis of the dataset, the conducted user test, and the experiments of selective annotation and re-training methods.

### 6.2.1 Dataset

The ideal scenario would be to utilize the entire NINA dataset for the study. However, due to time constraints, it was not feasible to annotate such a vast amount of data.

The distribution of classes among the data pool maintains relatively balanced for both the selective annotation and re-training experiments. In reality, it is unlikely that there will be an even distribution across classes in a dataset. The distribution depends on factors such as where the images are collected from, animal populations, and animal habits. Retaining the distribution of the NINA dataset would have provided a more realistic data pool. The distribution of the data pool could affect the random sampling, as the random distribution would most likely emulate the distribution of the data pool.

The NINA dataset comprises Swedish carnivores such as bear, lynx, wolf, and wolverine. These animals tend to be solitary in nature, which implies that most images in the dataset contains the presence of a single animal, with only a few exceptions of multiple animals. In contrast to the NINA dataset, the images from Ngulia sanctuary contains savannah animals. Given the natural behavior of savannah species, it is common to encounter multiple animals in herds and coexisting with other species within these images. Annotating an image from the Ngulia sanctuary would most likely require more time compared to annotating a NINA image due to the increased number of animals in the images.

### 6.2.2 User Test

The user test provided useful information in terms of identifying potential issues with the annotation tool. Valuable insights were gained, leading to the removal and modification of confusing and unnecessary labeling options, such as bad and incomplete. These refinements were made after the annotation times were measured and could potentially have made an impact on the results.

Additionally, we contemplated the possibility of assigning the same set of images to all subjects. This approach would have provided a standardized basis for evaluating the impact of annotation suggestions, ensuring that all subjects had the same prerequisites. The reason for deciding opposite to this, was that the images annotated by the subjects could be utilized to expand the image pool used for the experiments with selective annotation and re-training.

### 6.2.3 Selective Annotation

The effectiveness of the uncertainty sampling methods least confidence sampling and entropy sampling, heavily relies on the accuracy of the model's predictions. This could be problematic if the model is overly confident in an incorrect prediction. The base model used for generating these predictions was trained on a relatively small training set. Therefore, it is possible that this limited training data could have made an impact on the execution and performance of these uncertainty sampling methods.

All the methods for selective annotation used the image data pool that exclusively consisted of images classified as containing an animal. This implies that none of the methods would select images with missed detections. Essentially, the model cannot identify what it is unaware of. It is important to acknowledge that if we have a poor model this limitation could hinder its ability to learn and adapt to missed detections.

Moreover, since each method sampled approximately 72% of the available image pool, there is a considerable chance that similar datasets were generated. A larger data pool could have yielded in more distinct differences between the methods. Alternatively, if a smaller subset of the data pool was selected for each sampling method, it could have also revealed more distinct variations.

### 6.2.4 Re-training of the Model

As there was limited availability of existing research on this topic, we decided to investigate methods which incorporated unseen and seen data in different distributions. Incorporating a smaller subset of previously seen data was not a part of these methods. Exploring alternative approaches of including smaller subsets of seen data could have provided valuable insights.

In our comparison of different approaches for selecting unseen data, we experimented with two alternative methods: maintaining data balance and preserving class balance. Due to the inherent balance in the used data pool, the distribution of classes within the datasets for the two methods ended up being similar. This could potentially explain the similar performance of the two methods. It would have been interesting to further investigate this by incorporating a less evenly distributed data pool. Therefore, we are unable to draw a conclusion regarding the effect of preserving class balance.

## 6.3 The work in a wider context

The primary objective of this thesis was to employ an annotation tool that increases the volume of training data, thereby improving the accuracy of object detection models used from classifying the images obtained from cameras, mostly in the Ngulia sanctuary. While we have successfully achieved this objective, it is crucial to acknowledge that the security of the annotation tool has not received sufficient attention due to the time limit. As a result, there are vulnerabilities that make the website susceptible to hacking and unauthorized access to the database.

