GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Uploaddatum: 30.10.2024
Uploadzeit: 18:46

Dies ist ein von FlexNow automatisch beim Upload generiertes Deckblatt. Es dient dazu, die Arbeit automatisiert der Prüfungsakte zuordnen zu können.

**This is a machine generated frontpage added by FlexNow.
Its purpose is to link your upload to your examination file.**

Matrikelnummer: 23112861

GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Bachelor's Thesis

submitted in partial fulfillment of the
requirements for the course "Applied Computer Science"

A Data centric learning strategy for individual identification of lemurs

Georg Eckardt

Institute of Computer Science

Bachelor's Thesis
of the Center for Computational Sciences
at the Georg-August-Universität Göttingen

30. October 2024

Georg-August-Universität Göttingen
Institute of Computer Science

Goldschmidtstraße 7
37077 Göttingen
Germany

☎ +49 (551) 39-172000
FAX +49 (551) 39-14403
✉ office@informatik.uni-goettingen.de
🌍 www.informatik.uni-goettingen.de

First Supervisor: Prof. Dr. Alexander Ecker
Second Supervisor: Prof. Dr. Florentin Andreas Wörgötter

Contents

1 Introduction

Intelligent systems have fascinated and inspired humans for a long time. As early as 1770 an apparent chess robot called the "Mechanical Turk" caused a sensation. While this robot was uncovered to be a hoax, 227 years later the IBM chess computer Deep blue managed to beat the best chess player of the time Garry Kasparov. Since then systems following in the footsteps of Deep Blue have not only revolutionized the game of chess, but also beaten human in most others games (Go in 2016 [1], Poker in 2017 [2], 2019 [3]). While these systems can only be considered intelligent in a narrow domain, systems in other areas such as natural language processing, especially in the form of chatbots [4], are also showing promising steps towards understanding language at a human level.

However, the relevance of machine learning extends beyond board games and chatbots to scientific research and the limits of human understanding. The central importance of AI is underlined by the awarding of the 2024 Nobel Prizes in Chemistry and Physics: The use of AI enabled Demis Hassabis and John Jumper to develop a model for predicting the complex structure of proteins [5]. The ability to understand protein structures and beyond that to design artificial proteins is considered a very significant advancement for modern medicine. The work of the Nobel Laureates in Physics, John Hopfield and Geoffrey Hinton, laid in the 1980s the foundation for modern machine learning approaches with their contribution on neural networks. [6].

The current interest and relevance of the field of machine learning is often traced back to *AlexNet* [7], which was launched in 2012. It was trained for that year's *ImageNet Large Scale Visual Recognition Challenge* [8], where it convincingly beat its competitors. This sudden leap in performance led to the realization that deep learning architectures can achieve remarkable results with the right data and processing techniques. Since then, improvement have been made to training processes and architectures. However, these can only reduce the qualitative and quantitative data requirements of deep learning. Since 2012, the availability of computer vision data increased immensely. However, recently doubts have emerged about the correctness of the label in these datasets [9]. Besides the labeling errors, it has been repeatedly shown that correctness is not the only measure for data quality [10, 11, 12, 13]. However, even though many criteria for good data have been developed, these are often not enforceable. This is because the annotators are often underpaid click workers in Africa or Southeast Asia, without an adequate educational background [14]. To address these challenges and find automated or semi-automated solutions to combat them, a new field within machine learning has recently emerged: called data-centric learning or data-centric AI. The focus of the field is not on architectural improvements, but on ameliorating quality and quantity of the training data. This has the potential to improve model performance while reducing time and costs. The methods for collecting qualitative data can vary greatly depending on the domain and format of the data.

This thesis focuses on video data depicting social learning experiments with wild lemurs. One of the aims of this biological research is to record the social dynamics between lemurs during and

after the experiments. To date, over 500 hours of video data has been collected in an ongoing process. Due to this amount of video data, it is not possible to analyze these interactions by hand. To track social interactions between individuals, a solid understanding of the identities of lemurs is a key aspect of this task. The identification is performed by a Convolutional Neural Network (CNN), that makes its predictions based on bounding boxes for the lemurs. These bounding boxes are generated by a separate network, which existed prior to this work. The lemurs can be identified by their collars (Figure 1), which differ in shape and color.

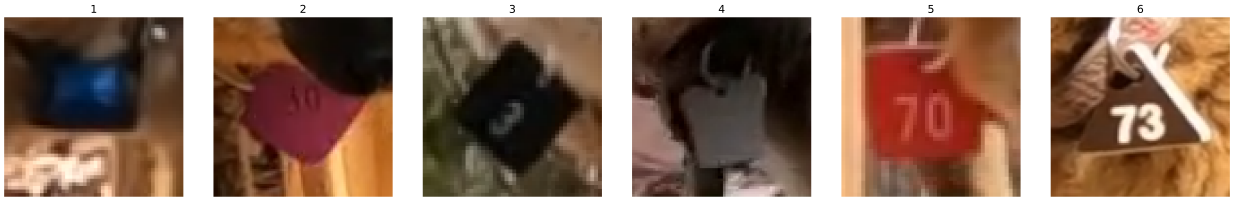

Figure 1: Collar tags of Lemur group A

This thesis builds on prior research of Richard Vogg et al. on the action capture platform for lemurs. One aspect of this work was the implementation of an object detection and tracking model [15]. This model predicts bounding boxes for all lemurs in the videos. Bounding boxes corresponding to the same object are stored in sequences or object tracks. Aside from the tracking model, further experimentation was done on the same identification task which is subject of this research. Vogg developed three different models, which predict the identity of a lemur based on a bounding box. These models are used to predict the identity of a lemur given a bounding box track. Achieved accuracy's range from 65% to 84%. The first model (65% accuracy) was trained with hand selected images in which the lemurs could be recognized. The second model (74% accuracy) was trained with labels generated by the previous model (so called pseudo labels - 2.2). The last model, which achieved with 84% the highest accuracy, was trained with a revised and corrected version of the pseudo label dataset. The creation of these models was time intensive and left space for further optimization. This is, because the previously used annotation process is slow, as in more than 80% of the frames the lemurs are not identifiable. Furthermore, multiple identification models have to be trained as there are several lemur groups.

The aim of the present work is it to propose a method to maximize the effectiveness of the human time spend on annotating data. The core idea is to train a model that preselects the data by collar visibility before labeling it. This is a modification of the idea of pool-based sampling, an active learning method. To evaluate the effectiveness of this methods, the speed and results of data annotated with this preselection step will be compared to models with data collected by random selection (prior method). For these models both the time invested and the resulting

performance were recorded. The second experiment then explores different techniques that can be used to optimise the model from the first experiment. It compares the effectiveness of two semi-supervised training methods with a model trained with additional manually labeled data. In an smaller third part, the relationship of dataset size and additional model performance will be explored. This is important in order to determine how many images are needed to achieve a certain performance.

This research found that the applied active learning methods in experiment one increased the annotation speed significantly. The selected data showed minimal differences to the raw data distribution and no measurable performance differences. The application of the preselection module in experiment two was not successful as it performed worse than the raw pseudo labels. Nevertheless, because of the achieved utility in experiment one the usage this preselection module is advisable in comparison to the prior labeling method.

2 Related work

2.1 Data engines and general structure

Large new deep learning models, like the *Segment Anything Model* (*SAM*) [16] and *DINO V2* [17] often not only develop architectures, but also curate new datasets. These datasets often surpass the previous one significantly in size. *Dino V2's* dataset for example is about an order of magnitude larger than *ImageNet* [8]. While in theory possible, in practice annotating these datasets solely through human annotators is not feasible. Thus researchers must use automated methods with which they collect and label their datasets. This part is sometimes referred to as "data engine" or "data pipeline". While their specific steps differ, they all follow the same general idea: they start with the training on handpicked labels and then gradually extend those by adding data using increasingly more unsupervised methods. One model that exemplifies this iterative process is *SAM*. It's data engine starts of with a model pretrained with existing datasets. Based on this it curates its own dataset through a collection of stages. Those are called "assisted manual", "semi automatic" and "fully automatic" stage. In the assisted manual stage the primary focus is on correcting basic errors of the model, where the focus in the semi automatic stage shifts to increasing prediction quality. In the last stage the model trains fully automated on its own predictions [16]. While *SAM's* learning target is very different from this classification task, which is subject of this work, the general idea remains the same. The amount of human involvement in the annotation process is reduced in multiple iterative steps. This concept is applicable to most deep learning problems.

2.2 Active learning and Human-in-the-loop

Active learning is comprised of different strategies to find a data subsets, which represent the totality of data well. The goal is to minimize the number of labeled samples by maximizing the performance of the model. These practices can be relevant to all kinds of machine learning problems, but are most often applied in field where data availability or the annotation capabilities smaller than by required to solve the task. This may be the case on smaller domain specific problems, in which extensive annotation work requires expert knowledge and thus is not feasible at a scale [18]. However, these techniques are also relevant as previously shown for large vision foundation models, where datasets are needed that are too large to be labeled manually [17].

Active learning is subdivided into three different approaches (Figure 2) with which the data is being selected: stream-based selective sampling, pool-based sampling and membership query synthesis [19, 20]. All three methods utilize the "human-in-the-loop" philosophy, in which the human (also referred to as oracle) annotates data proposed by the model. Thus the human remains "in the loop" in contrast to unsupervised learning methods.

