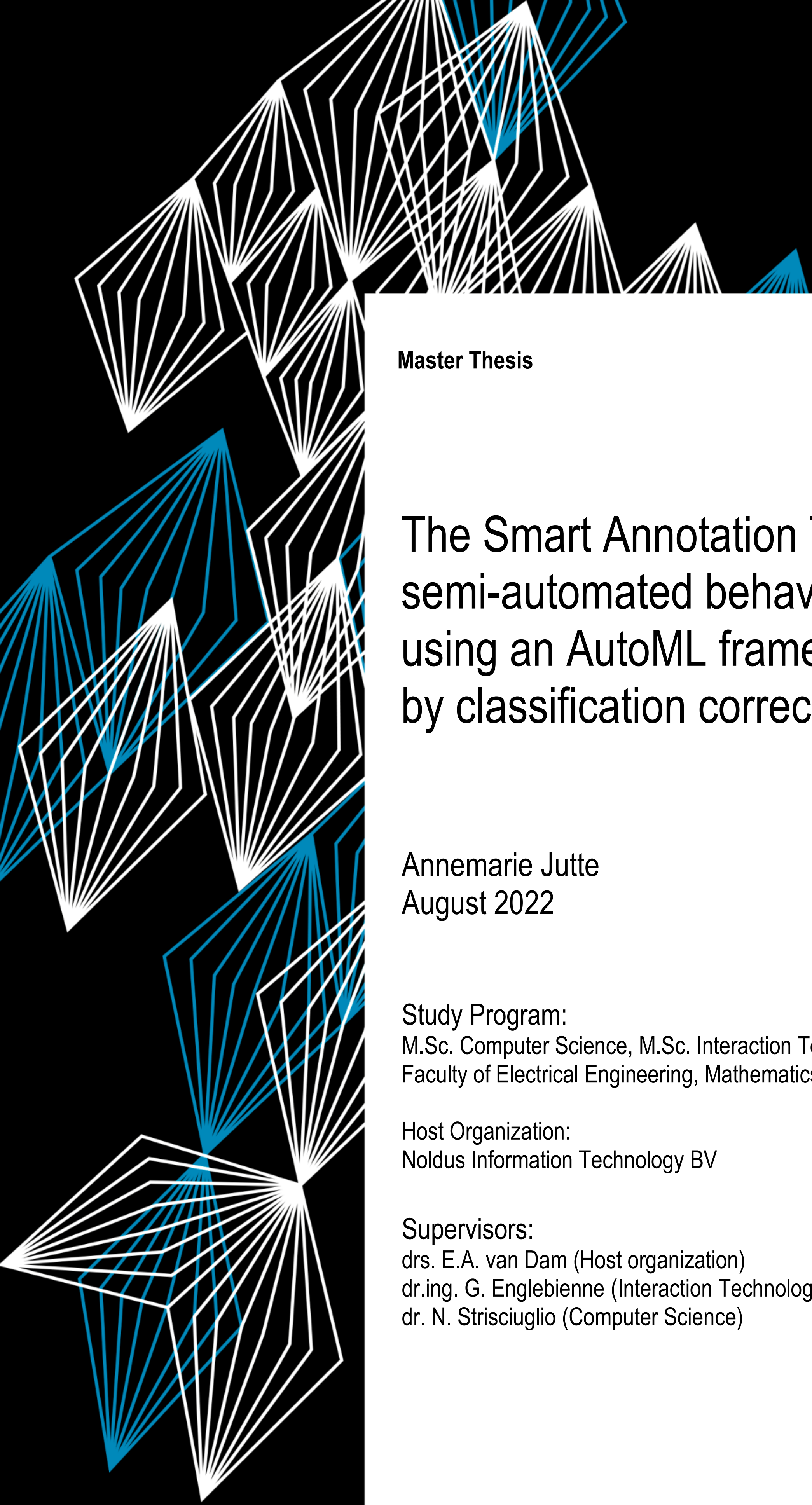


Master Thesis

The Smart Annotation Tool: optimizing semi-automated behavioural annotation using an AutoML framework supported by classification correctness prediction

Annemarie Jutte

August 2022

Study Program:
M.Sc. Computer Science, M.Sc. Interaction Technology
Faculty of Electrical Engineering, Mathematics and Computer Science

Host Organization:
Noldus Information Technology BV

Supervisors:
drs. E.A. van Dam (Host organization)
dr.ing. G. Englebienne (Interaction Technology)
dr. N. Strisciuglio (Computer Science)

UNIVERSITY OF TWENTE.

ABSTRACT

The manual annotation of behaviour is a time-consuming process,(semi-)automated methods could be employed to speed up the process. The data recorded and the behaviour classes that need to be annotated vary from task to task. This means that custom models need to be created for specific tasks. In this research, a tool is presented that can be used to quickly create annotations by optimizing human-machine interaction. With the help of the user and methods from AutoML (the field of AI that aims to automatically build machine learning systems), models for behavioural classification are trained.

The increase of efficiency reached through semi-automation may only come at a limited loss in the quality of the annotation. In this research, classification correctness prediction methods are used to control the annotation quality. Only the samples a model for classification is expected to be certain about are automatically annotated.

The tool resulting from these principles is the Smart Annotation Tool. The aim is to increase usability, compared to fully manually annotation approaches, with a limited loss of quality. Results are presented through automatic experiments on several datasets. Small-scale user studies are conducted to gather information regarding user satisfaction.

The Smart Annotation Tool comes close to creating annotations of controlled quality for some datasets. More research is required to preserve quality for all datasets. In future research, approaches that focus on obtaining more representative data should be used to increase both the quality and the efficiency of the tool.