While the data stored in the database itself may not pose an immediate security risk, the combination of data and images could potentially create vulnerabilities. If intruders gain access, they could easily retrieve all images with annotated rhinoceroses, potentially leading to the identification of regions with a higher likelihood of encountering these animals. Although the location data of the images is not directly accessible, there is still a risk if local individuals were to collaborate with poachers. With knowledge of the region, they might be able to identify the locations associated with the images. There is also a potential risk if the images contain identifiable geographical features or landmarks. Furthermore, if images portraits rangers working in the area it introduces the risk of their identification, potentially introducing the possibility of blackmail or temptation to collude with poachers.

# 7

---

## Conclusion

### 7.1 Research questions

- **Is ML-supported annotation with suggestions more efficient than manual annotation in terms of time spent on each annotation? If so, what is the extent of this efficiency gain?**

From the statistical analysis we gain evidence to support the conclusion that utilizing ML-suggestions for annotation leads to a statistically significant decrease in annotation time compared to manual annotation. The conducted user test reveals that using annotation suggestions results in a reduction of approximately two seconds per annotation in average. However, it is important to note that other factors such as the presence of a learning curve, may also have an effect. Conducting a user test where all subjects are presented with the same data pool and incorporating images with multiple animals could provide further valuable insights to the study and potentially impact the overall findings.

- **What is the impact of different sampling methods, such as even distribution and prioritizing uncertain images, on prediction accuracy when selecting a limited number of annotated images?**

In this study, we selected a limited number of annotated images and investigated the impact of different sampling methods on an object detection model's prediction accuracy. According to our findings, the even distribution sampling method outperformed other methods and achieved the highest F1-score, precision, and recall. The overall performance of the different models however, remained relatively consistent. This suggests that prediction accuracy may not be heavily influenced by a specific sampling method. Further investigation with a larger a more diverse image pool may pro-

vide valuable insights into the potential benefits of the suggested sampling strategies. In summary, our study highlights the importance of considering sampling methods and their potential impact on prediction accuracy when selecting annotated images.

- **How should newly annotated images be included in model re-training? Should the focus be on maintaining an equal proportion of seen and unseen images, preserving class balance, solely using unseen data, or simply maximizing data by using all available data?**

By comparing different methods of selecting training data, we conclude that the optimal method was the method of maintaining data balance between unseen and seen data, which achieved the highest F1-score and outperformed the other models. However, as the difference in performance is considerably small, it is not enough to conclude that this method is superior. The experiments did however indicate that incorporating previously seen data could be beneficial, as the method which exclusively used unseen data performed the worst. As the methods maintaining data balance and preserving class balance obtained a similar performance to maximize data, we can conclude that a reduced dataset can be sufficient. The benefits or drawbacks with utilizing an evenly distributed dataset remains uncertain, as the two datasets obtained similar distribution in the study. It would be interesting to further investigate the effect of this, as the experiments with selective annotation proved that even distribution of classes could be beneficial.

## 7.2 Future work

In this thesis, we have developed an annotation tool and explored various sampling methods, as well as data selection for retraining an object detection model. While progress has been made, it is important to acknowledge that there is still a substantial amount of work that lies ahead and improvement that can be accomplished with the annotation tool. There is still need to further investigate how to most efficiently select samples and re-train an object detection model.

### 7.2.1 Annotation tool

Similar to any other software, is annotation tool practically never considered as a finished product since there is always room for improvement. In terms of functionality and usability, there are several areas that could be enhanced, such as:

- Greater control over image selection, such as the ability to choose images from specific cameras or locations.
- Greater control over selecting specific images for export, allowing for more flexibility. Allow users to input previous training data and customize the

inclusion of the old data in the new training data. Generate separate training and validation files, allowing them to customize the distribution of data as needed.

- Implementing YOLO or a similar model within the annotation tool to classify external images without existing classifications, streamlining the annotation process.
- Move the bounding boxes with shortcuts to make it more efficient.

At Ngulia, a staff member from the computer section is assigned the responsibility of documenting the detected animals from the cameras. Each morning, they collect the cameras' SD cards since the images from the Ngulia dashboard cannot be extracted. After collection, the images are organized into separate files based on the identified rhinoceros. Additionally, important details such as time, location, rhino name, sex, body condition, etc., are recorded in a book. At the end of the month, the information from the book is entered into the KIFARU database. Considering the similarities between the documentation process and annotation, there is potential for integration. One possibility is developing a plugin program that combines both documentation and annotation, allowing the staff member to document and annotate simultaneously. Alternatively, the annotation tool itself could serve as a foundation for creating a documentation tool that also generates annotations. These approaches would streamline the process and improve efficiency in capturing essential data.