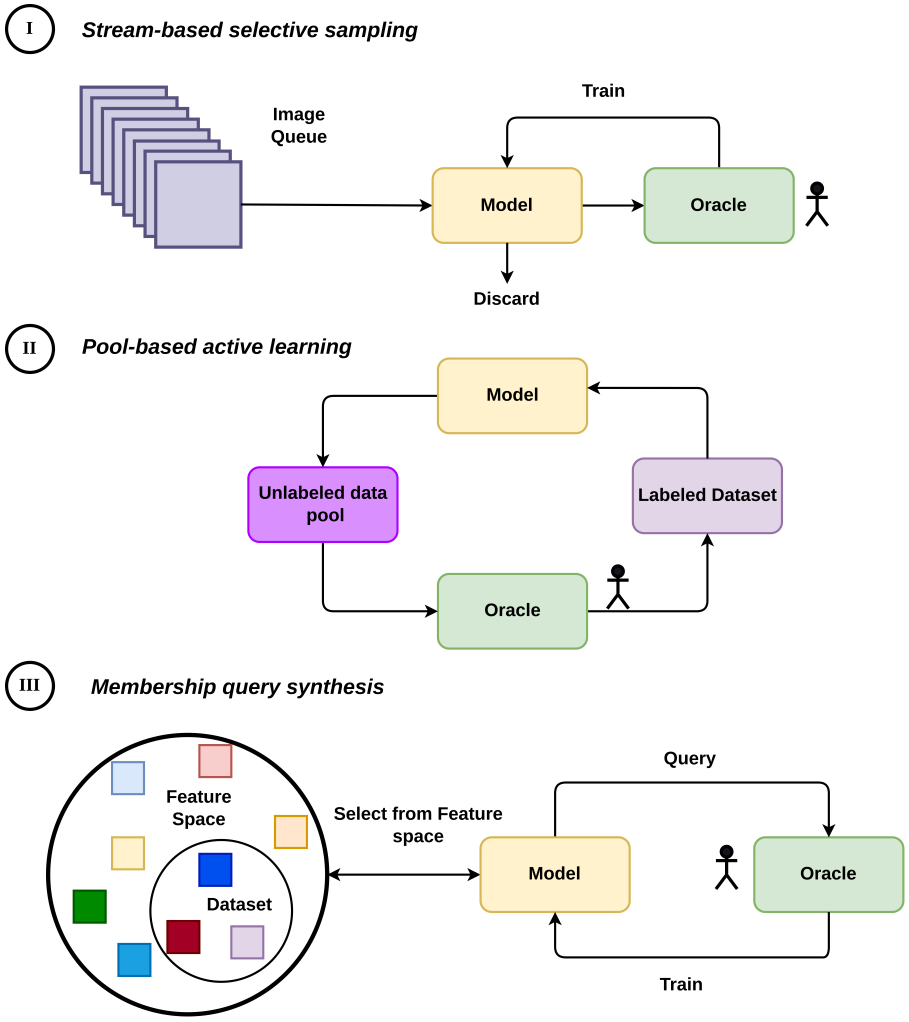

Figure 2: Schematic overview of active learning methods: I,II,III

Stream-based selective sampling, also called sequential sampling, views the dataset as a sequential data stream. The model can request annotations of its choosing form that stream. Pool-based active learning follows a similar strategy, but the decision whether a image will be annotated is not made for every instance sequentially, but across the entire dataset. Membership query synthesis allows models, in contrast to the other methods, not only to request labels for samples from data, but any sample from the feature space. This method is often not used in combination with a human oracle, since the resulting images might be hard to annotate, but in the context of a machine that labels the data. One example of such a process are General Adversarial Networks (GANs) [21].

This work applies a form of pool-based sampling. This sampling method often is used in an iterative manner: newly sampled images are used to retrain the model, which then selects new data. This research, completes two such steps: The first classifier, is trained with a binary task (collar/no collar) and the secondary model on ID-Labels. Therefore the approach used in this work is in contrast to the common method in which the classifier is always trained with the same task. In the first experiment the preselection module selects data, which is then hand

labeled. The second experiment evaluates whether the second iteration of pool-based-sampling is still necessary. It does so by comparing the model training on human annotated images in comparison to model annotations.

2.3 Semi supervised learning & Pseudo labels

Computer vision tasks can broadly be separated into supervised and unsupervised learning methods. These differ in their reliance on data: supervised methods require labels to compare their results against, while unsupervised are only reliant on the data itself. Between these two categories a number of methods exist, which use labeled and unlabeled data for the training process. These methods are called semi-supervised or weakly supervised. The aim of these methods is to acquire additional knowledge by using the unlabeled samples. Semi-supervised methods include FixMatch [22], MixMatch [23] and Mean Teachers [24]. These methods have already successfully been applied to video data of monkeys ([25]) in the form of curriculum learning [26]. A common method to do this is through the use of pseudo-labels. These are not assigned by a human annotator, but by the model itself. In the second experiment, the use of pseudo labels was compared against another iteration of active learning.

2.4 Deduplication

Large scale data collection is often carried out by using a type automated data scraping. One challenge that arises is that images can occur more than once in the data. These duplicate or near duplicate frames reduce learning performance and unnecessarily inflate the dataset as the contain redundant data. In addition this data reduces the informativeness of the data [27]. Also problematic is that the duplicates defeat the purpose of validation and test set. This results in poorer training outcomes but better reported model quality. This is because early stopping depends on a functioning validation set. The Dino V2 architecture uses a specifically trained network to solve this problem [28].
When utilizing pseudo labels generated from image data a similar problem occurs. Neighboring frames will be almost identical and thus pose a problem to the training process. This work will present a simple method to reduce the number of such frames.

3 Methodology

3.1 Data

All methods explored in this work are applied to videos of behavioural experiments on wild ring-tailed lemurs. The experiments were conducted in September 2022 during dry season and April 2023 during rainy season in the wild in Kirindy Forest on Madagascar. In this research lemurs are confronted with a puzzle box with raisins as treats inside. This box opens through mechanism that changes between experiments. However, aim of the study was not to explore problem solving abilities of lemurs. Main focus of this scientific investigation was the transmission of knowledge throughout the group once it is obtained. Furthermore, social benefits of primates who teach (by demonstrating the mechanism of the box) other members of the group where of interest in this research. The lemurs are identifiable by the collars they wear, which differ in color and shape (Figure 1). The surrounding of the box was recorded by eight cameras. Camera angles were ranging from close-ups of the ground (camera 1 to 4), to aerial shots from a couple of meters distance (camera 6 to 9). This camera setup are displayed in Figure 3. A total of four groups called A, B, J and R1 of lemurs with a total of 35 individuals were studied. Over both experiment and more than 500 hours of video footage were collected.

Figure 3: Setup of the experiments with the camera positions marked in red.

3.2 Models

Labeling method

A central component of an annotation process is the way in which data is labeled. A script was used to annotate more efficiently. This program, which was implemented specially for this tasks uses OpenCV to extract bounding boxes and display them on the screen. The script then waits for a keyboard input corresponding to the displayed image. The label, the displayed bounding box and additional information about the frame and track are appended to a CSV file. Depending on the task, the script was adjusted to binary (collar visible/not visible) or ID labeling. Alternative annotation methods could be used and potentially yield in better base results. These methods and their advantages and disadvantages will be discussed in 5.2.

General model architecture

Both the preselection module and the identification models have a similar structure, which will be described in this section. The models are based on the 18 layer *ResNets* [29] architecture, pretrained with *ImageNet 1K* [8]. Fine tuning of *ResNets* or similar networks is standard practice for classification tasks. The learning rates were identified by systematically testing, until the models showed characteristic of a good learning progression. The same learning rates were used for similar models, as it was not possible to optimize the parameters for each model. This was due to the large number of trained model and the time constraints of this work. Cross-Entropy loss was used in combinations with Adams (PyTorch default parameters) optimizer. All models were trained with batch size 128 and image shuffling between batches enabled. During training data augmentations using *PyTorch v2* augmentations were applied. As augmentations horizontal and vertical inversions, random rotations and random crops were used (Figure 4). Each flipping operation was applied half of the time. Rotation angles were randomly picked between +45 and -45 degrees. Crops were applied after the image had been resized to 224×224 pixels. The cropping size was chosen arbitrarily, but not exceeding 80% of its size. After cropping the image was restored to its original size using linear scaling. Elastic and color transformation were not used as they might alter the color or shape of the collar. The effectiveness of individual augmentations was not measured, however, in combination a significant positive effect on performance could be measured.

Figure 4: Visualization of the applied augmentations.

Preselection network

The preselection module is trained with a binary classification task. The two classes are “collar is visible“ and “collar is not visible“. This network is used to select frames in which the lemur collar is visible. To acquire higher quality frames a threshold was applied to the results of the preselection module. With this threshold the balance between dataset size and percentage of correct predictions can be influenced. Higher thresholds reduce the dataset size but increase the prediction quality.

Identification network

The identification network was only trained for the lemur group A. This group is comprised of seven individuals, however, one of them does not appear in the videos and therefore could not be annotated. The identification network performs a multi-class classification task on eight classes. The first seven represent the lemur individuals, the additional class is used whenever identification is not possible or the network is unsure. During the training of the identification models a trade off between either wrongly predicted identification labels or additional “unsure“ prediction could be observed. The tendency of the model in one or the other direction could be modified by the number of unsure samples in the dataset. The lower the number of wrongly predicted unsure frames, the more mix-ups there were among the other classes.

Identifying distinct subset of frames

Before training models based on model generated pseudo labels, the number of similar frames needs to be minimized. In video sequences adjacent frames are often very similar. This leads high numbers of near duplicates. These similarities will reduce the quality of the final model [30].
Because of these local similarities, the easiest way to find distinct frames is to sort them by their position in the video and track number. Then selecting each xth element will reduce duplicates. By varying the x values the distinctness of frames and the size of the dataset can be adjusted. However, this process discards unnecessarily large amounts of data and still contains similar frames. This due to long series of similar images. As alternative method *K-means* and *DBSCAN* were tried. The idea was to cluster similar images and then to select one sample from each cluster. However, neither of these methods found very efficient clusters center, resulting in near-duplicates in separate clusters.