### 7.2.2 Selective Annotation and Re-training

As mentioned in the discussion, the size of the current data pool was limited, and its distribution did not accurately represent real-world data. Thus, it would be valuable to explore the methods using a larger dataset with a more imbalanced distribution.

It would also be intriguing to investigate the potential combination of selective annotation sampling methods with re-training using both seen and unseen data. This raises questions about the optimal ratio of seen to unseen data. In our study, we either used all of the seen data or none of it. Notably, excluding the seen data resulted in lower accuracy compared to combining it with unseen data. Thus, delving further into the selection of data for image object detection would provide valuable insights.

Our study primarily focuses on uncertainty sampling, which involves identifying areas of uncertainty for the model. Another approach that might be worth exploring is diversity sampling, which aims to identify what is missing from the model. This would be particularly interesting since uncertainty sampling solely selects samples that the model is already familiar with. Consequently, exploring diversity sampling methods could shed light on areas where the model lacks proficiency. If these sampling methods prove to be effective, it would be highly beneficial to implement them in the annotation tool.





# A

## User Test Questions

Question	Necessity	Description
How efficiently do you think you annotated?	Mandatory	Rank 1 to 5
Do you perceive that any part of the annotation took longer time than necessary?	Mandatory	Yes or No
If you answered yes to the question above, please describe which parts that took longer time than necessary	Optional	Free Text
Do you think the 'show suggestion' option was useful?	Mandatory	Rank 1 to 5
Do you think that the keyboard shortcuts improved your efficiency?	Mandatory	Rank 1 to 5
Did you find yourself learning the majority of keyboard shortcuts?	Mandatory	Yes or No
If you answered yes on the question above please explain why it was cumbersome. (if you have suggestions for other keyboard shortcuts, please state them)	Optional	Free Text
Do you find any of the keyboard shortcuts cumbersome to use?	Optional	Multiple Choice
What did you think was good about the tool?	Optional	Free Text
Was there anything that you would like to change, regarding functionality or design?	Optional	Free Text
Was there anything that made you insecure or that you thought was unclear?	Optional	Free Text
Other thoughts?	Optional	Free Text
Was there anything that you would like to change, regarding functionality or design?	Optional	Free Text

**Table A.1:** The table includes the questions from the survey as well as the necessity and a short description of the type of answers collected.



# B

---

## User Test Annotation Times

Image	Subject									
	1	2	3	4	5	6	7	8	9	10
1	23.429	60.298	8.786	13.42	20.295	18.703	68.943	12.264	27.788	22.831
2	6.698	22.617	3.524	11.469	22.394	8.918	9.885	25.965	20.018	22.155
3	34.977	12.403	5.22	13.512	10.881	20.812	14.916	9.892	8.165	23.228
4	19.861	8.788	4.555	10.004	22.576	12.358	15.521	9.301	3.95	11.518
5	6.935	14.142	16.162	11.562	15.867	6.224	16.953	19.931	39.408	25.192
6	8.91	10.72	11.607	129.557	21.587	11.717	10.671	18.481	68.926	7.613
7	5.629	18.794	14.24	9.466	3.753	6.728	36.636	14.357	37.873	11.578
8	4.037	15.947	3.532	8.75	8.866	14.613	5.693	11.798	32.802	2.972
9	11.192	9.341	6.623	19.939	3.982	20.073	27.559	13.51	4.101	11.239
10	13.424	10.047	14.946	11.882	1.736	13.144	3.21	8.406	2.843	28.053
11	2.626	8.421	10.322	16.186	25.342	9.714	6.627	8.106	16.799	25.82
12	13.481	10.028	9.08	16.879	6.372	9.947	9.246	4.364	6.878	14.296
13	10.236	9.517	19.4	6.184	2.687	17.781	4.58	17.621	4.74	18.974
14	9.82	6.928	4.145	4.67	8.926	9.716	8.59	6.731	5.257	11.809
15	6.826	4.42	14.278	7.695	1.348	8.924	6.237	2.6	6.061	9.116
16	20.8	8.814	1.676	14.543	18.13	3.612	32.693	8.613	3.609	19.03
17	6.891	9.02	16.808	11.521	1.337	10.912	15.448	15.076	1.497	9.289
18	3.885	5.433	6.946	7.619	5.253	3.368	9.468	14.188	13.401	6.723
19	3.15	17.438	3.33	7.584	9.607	8.879	11.269	5.702	9.164	4.152
20	4.606	46.141	2.824	8.126	3.318	8.583	4.116	4.385	2.647	11.966
21	3.586	10.951	3.073	8.297	3.196	14.294	9.701	5.64	10.293	13.638
22	3.971	8.942	8.24	6.293	2.243	9.478	8.262	2.2	14.953	36.344
23	1.502	9.374	1.115	9.484	2.824	3.506	32.086	9.482	15.421	4.539
24	9.922	10.48	6.947	11.692	19.187	8.39	11.799	10.399	24.226	9.569
25	2.853	5.928	2.84	8.328	2.116	11.387	4.105	22.062	6.121	1.492
26	2.834	5.171	3.566	4.198	2.054	10.058	16.667	12.148	15.731	4.058
27	2.643	2.531	8.483	7.931	3.42	5.7	5.589	7.465	2.625	7.139
28	4.375	11.512	2.381	6.338	4.085	5.018	18.566	5.052	5.346	9.815
29	3.693	8.103	5.696	6.483	2.294	11.495	24.552	20.911	6.897	5.383
30	2.606	9.045	3.42	8.469	5.822	8.916	12.845	2.313	8.442	9.702
31	2.232	7.912	3.421	12.847	12.618	6.903	8.121	5.963	2.846	8.226
32	3.859	10.129	3.023	10.214	2.006	8.521	30.6	5.777	14.279	3.05
33	2.476	13.537	3.235	6.225	2.42	10.087	7.627	9.03	13.289	11.027
34	4.905	11.002	5.24	6.172	15.226	7.005	13.267	9.844	2.254	8.078
35	25.506	6.42	3.929	6.658	18.579	8.603	6.953	2.167	12.686	9.54
36	20.274	7.69	4.422	3.634	2.479	6.401	4.758	13.854	5.193	8.88
37	19.133	7.606	5.027	23.08	0.812	8.56	9.017	9.396	1.855	6.004
38	4.448	10.343	21.832	4.301	2.221	7.099	29.894	9.359	19.176	3.228
39	3.563	8.354	2.678	7.694	18.34	9.66	19.212	3.281	12.938	9.012
40	35.911	6.332	7.469	9.964	4.555	8.722	12.338	11.861	20.249	1.779