One potential issue of these clustering methods is that they attempt to find clusters across the entirety of the data. This unnecessarily complicates the given task, because data that cannot be a duplicate because it is not in the same video section still might influence the clustering outcome. To remove these, the following process was used: images were sorted as described above, downsized to 56x56 pixels, and converted to monochrome. This reduction not only decreases computing time and memory usage, but also increases quality of results. Averaging pixel values across color channels and areas, reduces the susceptibility to small deviations in the bounding boxes. Then, the euclidean distance from each image to its successor are calculated. This process is identical to *K-means*, except that distances are only calculated for the $n-1$th and $n+1$th elements. Ideally, element with the shortest distance to its successor could now be removed. However, this would have been not feasible from a run-time perspective or would have been much harder to implement. To shorten development time, the 1st percentile (smallest 1%) of the remaining data was removed instead. The described process of removing images was repeated until the dataset had reached desired size. This process is visualized in Figure 5. As illustrated the mean frame distance (a measurement of how similar consecutive frames are) is significantly increased through this method.

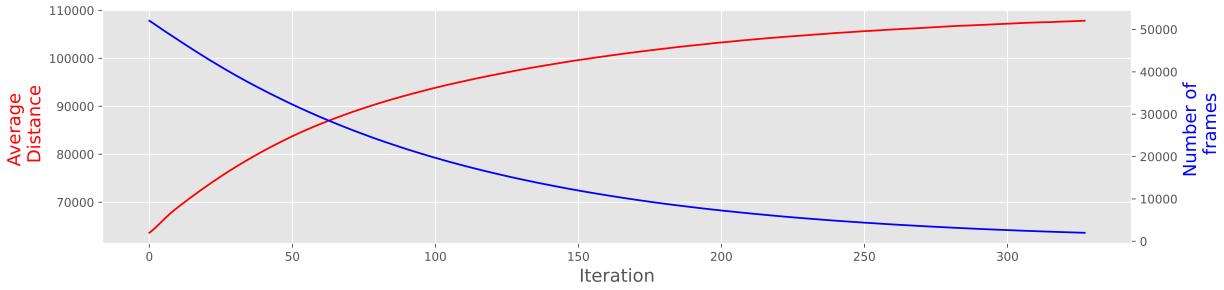

Figure 5: Visualization of duplication reduction process

This method has multiple advantages. First and foremost the results are significantly better than using competing methods. It seems not to be entirely preventable, that some similar images still remain in the data. This is especially the case when the relation of specified size to dataset size becomes too large. Other advantages are that this method scales well with dataset size and offers a simple method to determine the achieved reduction of duplicates. Using the described method the mean distance between adjacent images can easily calculated (Figure 5) and be a reference for remaining similarities in the data. This could also be used to compare different sets of data against each other. A disadvantage is that the algorithm is not available through a library and thus had to be implemented.

3.3 Evaluation

The evaluation method for the models was adopted from the earlier work on this dataset. The models are evaluated using the same process, with which they would be used in their application in the field. This was done using a collection of 30 video snippets, each one minute long. Those videos are 92 annotated lemurs tracks to which the model was applied. The aim was to assign an ID to all bounding box tracks. The first step of this process was to predict the identity values for all frames in the video track. These predictions were then weighted according to the confidence of the model. At this point in the model development process, labels with the "unsure" label were not considered (see limitations 5.1). The exponential formula $w = \exp(9.2 * (x - 0.5))$ was used as weighting function, where x is the confidence. The coefficients were chosen, such that a confidence of 0.5 leads to a weight of 1, and a confidence of 1 leads to a confidence of 100.

Aside form model accuracy the annotation and development time was also tracked. The calculated time for each model includes only human time, the compute time was discarded, as it is usually cheaper in comparison to human labor. Furthermore, to accurately compare methods, time is only attributed to methods if it is not the same for all others. Therefore, the time spend on implementing the training of the model for example does not appear in the reported numbers. Time that can be attributed to multiple sources, such as the preselection module to all lemur groups, was divided equally between them. Even though only data of three of the

four groups were used, code like the duplication reduction was distributed among all four groups.

To obtain accurate annotation speeds per frames, several sessions for different annotation scenarios were measured and the results averaged. The final times were calculated by multiplying the number of annotated frames with the averaged speed. The reported numbers were collected with a high degree of prior annotation experience with the given dataset and only during times of high concentration. Since longer labeling sessions, eventually caused a lack of concentration only the first hour of each session was used in the speed calculation. This was necessary, because otherwise different concentration levels between sessions would have had an impact on a process whose time should be constant.

3.4 Experiments

Two experiments were carried out to examine the relationship between time and performance for different data annotation strategies. The performance of all identfication models is measured using a validation set, consisting of human annotated lemurs tracks (see 3.3). The time is recorded according to the rules described in 3.3. Each identification model will evaluate these tracks as part of the larger action capture platform. The performance of a model is the percentage of correctly identified tracks. To reduce variance of the results the model is trained three times with the same parameters but different seeds.
To reduce the effort of annotation data for the sorting model of the four groups A, B, J and R1, only the first three (A, B and J) were labeled. In addition, only camera angles 1 to 4 were used, as the visibility of the collars is much higher in these angles than in the aerial shots.

Part one: frame preselection

The aim of the first experiment is to evaluate the potential benefits of preselecting the data using a classifier before manual annotation. The classifier is also refereed to as "preselection module". Therefore, two different annotation methods were compared. Figure 6 provides a schematic overview of the strategies. The experimental work of this study is embedded in between generation of the *Bounding Box Tracks* and model *Evaluation*.

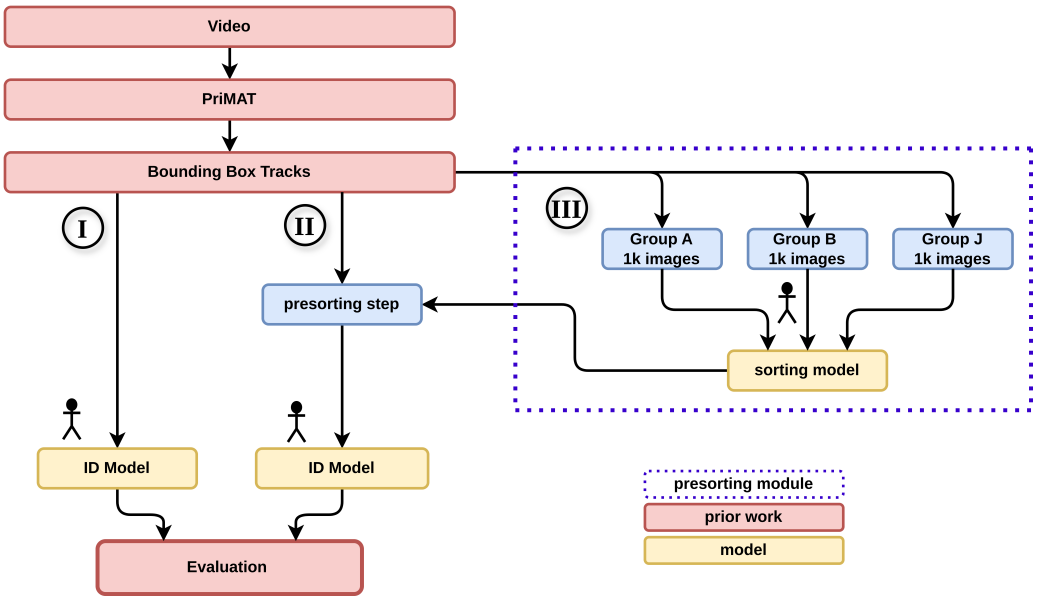

Figure 6: Schematic illustration of the first experiment embedded in larger research project. All red boxes represent the prior work of Richard Vogg described in the introduction. Path I shows the labeling strategy used in the previous work. The alternative approach II utilizes a *preselection* step before the images are annotated by a human. The purple dotted box (III) illustrates the training of the preselection module.

In the first method (I) the files are annotated without a complex selection method. However, to prevent near-duplicate frames, once a frames is labeled the second before and after that frames

is removed from the data pool. This data was annotated according to the procedure described in 3.2.
The preselection module, for method II is trained with 1000 annotated images from groups A,B and J. Training one classifier, for multiple models has the advantage that frames from other groups are also likely to contribute to the training success of the model. This way, the labeling time for the model can be split across multiple lemur groups. It is to be expected, that frames from other groups will improve performance because the collars are similar in shape and color. Furthermore, the network does not train to find specific collars, but to detect whether a collar is visible at all. Even if collars were not similar, the concept of "collar" might be trained by the images of other groups. The annotation can be executed in two ways: either the images are labeled directly with their ID, or they are assigned binary labels. In the first case, the labeling itself is faster. ID annotations, on the other hand, result in labeled data which can be used to train ID models. Which of the two methods is advantageous depends on the annotation speed and the amount of annotated data. The trained preselection module is then applied to all bounding boxes. To increase the quality of the predictions the pseudo-labels were thresholded with the values 0.9 and 0.99, resulting in two datasets. Lower thresholds have the advantage that can lead to a larger and more diverse dataset. Higher thresholds result in more frames in which the collar is visible. This increases the speed of manual annotation. To illustrate this process Figure 7 displays 14 images sampled from the raw data and 14 images sampled from the output of the preselection process. These data exhibit the same reoccurring problem of many near duplicate frames. However, due to the significantly smaller size of the dataset, the method used in I is not sufficient to remove these duplicates. For this reason, the method described in 3.2 was used. The resulting set of frames was then manually annotated.

Figure 7: Exemplary images illustrating quality difference between methods, collected by random selection (rx) and by model selection (sx) with threshold 0.99. Lemur collars are identifiable in images r5, r13, r14, s1, s3, s4, s5, s6, s8, s10, s11, s13 and s14.