**Table B.1:** The measured annotation times for image 1 to 40 for all ten participants in the user test.

Image	Subject									
	1	2	3	4	5	6	7	8	9	10
41	8.572	13.316	7.078	6.938	11.896	10.072	12.455	47.851	6.905	13.519
42	15.347	16.287	2.376	3.716	11.174	39.347	9.851	18.883	20.999	12.33
43	7.675	11.507	6.024	6.987	10.603	17.641	21.553	11.154	8.489	6.877
44	9.934	4.975	1.427	6.014	30.55	9.72	6.925	9.015	11.393	2.982
45	10.711	10.99	8.174	20.725	12.549	11.701	25.747	13.218	8.021	4.81
46	9.58	7.465	5.47	5.729	21.932	5.692	15.291	24.481	13.058	2.205
47	3.841	9.189	7.187	4.362	13.422	7.533	21.625	4.367	8.61	1.607
48	10.519	5.861	8.96	2.936	13.035	18.254	23.279	3.148	6.414	2.651
49	8.318	5.397	7.783	2.679	8.347	3.981	9.345	2.069	12.159	8.784
50	5.678	3.755	5.193	7.359	2.529	2.041	18.892	6.843	5.374	7.112
51	8.664	4.663	5.374	5.642	5.653	5.785	16.495	3.492	8.301	4.429
52	7.702	6.961	7.401	2.151	15.519	8.976	7.437	2.592	22.926	15.891
53	6.252	5.937	4.979	4.167	2.748	3.041	6.24	3.031	16.975	2.688
54	9.52	7.096	6.107	6.535	8.974	6.006	16.031	4.168	14.004	34.273
55	19.77	2.35	2.573	3.825	11.506	8.847	4.888	1.905	13.091	4.469
56	4.593	9.024	9.878	4.683	8.587	2.621	7.819	3.378	4.242	7.631
57	10.344	3.081	1.597	4.6	9.682	3.226	12.26	2.279	5.9	2.914
58	18.125	5.17	1.884	12.135	16.217	2.004	12.54	11.078	27.165	11.766
59	6.393	3.594	1.254	2.096	9.493	4.507	25.348	4.373	8.86	2.467
60	4.191	6.623	9.436	3.749	8.521	2.652	12.16	3.305	6.003	8.56
61	6.957	9.364	4.94	2.217	8.477	2.332	3.159	4.669	0.774	12.171
62	6.405	10.622	1.546	6.483	9.521	2.116	1.827	5.611	12.067	10.428
63	9.755	10.644	7.612	12.626	11.885	3.027	1.471	1.959	14.176	4.042
64	6.633	12.648	7.33	1.623	12.643	2.688	1.686	1.788	5.379	16.003
65	3.229	6.521	5.409	12.116	13.787	17.608	8.189	4.995	24.788	10.575
66	6.338	2.442	5.915	15.503	15.964	2.586	12.022	9.088	19.664	3.403
67	9.845	11.408	10.609	2.236	17.161	2.639	15.578	6.928	3.03	11.823
68	5.712	7.825	6.729	3.288	18.846	1.561	24.265	4.759	6.378	6.025
69	6.401	3.776	7.968	2.016	11.123	3.136	13.155	6.952	18.853	3.341
70	7.637	5.07	9.203	1.606	7.559	1.874	14.574	9.637	7.274	2.464
71	6.622	3.976	7.959	1.42	2.175	2.506	10.932	6.898	3.506	29.276
72	7.117	4.896	5.292	3.423	3.875	2.679	56.714	3.784	2.477	3.055
73	6.794	2.221	4.898	4.363	8.668	2.506	26.684	2.386	11.639	1.538
74	9.403	1.589	9.339	7.3	12.074	2.25	11.142	3.03	16.387	2.658
75	11.207	2.117	7.104	1.45	12.381	2.392	2.861	5.081	5.897	12.958
76	7.007	2.738	6.087	0.972	10.251	1.906	13.462	3.889	9.571	3.154
77	10.778	9.681	4.513	2.132	10.903	16.019	13.831	1.84	14.297	7.064
78	6.781	4.155	4.811	13.577	6.281	1.82	16.655	1.217	4.394	3.453
79	9.627	1.992	4.544	4.66	10.675	1.932	12.064	1.462	9.921	5.305
80	11.409	13.081	1.914	39.733	13.543	2.05	12.924	6.709	8.377	2.44

**Table B.2:** The measured annotation times for image 41 to 80 for all ten participants in the user test.



---

## Bibliography

- [1] Johan Linder and Oscar Olsson. A smart surveillance system using edge-devices for wildlife preservation in animal sanctuaries. Master's thesis, Linköping University, Automatic Control, 2022.
- [2] Johan Forslund and Pontus Arnesson. Edge machine learning for wildlife conservation. Master's thesis, Linköping University, Automatic Control, 2021.
- [3] Sara Olsson and Amanda Tydén. Edge machine learning for animal detection, classification, and tracking. Master's thesis, Linköping University, Automatic Control, 2020.
- [4] cloudfactory. Image annotation for computer vision. <https://www.cloudfactory.com/image-annotation-guide>, 2022. (accessed: 05.12.2022).
- [5] Cheng-Chieh Chiang. Interactive tool for image annotation using a semi-supervised and hierarchical approach. *Computer Standards & Interfaces*, 35(1):50–58, 2013.
- [6] Bishwo Adhikari and Heikki Huttunen. Iterative bounding box annotation for object detection. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4040–4046, 2021.
- [7] Bishwo Adhikari, Jukka Peltomäki, Jussi Puura, and Heikki Huttunen. Faster bounding box annotation for object detection in indoor scenes. In *2018 7th European Workshop on Visual Information Processing (EUVIP)*, pages 1–6, 2018.
- [8] V7. One image annotation platform for all your ml needs. <https://www.v7labs.com/image-annotation>, 2022. (accessed: 06.12.2022).
- [9] superannotate. Powerful annotation tool. <https://www.superannotate.com/annotation-tool>, 2022. (accessed: 07.12.2022).