To evaluate the data quality four models were trained based on labels from process I and four models were trained based on the model selected data. The first three models were all trained with 1000 bounding boxes, with 170, 400 and 800 identifiable images respectively (h170_1k, h400_1k, h800_1k), all other images are of the "unsure" class. The fourth model was trained with the same dataset as the h800_1k model, and additionally on 1000 additional unsure images (h800_2k model). The four models based on selected data are called "s90_1k", "s90_2kp", "s99_1k" and "s99_2kp". As before, the "1k" models are trained with 1000 and the "2k" models are trained with 2000 images. "s90" and "s99" indicate the threshold of that was used with the preselection model. As discussed previously during the annotation process for the preselection module it is either possible to assign binary labeled, or ID labels. Models that end with an "p" (plus) are trained with a combination of model selected data and this data.

Experiment two: Human-in-the-loop alternatives

The second experiment compares three ways to utilise pseudo labels generated by the identification model trained in the previous experiment. The focus is to identify if the preselection module can be used in this process to additionally increase the efficiency. The three ways are visualized in Figure 8. The first step of this process is always to apply the identification model to the entire dataset. Both "unsure" and "identification" labels are only stored if the model predicts them with a certainty larger than 0.9, otherwise the data is discarded. This

parameters could be adjusted depending on the dataset size. Lower thresholds result in larger datasets, while larger thresholds increase quality. This process results in a pool of more than 400.000 unsure and 52.000 identification pseudo labels. The resulting data still encompasses many duplicates, these are again reduced by using the process described 3.2. The resulting pseudo label dataset (which is always 2000 frames large for comparability) is then utilised in three different ways. The first and most simple method is to just add the raw labels to the existing dataset. The second method is to apply the preselection module, that was trained in the first step, to the pseudo labels, removing data where the collar is either not or poorly visible. For the preselection module the threshold was also set to 0.9. Both of these methods require no extra annotation time and are entirely automatic. The third alternative is to correct these labels by revising these annotations. For revising model predictions the labeling setup, was slightly changed to display the pseudo label along with the bounding box. Then there was one button which saved that label, if it was correct and the buttons previously used to label could be used to correct errors.

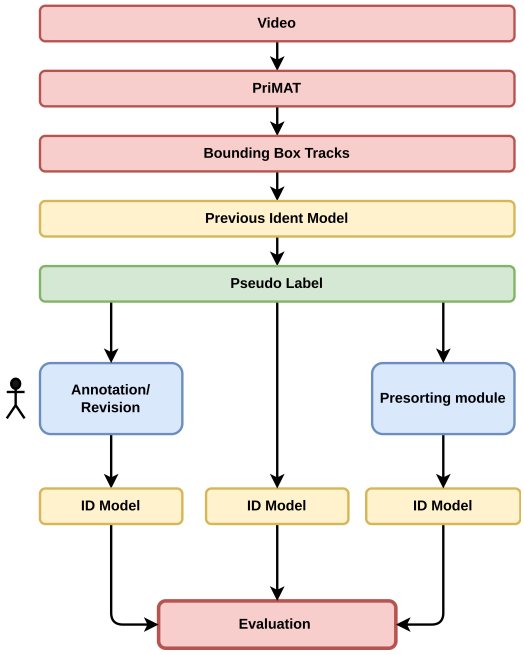

Figure 8: Schematic illustration of the second experiment - exploration of further model optimization methods: raw data, identification model with preselection module and manual annotation (revision)

Ablation study:

One important information about a learning task is, how much additional information helps the model, to learn the underlying task. To explore this relationship between labeled data and achieved results a third simple experiment was conducted. For this task the labeled data from all experiments was merged to one large dataset. This dataset has a total of 2943 identifiable images. From that large dataset smaller datasets with 250, 500, 750, 1000, 1250, 1500, 2000, 2500 images of each of the two categories were sampled. In each dataset the proportion of

identity to unsure labels is 1:1. This ratio an all further training parameters were kept the same for all model. The dataset size and models will be referenced not by total size of the dataset, but by identity labels since those are hard to obtain during the labeling process. Model were trained for dataset sizes of 250, 500, 750, 1000, 1250, 1500, 2000, 2500 images.

4 Results

The presentation of the results is structured into three separate sections: data distribution, time and efficiency and lastly the model accuracies.

4.1 Data distribution

The distribution of classes is important for many deep learning applications. Unbalanced class distributions generally lead to worse learning results [31]. Most methods assume that all classes have the same or similar number of samples. However, this assumption if frequently invalid.
A potential problem with preselection might be that the natural occurrence of classes could be altered by over-predicting some classes and under-predicting others. In this section examines the class distribution of the different methods is examined.

This first requires a understanding of the distribution in the videos. As a proxy for the raw data, we examine the 800 manually labeled images. Of these ∼83% are of the unsure class. As already mentioned, one of the lemurs does not appear in the data, so its percentage is always at zero and is not displayed in and subsequent plots. The distribution of the remaining images is shown in Figure 9. As can be seen, there are considerable differences between the classes. These could be partly due to statistical noise and individual lemurs behaviour (i.e. it could be that some lemurs are more shy). However, there are also other reasons, why classes one and four have the lowest occurrence. Their two collars were more difficult to see, than the rest of the collars. Lemur one is hard to see, because unlike other its collar is not a tag, but a small case directly beneath the head. The tags are visible from far more angles and cannot be as easily be confused as collar one. The visibility of collar four heavily depends on the lighting conditions. In low light, the silver blends in very well with any brown or grey background (i.e. fur or the leaves on the ground). The different collars are displayed in figure 1.

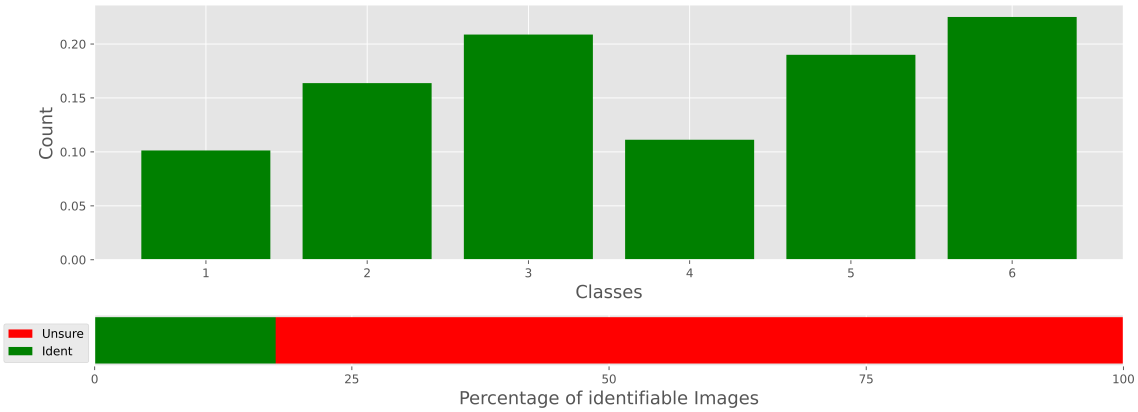

Figure 9: Graphical illustration of the distribution of raw data

As a first comparison the distribution of the classes is going to be analyzed for the preselection module. The two thresholds 0.9 and 0.99 are looked at separately. These thresholds were used with the preselection module to only select data where the model was certain. For the 0.9 threshold (Figure 10) the percentage of identifiable images (where a collar is visible) is up from 17.6% to 39%, indicating that the preselection module works as intended. However, for an ideal manual labeling process it would be advantageous if this percentage was higher. The relative distribution between the classes is not the same as in the raw data, but shows similarities. The largest difference is that the sixth class occurs drastically much less. Except that the relative differences between classes remain the same.

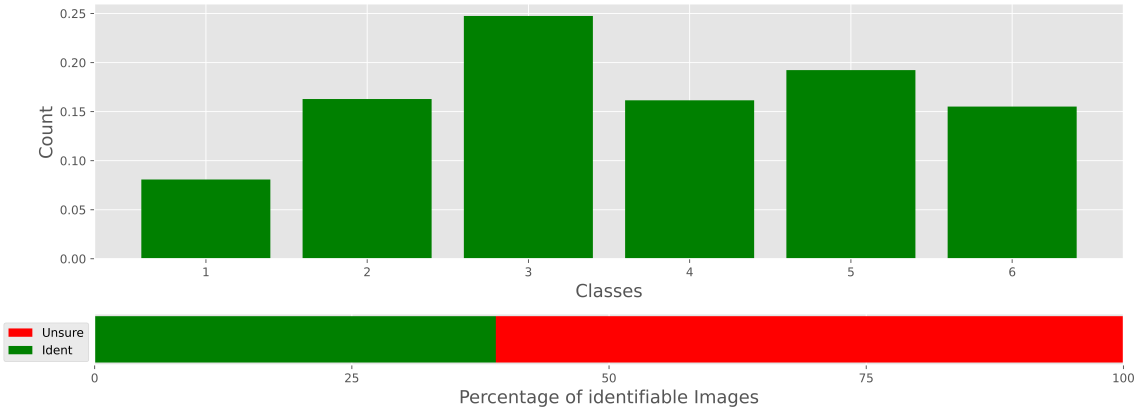

Figure 10: Graphical illustration of the distribution the preselection module - threshold 0.9

The 0.99 threshold does primarily shows a difference in the number of positively identifiable images. Now of the images 50.5% are identifiable. However, these images are not distributed equally among all the classes. Class three, which was already the largest of the classes, occurs even more frequently in this class distribution. All other classes gain absolute size, and their relative proportions remain similar.

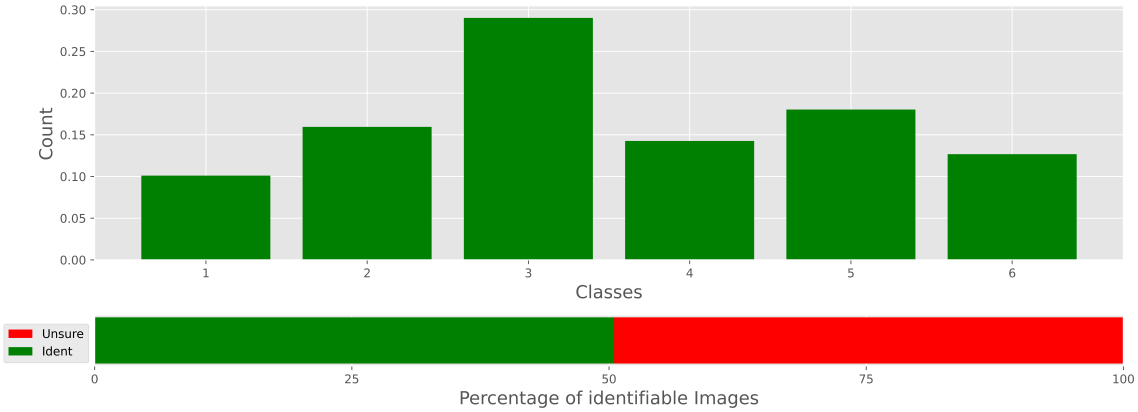

Figure 11: Graphical illustration of the distribution the preselection module - threshold 0.99

In the second experiment three models were trained: a model based on raw pseudo labels, a model based on a annotated version of that dataset and a third model trained with a combination of ID and preselection module. The distributions of the raw pseudo label dataset and it's annotated sibling are compared in Figure 12. The pseudo labels show as all other model a weakness for the first class. However, the distribution of all the other classes is more uniform compared to the raw data of the preselection model. This changes, if the corrected version of the dataset is examined. The new distribution shows that many of the predictions for the fourth and sixth class are wrong. The new distribution is more similar to data from the preselection module, with the exception that class two which is more prominent. Most of the human corrected labeling error, were of the unsure class.

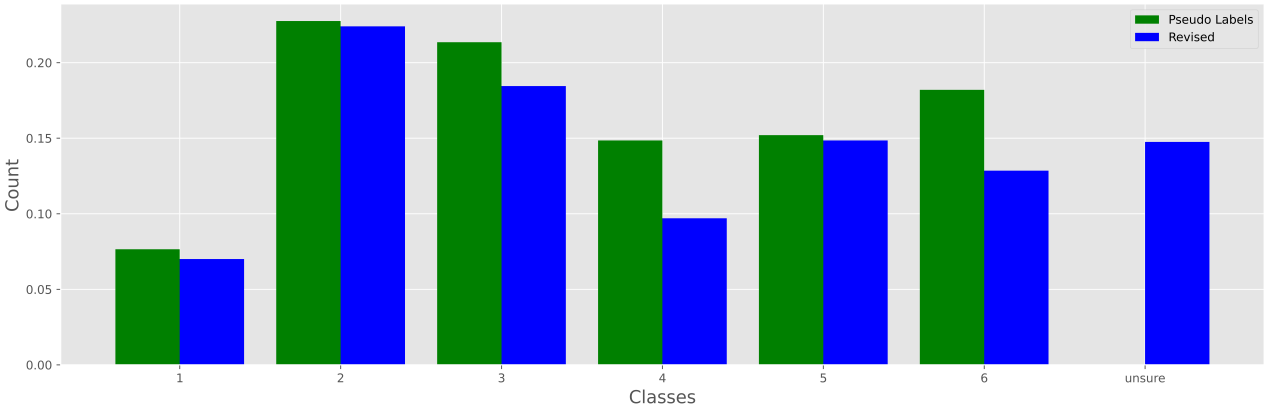

Figure 12: Graphical illustration of the distribution the ID pseudo labels before and after manual revision

The distribution of the data collected from the combination of preselection module and identification model, surprisingly shows a very similar distribution to the corrected data, and not to the raw data. This would lead to the conclusion that the preselection module reduced the number of frames which were wrongly predicted as identification labels, instead of unsure. Unfortunately in these models the difference between the smallest and largest class is even larger

than in the raw data. In the raw data the largest difference is about 2.5:1, here the difference is 3.5:1.

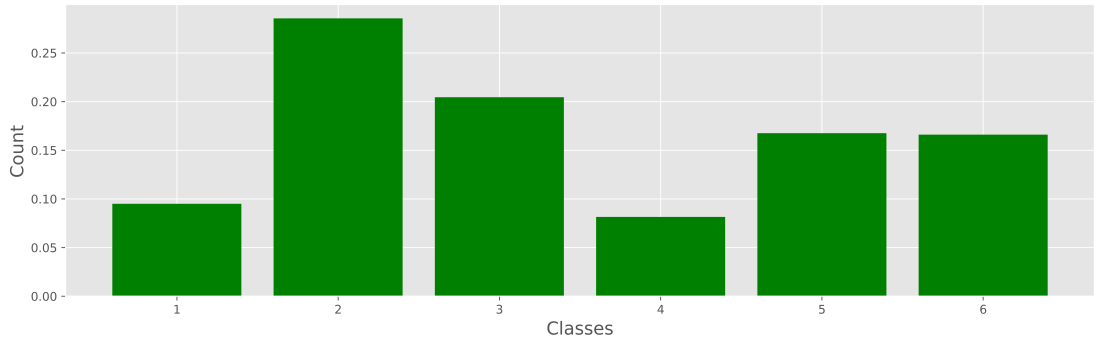

Figure 13: Graphical illustration of the distribution the identity pseudo labels of the preselection module and ID model

All in all, the datasets show difference in theirs class distribution. The most consisted trends are that the classes 1 and 4 are underrepresented, but that is true for manually labeled data and model selected data alike.

4.2 Time & efficiency

Annotation speed is understood as the speed of the labeling practice described in 3.3, with the correction of incorrect labels caused by accidental or wrongly annotated images. This speed can vary significantly according to concentration levels and annotation dexterity. Especially during longer annotation sessions, the speed will be significantly slower. It must be stated at this point that the measured times cannot provide precise data. Possible sources of error or inaccuracies can arise because the human labeling speed can vary from day to day and person to person.

Annotating based on the visibility of the collar proved to be easier than ID labeling, because there are more possible labels. The revision process (see 3.4) has proven to be slower than the ID labeling. This is because the labeling process becomes more complicated. During revision the image not only has to be categorized, but also compared against the assigned pseudo label. This means when verifying the correctness of pseudo labels, relabeling the dataset if faster than error correction. The measured labeling speeds of the three methods are shown below in Table 1.

	Binary collar labeling	ID labeling	Revision
Images per Second	1.3	1.7	1.9

Table 1: Annotation speed per processed image

In addition to the annotation time only the development time for the duplication reduction was recorded, as all other time could be attributed to both methods. The development time of the duplication revision was 40 minutes. These 40 minutes were allocated to all four groups, resulting in 10 minutes per group.
With these values the time spend on the individual models can be calculated using the equations 1 to 3. Equation 1 and 2 are applied for all models that use the preselection module. Equation 1 is used when the data are labeled for the preselection module receives binary labels. When this data is ID labeled equation 2 is used. The last equation (3) is used for annotation based on random selection. The variable p is here the probability of an identifiable image is being labeled. In the case of formula 2 p_1 is the probability in the preselected data, while p_2 refers to the raw data. ni always refers to the number of images that is labeled in time t.

With preselection module:

$$t = 600 + (1.3 \cdot 1000) + 1.7 \cdot \frac{ni}{p} \tag{1}$$

With preselection, but identity anntation:

$$t = 600 + (1.7 \cdot 1000) + 1.7 \cdot \frac{ni - \frac{p_2}{1000}}{p_1} \tag{2}$$

Random selection:

$$t = 1.7 \cdot \frac{ni}{p} \tag{3}$$

Where:

$$p = \textit{probability of identifiable images}$$
$$ni = \textit{number of identifiable images}$$

With these equations and the knowledge of the p and ni values, the development times of the models can be calculated. In Table 2 resulting development time is listed for each identification model trained in this work. For small numbers of annotated ID frames, the preselection module is slower than hand selection. However, for larger volumes the preselection module offers significant time savings (s_90_2k_p,s_99_2k_p requires less time than h_800_1k at similar amounts of ident images).

Name	Ident	Time in minutes
h_170_1k	170	28
h_300_1k	300	50
h_500_1k	500	83
h_800_1k	800	133
h_800_2k	800	133
s_90_1k	393	60
s_99_1k	520	60
s_90_2k_p	952	83
s_99_2k_p	1180	83
i_hand	2180	145

Table 2: Annotation time for trained models. All models starting with an “h“ are trained with hand preselected images, the “s“ models are trained with data from the preselection module

This increasing difference between the raw data labeling and the preselected labeling speed is further visualized by Figure 14, which plots Equations 1 and 3 for increasing ni's. The intersection of the two functions is at 271 images. Before this point raw data labeling is more efficient, after it the preselection module becomes increasingly much more favourable. Equation 2 was not included in this plot as it's difference to equation 3 was only minimal and not visible.