- [10] labelbox. The most powerful data labeling solution at your fingertips. <https://labelbox.com/product/annotate/>, 2022. (accessed: 07.12.2022).
- [11] Make Sense. Alpha make sense. <https://www.makesense.ai>, 2022. (accessed: 01.12.2023).
- [12] Abhishek Dutta and Andrew Zisserman. The via annotation software for images, audio and video. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, page 2276–2279, New York, NY, USA, 2019. Association for Computing Machinery.
- [13] Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. Active Learning for BERT: An Empirical Study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962, Online, November 2020. Association for Computational Linguistics.
- [14] Yukun Chen, Subramani Mani, and Hua Xu. Applying active learning to assertion classification of concepts in clinical text. *Journal of Biomedical Informatics*, 45(2):265–272, 2012.
- [15] Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1070–1079, Honolulu, Hawaii, October 2008. Association for Computational Linguistics.
- [16] F. Olsson. A literature survey of active machine learning in the context of natural language processing. Swedish Institute of Computer Science, 2009.
- [17] Yi Wu, Igor Kozintsev, Jean-yves Bouguet, and Carole Dulong. Sampling strategies for active learning in personal photo retrieval. In *2006 IEEE International Conference on Multimedia and Expo*, pages 529–532, 2006.
- [18] G. Sychay, E. Chang, and K. Goh. Effective image annotation via active learning. In *Proceedings. IEEE International Conference on Multimedia and Expo*, volume 1, pages 209–212 vol.1, 2002.
- [19] Mohan Singh, Eoin Curran, and Pádraig Cunningham. Active learning for multi-label image annotation. Technical report, University College Dublin. School of Computer Science and Informatics, 2009.
- [20] Burr Settles. Active learning literature survey. *Computer sciences technical report.*, April 2010.
- [21] SU Hongjin, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. Selective annotation makes language models better few-shot learners. In *The Eleventh International Conference on Learning Representations*, 2023.



- [22] Robert (Munro) Monarch. Human-in-the-loop machine learning. *Manning Publications*, 2021.
- [23] Fabrizio Sebastiani. Advances in information retrieval. 25th european conference on ir research, ecir 2003, pisa, italy, april 14-16. pages 410–424, 2003.
- [24] Tae Kyun Kim. Understanding one-way anova using conceptual figures. *Korean journal of anesthesiology*, 70(1):22–26, 2017.
- [25] Stephanie Glen. F statistic / f value: Simple definition and interpretation. <https://www.statisticshowto.com/probability-and-statistics/f-statistic-value-test/>. (accessed: 06.06.2023).
- [26] Figma. <https://www.figma.com/>, 2022. (accessed: 06.12.2022).
- [27] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6568–6577, 2019.
- [28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.
- [29] Socket.IO. Socket.io. <https://socket.io/>, 2023. (accessed: 25.05.2023).
- [30] Bcrypt. <https://www.npmjs.com/package/bcrypt>. (accessed: 01.06.2023).
- [31] Jsonwebtoken. <https://www.npmjs.com/package/jsonwebtoken>. (accessed: 01.06.2023).
- [32] Google. Google colab. <https://colab.research.google.com/>. (accessed: 30.05.2023).
- [33] Tensorflow. Training and evaluation with tensorflow 2. [https://github.com/tensorflow/models/blob/master/research/object\\_detection/g3doc/tf2\\_training\\_and\\_evaluation.md](https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_training_and_evaluation.md). (accessed: 30.05.2023).
- [34] Tensorflow. Object detection api with tensorflow 2. [https://github.com/tensorflow/models/blob/master/research/object\\_detection/g3doc/tf2\\_training\\_and\\_evaluation.md](https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_training_and_evaluation.md). (accessed: 30.05.2023).