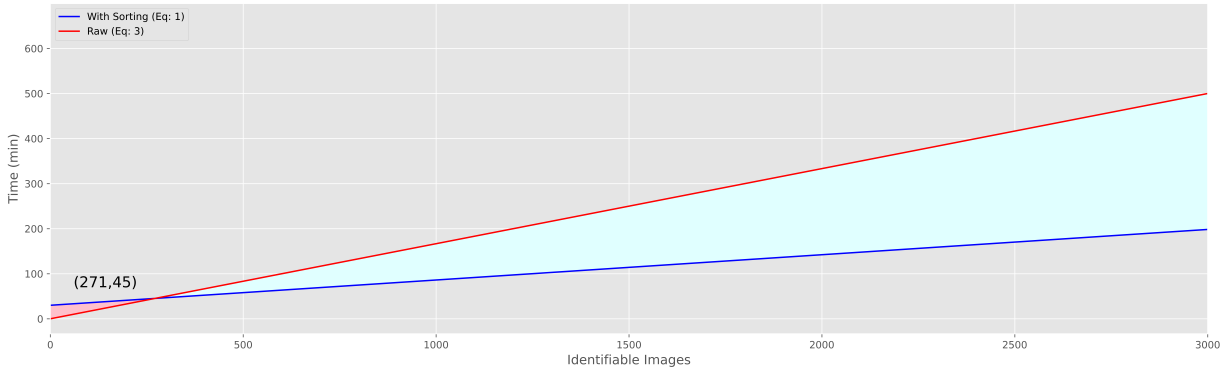

Figure 14: Plot of functions describing labeling speed on raw data compared to preselected data

4.3 Accuracy

All tables shown in this section, show the training results of identification models that were trained with different datasets. The columns “Accuracy“ and “Variance“ are based on three different training runs with varying seeds. “Ident“ and “Unsure“ specify the number of images of the identifiable and the unsure class. All models that start with an “h“ are based on data collected using the annotation of randomly selected points, all models with an “s“ use the preselection module.
The presentation of the data is structured in such a way that data illustrating a specific result is consolidated in one table. Some the names appear in several tables, but these correspond

to the same set of models. Data for all models, including those not presented in detail can be found in the appendix. The model variance was very high in all experiments. This is partially because three models is a small sample and mainly because the validation set is small (5.1). A difference of a few percent in one model can be due to one or two more or less identified video tracks.

Experiment 1

In the first experiment, models trained using manually selected data were compared with models trained with data collected using pool-based active learning. The results of these experiments are shown in Tables 3 and 4.
Table 3 displays data that illustrates the susceptibility of the training process to the ratio between identifiable and unsure frames. The h_800_1k and h_800_2k models are both trained with the same core set of 800 identification images with one difference: one is trained with 1000 additional unsure frames. The model without the unsure frames and thus with the better ratio (4 to 0.67) performs four percent better and has a smaller variance. The s_99_1k and s_99_2k_p models confirm this finding. The performance of these two models is comparable, although s_99_2k was trained with more than twice the amount of data. This is probably due to its lower image ratio.

Name	Accuracy	Variance	Ident	Unsure	Ratio	Time in minutes
h_800_1k	0.837	0.017	800	200	4	133.3
h_800_2k	0.797	0.065	800	1200	0.67	133.3
s_99_1k	0.801	0.051	521	479	1.08	60
s_99_2k_p	0.804	0.035	1181	1819	0.65	83.3

Table 3: Analysis of the impact of the ratio between identification and unsure images in the training data (Ident - identifable frames | Unsure - not identifiable frames)

The core objective of this research was to make more efficient use of human time during the labeling process. For this not only quantity but also the quality of the annotated data is important. Table 4 shows training data from models, that allow conclusions to be drawn about the informativeness of the data. It can be seen that the data from the preselection strategy perform as good as manually selected data. The best illustration of this are the h_500_1k and s_99_1k models, which have a very similar ratio of identifiable to unsure frames. Furthermore, they have the same dataset size. Here, the model trained with preselected data (s_99_1k) has a higher accuracy (80.1% to 79%). There is no such one-to-one comparison for the larger models in the results. h_800_2k and s_90_2k_p are similar in the sense that the former has a better image ratio, while the latter is trained with 19% more data. It has been shown that a higher ratio and more images improve performance and should therefore cancel each other out to some extend. Again the preselected model performs better (80.4% to 79.7%). However, in both cases the differences between the models is not very significant especially regarding their individual

variances.

Name	Accuracy	Variance	Ident	Unsure	Ratio	Time in minutes
h_500_1k	0.790	0.014	500	500	1	83.3
s_99_1k	0.801	0.051	520	480	1.08	60
h_800_2k	0.797	0.065	800	1200	0.67	133.3
s_90_2k_p	0.804	0.053	952	2048	0.46	83.3

Table 4: Comparison of the quality of raw and model selected data

Experiment 2

In the second experiment semi-supervised learning techniques were tested to further improve the performance of the models and to evaluate the benefits the preselection module might have on this process. Three different approaches were compared: the use of raw pseudo labels, the manual annotation of video frames and the use of the preselection module. Their results and the result of the model from the previous experiment used to generate these pseudo labels (s_99_2kp) are listed in Table 5. All three models showed an increase in performance. The manually revised dataset had the strongest performance (87.0%), however, its performance was only slightly better than the model trained with the raw pseudo labels (86.6%). This small difference between these two models was surprising, as during the revision process around ∼30% percent of the added training data turned out to be incorrect. The performance of the models trained with the added preselection, showed much worse performance. However, in previous experiments this model provided the best performance (Table 7). In these experiments the models were trained with a lower ratio of "Identity" to "Unsure" (ratio = 1) frames. This makes it difficult to completely dismiss the effectiveness of this method. However, with this parameter set, it did not prove effective.

Name	Selection method	Accuracy	Variance	Ident + Pseudo	Unsure	Ratio	Time in minutes
s_99_2kp	Preselection module (PM)	0.804	0.035	1180 + 0	1790	0.66	83.3
i_nosort	Raw pseudo labels	0.866	0.059	1210 + 2000	803	4	83.3
i_sort	PM filtered pseudo labels	0.826	0.013	1210 + 2000	803	4	83.3
i_hand	Manually revised images	0.870	0.023	3210 + 0	803	4	83.3 + 70.8

Table 5: Result comparison of the different optimization techniques

The results of the revision process allow to generate a confusion matrix (Figure 15) for the prediction of individual frames predictions on which the i_nosort model is trained. Confusion matrices are usually quadratic, but in this case the x-axis has no unsure column. This is because only frames that were predicted as identifiable were selected for revision process. The matrix shows that the performance for columns one to three and column five is good. The performance is particularly good for columns two and five, which is not surprising as these were the collars with the brightest and therefore most recognizable colors (pink, red). The model predictions for classes four and six are significantly worse those for the other classes. For collar four in

particular, this can be partly explained by the poor visibility of the silver collar, which already mentioned earlier. These two classes were also the least occurring classes in the distribution (see 4.1). On the positive side there is hardly any confusions between identifiable classes as almost all errors are linked to the unsure class.

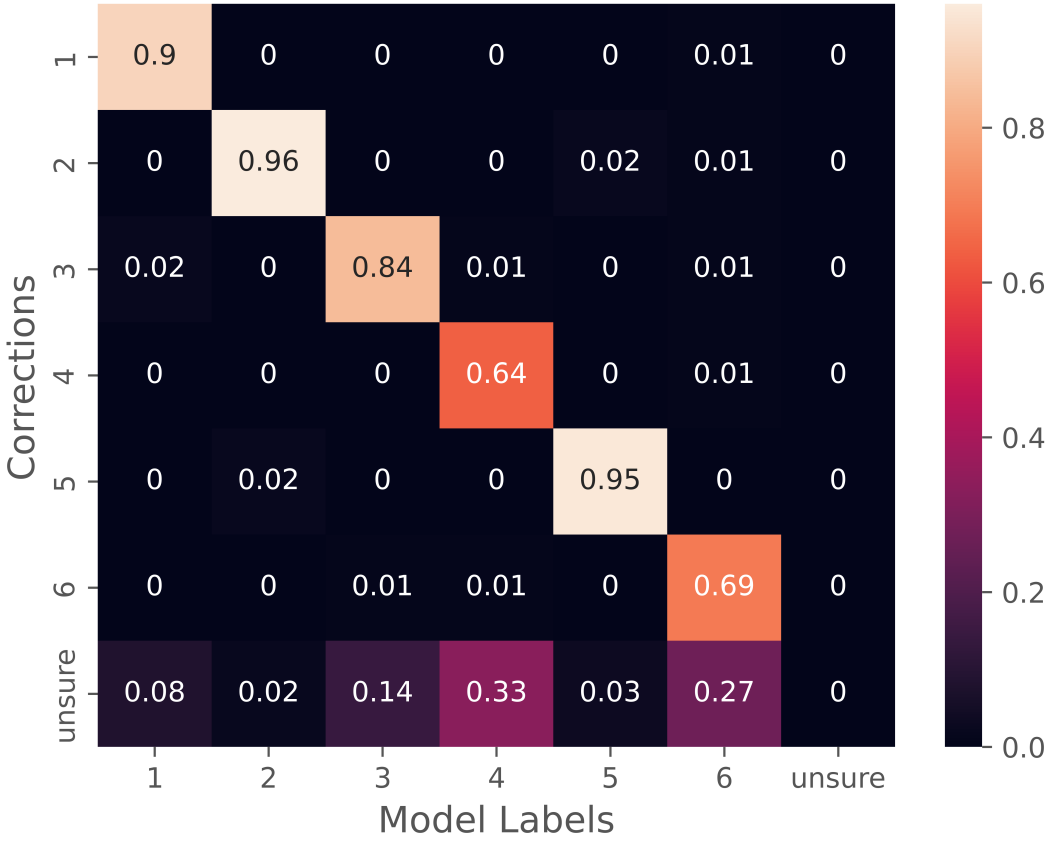

Figure 15: Confusion matrix for s_99_2k showing corrected pseudo labels by a human annotator.

Lastly the confusion matrix (Figure 16) of the best performing model on the test set is analyzed. Usually these matrices display percentages, however, in this case absolute values are displayed to also show differences in the occurrence of classes on the test set. Furthermore, this once again illustrates the small size of the validation set. In the matrix the previously identified trend of the models weakness for the fourth class is visible. Furthermore, the model seems to mix up the classes three and four. These categories correspond to the only two males in the group. They have a different fur color than the female and therefore might get mixed up more often. However, the same mix ups are not observable for the females, which indicates there might be additional reasons behind this phenomenon. The other class that was previously observed to be weak, class six, is now the only of two classes that shows no mix ups at all.

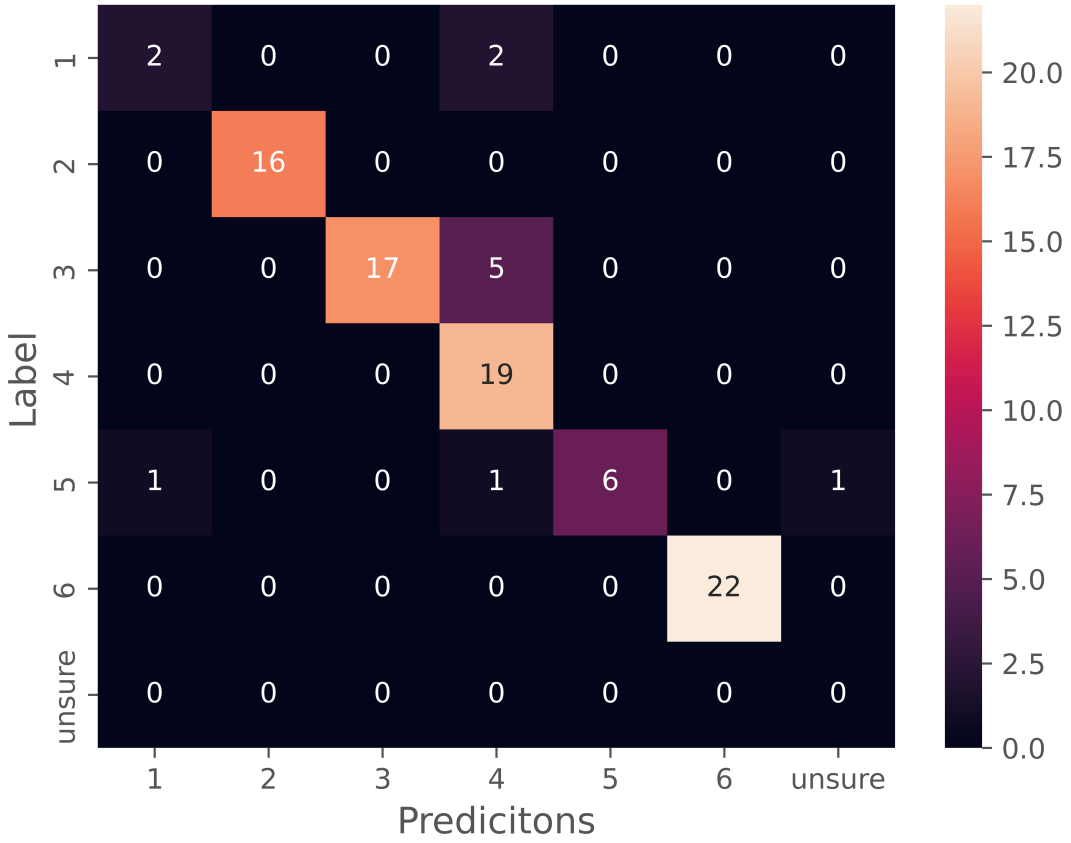

Figure 16: Confusion matrix for i_hand showing performance on test set.

Ablation study

The aim of the ablation study was to investigate the relationship between training data and model results. For this purpose, nine datasets with increasing amounts of data were examined. Three models were trained with each dataset size. The results of these models are plotted in Figure 17. The exact values of the model results are listed in the appendix. The positive relationship between training data and increased model performance is clearly visible. To further visualise this relationship, an exponential formula (ae^b) was fitted to the data. The raw results were used rather than the displayed mean values. In order to obtain results that are easier to interpret, an additional function was fitted to the points, using linear regression. For this function the first datapoint (x=250) was removed from the data as it is not reflective of the general scaling behaviour, once adequate dataset size is reached. The linear function has a gradient of 0.0002%, or 2% performance increase for every 1000 added images. As it is visible that the gradient of the exponential function is throughout the plot higher than then linear function this might be considered a "lower bound" for the expected increase in performance between 500 and 2500 images. However, the scaling behaviour might change with the specific training parameters and methods.

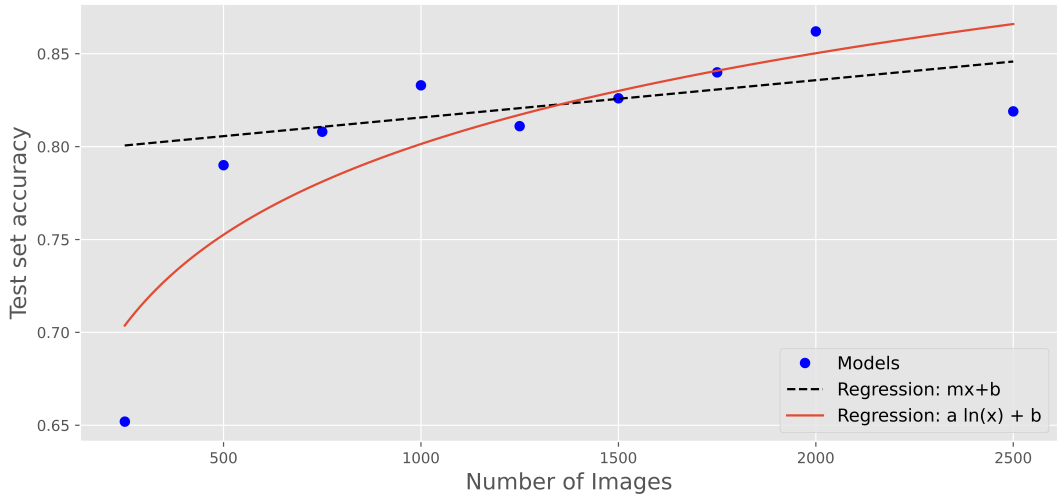

Figure 17: Visualization of test set performance versus dataset size

5 Limitations

5.1 Test set

The dataset for the current work was provided by the previous work of Richard Vogg. Unfortunately it is not very representative of the underlying data. In the current iteration, the dataset only features video tracks in which the lemurs are identifiable. However, there are also many tracks of lemurs in the videos where the identity cannot be determined. Due to this difference between test set and data, certain models might perform well on the test set, but not in their final application (the processing of the raw videos). For examples, it has been observed that an overconfidence of a model has a beneficial effect on the results. This is because it is advantageous for the model rarely predict "unsure". This aspect was not considered in Richard Vogg's earlier work and will not be addressed in this research project. Additionally the validation set is not very large, only consisting of 93 tracks. Thus large differences in model performance sometimes are cause by small numbers of correctly or falsely identified tracks.

5.2 Alternative Labeling Method

All the presented results are related to the labeling practice described in 3.2. However, this is only one specific choice of labeling method. The advantages of the chosen method are that it is not only applicable to the videos, but also to frames extracted by models. Furthermore, by labeling across the entire dataset using this method, a high diversity of images can be achieved. An alternative to labeling individual frames could be the labeling of video data. In that case,

the script would show the video with displayed bounding boxes. The labeling could be done by pausing the video, clicking on a bounding box and then pressing the corresponding key. A advantage is that the annotator can easily prevent duplicates from occurring and is likely to be able to choose higher quality frames than with other methods. However, it is unclear whether resulting labeling speed would be quicker than the chosen method. Video-based labeling might also introduce hidden difficulties: As such methods require large parts of the video to be viewed, this results in the frames being selected from a smaller data pool. This could decrease performance. Furthermore, it could even be that labeling this way results in fewer positively identifiable frames per time. This method also requires a conscious labeling of unsure frames, which is an automatic byproduct of the other method. In addition, development time of such a script will also be considerably longer.

6 Discussion

There are many potential methods for accelerating the annotation of data. Some of these methods attempt to select a more effective subset of data for annotation (active learning). This is important for the task at hand, as the annotation process with the methods used so far has been slow due to the low prevalence of identifiable data.
To increase the annotation speed, this work investigated the efficiency gains that can be achieved by preselecting the data using a classifier prior to annotation. The training process and architecture of the classifier were optimized so that only little additional development or annotation time is required.
This preselection module significantly improved the labeling performance for datasets used in practice. For a dataset of 500 identifiable images, the performance advantage was about 40 minutes. For the dataset of 3000 images, which was the largest model trained in this work the gain in time would have been about 465 minutes or 7.75 hours. These improvements were achieved by an increasing rate of images in which a collar is visible from $\sim$17% to $\sim$50%. It has been shown that these percentages can be controlled by the threshold used for the predictions of the preselection module. With one exception, the distribution of all classes resembled the distribution in the raw data. This indicates that the preselection module picks a representative and most likely quite complete sample of the data. The other class whose distribution changed was still more frequent than other classes in the dataset. The comparison of the training results with random and model-selected data showed no significant difference in data quality. These results indicate that the use of the preselection module is advisable for the given problem, as both the distribution and the data quality seem to be comparable to the data collected with the prior method. However, these results might be specific to this data and cannot be generalized to other datasets. Still the core idea of training a classifier to select data likely would work in other domains. Areas of application could be any problem were traditional labeling methods are slow due to the distribution of classes.
It was also investigated whether the preselection module can be used in subsequent training steps to further improve the model. An attempt was made to apply the preselection module to pseudo-label predictions to validate these pseudo-labels. While the data distributions suggest that this process leads to a reduction in errors, the performance is below the alternative methods of using the raw or manually revised data. Therefore, at this stage of the research, it cannot be recommended to use the preselection module in this way. However, as this method showed good results in previous trials and the distributions indicated a reduction in errors, further research in this area could lead to more beneficial results.
Unsurprisingly, the ablation study showed that the model results improved with increasing data volumes. A performance increase of about 2% per 1000 images (in the analyzed interval) was measured. These figures are confirmed by other performance measurements collected (h_800_1k vs. i_handsort - 2000 images more and 4% better). With the hand revised data,

the mean performance of the best model improved by 3% to 87% compared to the previous best model. Individual models achieved results of up to 89%. This shows that not only is the efficiency better, but that the performance also improves as a result.

Further research could investigate whether the presented preselection module can be coupled with other active learning methods such as entropy and representativeness sampling. These would not further improve the number of identifiable images that are labeled, but could select frames which are more valuable for the training process of the ID model.

Another way to improve, could also be the use of a more iterative approach. This could further increase the effectiveness and speed of the process. Pool-based active learning traditionally relies on many iterations. In this work, however, the only one large iteration was performed. By using multiple iterations, not only the results in the first phase but also at later stages could further increase.

Finally, very outedated methods were used in the semi-supervised part of this work. If current data had been paired with methods such as *Fixmatch* [22], *MixMatch* [23] or *Curriculum learning* [25], the gains in performance would have been more than the achieved $\sim$4 percent.

7 Conclusion

This research presented an alternative method to the previous approach of annotating data hand selected data. It was shown that this method utilises human time more efficiently then prior methods. This was achieved by training an preselection module that selects data based on the the visibility of an identifiable feature (collar), before it is annotated. This classifier increase the speed of annotation of the identifiable frames, but it was also shown that the distribution of images is similar to the raw data and therefore unlikely to result in significantly different identifiable frames than annotated by prior methods. This finding was confirmed by the undetectable difference in the training performances per amount of images between the two sets of data. With these increases in efficiency a model was trained, that outperformed the previous best model by 5%.

The incorporation of the preselection module in the semi supervised methods to further improve the model prediction led not to the desired results. However, the results were not fully conclusive.

Nevertheless, the results of the first experiment underline the advantages of the preselection module over the previous methods, even if it usage is limited to annotation of initial data. Furthermore, the idea of selecting data based on color visibility to speed up the annotation process could be confirmed. Further research on methods based on this idea might offer additional performance improvements over the presented method.

8 References

References

[1] David Silver et al. *Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm*. 2017.

[2] Noam Brown and Tuomas Sandholm. *Superhuman AI for heads-up no-limit poker*. 2017.

[3] Noam Brown and Tuomas Sandholm. *Superhuman AI for multiplayer poker*. 2019.

[4] OpenAI et al. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL].

[5] Royal Swedish Academy of Sciences. *The Nobel Prize in Chemistry 2024*. 2024.

[6] Royal Swedish Academy of Sciences. *The Nobel Prize in Physics 2024*. 2024.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. *ImageNet Classification with Deep Convolutional Neural Networks*. Ed. by F. Pereira et al. 2012.

[8] Jia Deng et al. *ImageNet: a Large-Scale Hierarchical Image Database*. June 2009. DOI: 10.1109/CVPR.2009.5206848.

[9] Curtis G. Northcutt, Anish Athalye, and Jonas Mueller. *Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks*. 2021.

[10] Nitin Gupta et al. *Data Quality Toolkit: Automatic assessment of data quality and remediation for machine learning datasets*. 2021. arXiv: 2108.05935 [cs.LG].

[11] Steven Euijong Whang et al. *Data Collection and Quality Challenges in Deep Learning: A Data-Centric AI Perspective*. 2022. arXiv: 2112.06409 [cs.LG].

[12] Lukas Budach et al. *The Effects of Data Quality on Machine Learning Performance*. 2022. arXiv: 2207.14529 [cs.DB].

[13] Haihua Chen, Jiangping Chen, and Junhua Ding. *Data Evaluation and Enhancement for Quality Improvement of Machine Learning*. 2021. DOI: 10.1109/TR.2021.3070863.

[14] Julia Zorthian. *Exclusive: OpenAI Used Kenyan Workers on Less Than $2 Per Hour to Make ChatGPT Less Toxic*. 2023.

[15] Richard Vogg et al. *PriMAT: A robust multi-animal tracking model for primates in the wild*. 2024. DOI: 10.1101/2024.08.21.607881.

[16] Alexander Kirillov et al. *Segment Anything*. 2023.

[17] Maxime Oquab et al. *DINOv2: Learning Robust Visual Features without Supervision*. 2024.

[18] Xu Pengcheng et al. *Small data machine learning in materials science*. 2023.

[19] Pengzhen Ren et al. *A Survey of Deep Active Learning*. 2021.

[20] Eduardo Mosqueira-Rey et al. *Human-in-the-loop machine learning: a state of the art.* 2022.

[21] Raphael Schumann and Ines Rehbein. *Active Learning via Membership Query Synthesis for Semi-Supervised Sentence Classification.* Jan. 2019. DOI: 10.18653/v1/K19-1044.

[22] Kihyuk Sohn et al. *FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence.* 2020. arXiv: 2001.07685 [cs.LG].

[23] David Berthelot et al. *MixMatch: A Holistic Approach to Semi-Supervised Learning.* 2019. arXiv: 1905.02249 [cs.LG].

[24] Antti Tarvainen and Harri Valpola. *Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results.* 2018. arXiv: 1703.01780 [cs.NE].

[25] Xinyu Yang, Tilo Burghardt, and Majid Mirmehdi. *Dynamic Curriculum Learning for Great Ape Detection in the Wild.* 2023. arXiv: 2205.00275 [cs.CV].

[26] Y. Bengio et al. *Curriculum learning.* June 2009. DOI: 10.1145/1553374.1553380.

[27] Xueying Zhan et al. *A Comparative Survey of Deep Active Learning.* 2022. arXiv: 2203.13450 [cs.LG].

[28] Ed Pizzi et al. *A Self-Supervised Descriptor for Image Copy Detection.* 2022. arXiv: 2202.10261 [cs.CV].

[29] Kaiming He et al. *Deep Residual Learning for Image Recognition.* 2015. arXiv: 1512.03385 [cs.CV].

[30] Qingyu Chen, Justin Zobel, and Karin Verspoor. *Evaluation of a Machine Learning Duplicate Detection Method for Bioinformatics Databases.* Melbourne, Australia, 2015. DOI: 10.1145/2811163.2811175.

[31] Kushankur Ghosh et al. *The class imbalance problem in deep learning.* Dec. 2022. DOI: 10.1007/s10994-022-06268-8.

9 Appendix

First Experiment

Name	Accuracy	Variance	Ident	Unsure	Ratio	Time in minutes
h_170_1k	0.391	0.263	170	830	0.2	28.2
h_300_1k	0.685	0.032	300	700	0.43	50
h_500_1k	0.79	0.014	500	500	1	83.3
h_800_1k	0.837	0.017	800	200	4	133.3
s_90_1k	0.743	0.061	393	607	0.65	60
s_99_1k	0.801	0.051	520	480	1.08	60
h_800_2k	0.797	0.065	800	1200	0.67	133.3
s_90_2k_p	0.804	0.053	952	2048	0.46	83.3
s_99_2k_p	0.804	0.035	1180	1820	0.65	83.3

Table 6: Complete overview of all models

Second Experiment

Name	Accuracy	Variance	Ratio
i_nosort	0.797	0.059	1
i_sort	0.842	0.013	1
i_hand	0.812	0.024	1

Table 7: Alternative results for experiment two.

Ablation Study

Name	Accuracy	Variance	Ratio
st_250	0.652	0.070	1
st_500	0.790	0.035	1
st_750	0.808	0.027	1
st_1000	0.833	0.027	1
st_1250	0.811	0.018	1
st_1500	0.826	0.027	1
st_1750	0.840	0.027	1
st_2000	0.862	0.010	1
st_2500	0.819	0.005	1

Table 8: Testset results of ablation study.

Other

Name	Duplication reduction	Ident annotated images (x1.7)	Collar anotated images (x1.3)	Total time (minutes)
h_170_1k	0	1000	0	1700 (28.3)
h_300_1k	0	1765	0	3000 (50)
h_500_1k	0	2941	0	5000 (83.3)
h_800_1k	0	4706	0	8000 (133.3)
s_90_1k	600	1000	1000	3600 (60)
s_99_1k	600	1000	1000	3600 (60)
h_800_2k	0	4706	0	8000 (133.3)
s_90_2k	600	2000	1000	5300 (83.3)
s_99_2k	600	2000	1000	5300 (83.3)
i_nosort	600	2000	1000	5300 (83.3)
i_sort	600	2000	1000	5300 (83.3)
i_hand	600	4000	1000	8700 (145)

Table 9: Time calculations for all methods.

Acknowledgments

I would like to express my deep appreciation for all the support, guidance, and encouragement, I received on my journey to complete this thesis.
First and foremost, I like to than my supervisor, Prof. Dr. Alexander Ecker, Professor of Data Science, head of Neural Data Science group of the University of Göttingen, for providing a very interesting research topic, valuable scientific expertise and excellent working condition.
I am especially grateful to my advisor, Richard Vogg, Phd student in Prof. Dr. Ecker's research group, whose insight and scientific expertise and guidance were key for this thesis. Thank you to him for being approachable and supportive throughout the thesis process.
Further, I would like thank the entire research group for warmly welcoming me for the time of this thesis.
I am also deeply thankful to my family and friends for their unwavering support. Special thank to my parents, for their constant encouragement and belief in me. Without them i would not be where i am today.

Thank you all for your contributions to this project, from topic selection to final pages.

Georg Eckardt