Disclaimer: the methodology presented in this report is not to be used for non-ethical surveillance settings or the development of non-ethical AI applications. All uses should be in accordance with the legislation of the EU.

ACKNOWLEDGEMENTS

This master thesis is the result of my work over the past year. The thesis marks the final step toward the master's degrees in Computer Science and Interaction Technology. There are some people I would like to thank for their support during this past year.

Firstly, I would like to thank my supervisors: Elsbeth van Dam, Gwenn Englebienne and Nicola Strisciuglio. Elsbeth, thank you for our weekly meetings. I really appreciated your support. You were always there to discuss any issues that came up. Gwenn and Nicola, thank you for the insightful discussions and helpful suggestions.

This research was part of an internship at Noldus Information Technology. I would like to thank the people at Noldus IT, specifically Marco and Timon, for the nice work environment.

Next, I want to thank my study friends for all the great times we have had during these past six years of studying. Femke, Jarco, Jente, Lotte, Leander, Lucas, Nienke, Sven, Tessa and Wisse, thank you for some wonderful times to look back on. Here is to many more in the future!

I also want to thank my housemates for giving me a great place to wind down after studying. Thank you for all the fun evenings this past year.

Finally, I would like to thank my family for all their support.

Enjoy reading this thesis!

Annemarie Jutte
August, 2022

TABLE OF CONTENTS

DISCLAIMER

While the focus of this research is on the acceleration of creating behavioural annotation in a consensual context, the methodology could be used in any annotation context. The Smart Annotation Tool and the methodology presented in this paper are only intended to be used for ethically sound research. This means that the tool should not be used for non-ethical real-world surveillance settings. Furthermore, the tool should not be used for the development of non-ethical AI applications. The tool is meant to be used in accordance with the legislation of the EU.

1 INTRODUCTION

1.1 Background

The annotation of observational data is important for researchers in behavioural sciences. Annotations are used to draw important conclusions about the behaviour of humans and animals. Annotation tools facilitate and simplify the annotation process for researchers. Currently, most of these tools support users in manual annotation processes, users have to indicate by hand which behaviours occur at what time. This manual annotation process can be a tedious and time-consuming task. Researchers could save a lot of time if this annotation process would be (partially) automated. However, what is won in speed should not severely degrade the quality of the results. Researchers require high-quality annotations to be able to safely draw conclusions.

To automate the annotation process a machine learning model that can automatically recognize behaviours may be employed. However, a model that has been previously trained for a set of observations and behaviours may not perform well if some of the conditions change [1]. An annotation tool should ideally be useful for any set of observations (within some limits). Therefore, the tool should support the creation or adaptation of custom models.

The custom models should specifically fit the annotation task at hand. The models should be created in a way that helps reduce the time a user needs to spend annotating while having a minimum impact on the annotation quality.

To minimize the effect on annotation quality, the process does not have to be a singular effort by the tool. Instead, the tool can utilize the expert knowledge of behavioural researchers, resulting in a collaboration of machine learning and users. However, the tool should be such that the user does not need to be a machine learning expert. Therefore, techniques need to be employed such that the tool can be trained independently, without user interference.

The user involved in the process can be asked to provide information on the data, specifically on data of which the tool is uncertain. A separation between certain and uncertain data may be estimated using machine learning techniques. If this separation is made, the certain data can be annotated by the machine learning back end. This is where the gain in efficiency is won. As a result, the tool combines the strong suits of both worlds. Machine learning increases efficiency through automation, while the eye of the human experts keeps the annotation quality intact.

Noldus Information Technology is a company that supplies software and hardware solutions for high-quality behavioural research in a large variety of domains. At Noldus IT, the aim is to create such a tool that combines the expertise of machine learning and human researchers, this tool is called the Smart Annotation Tool. This research revolves around the development of this tool.

1.2 Use case analysis

At Noldus IT the following use case for the Smart Annotation Tool has been identified:

> **The annotation of a signal from scratch, using a video.**
> The user has a set of data streams and videos that they need to annotate. The user is a domain expert, for example, a neuroscientist or psychologist (in training). In the end, the user needs a high-quality annotation.

For this research, the use case is extended with the following additional requirements:

- Accelerometer and video data must be available.
- The machine learning back end may solely use accelerometer data.
- Behaviours to be annotated must be deducible from both the accelerometer and video data.

This means that only the accelerometer data is used by the machine learning back end. The user is shown video data, to make annotation possible without requiring experts who can directly recognize behaviours from accelerometer data.

Ideally, the user and the model would be presented with the same data such that there can be no mismatch between the data annotated by the user and processed by the machine learning back end. However, incorporating video data into the model leads to new challenges. Video data generally requires heavier models for proper classification, this would require bigger and slower models. Furthermore, there could be multiple people (or animals) in the video. This would raise issues with identifying the target subject. The scope of this research is limited and due to these new challenges, using video in the model will not be explored.

The requirement of the behaviour to be present in both media avoids the possibility of a mismatch between the user's expectations of the model and the actual working of the model. The task of ensuring the requirements are met lies with users of the tool.

1.3 Problem statement

Manual annotation is a time-consuming and mentally heavy task. Semi-automating this task would save researchers time so they can focus on other important aspects of their jobs. However, in behavioural research, high precision may be very important. Think for example of a situation where researchers are looking for a marker of a specific disease in a subject's behaviour. This marker should not be missed due to mistakes in the annotation. Therefore, any time reduction may only come at the cost of a limited loss of quality.

There is no one definition of usability [2], but ISO 9241-11 [3] proposes that usability should cover: effectiveness, efficiency and satisfaction. These are exactly the properties to which the Smart Annotation Tool should adhere. The system should: 1. be effective such that users can create good annotations, 2. be efficient such that users can more easily and quickly create annotations and 3. increase user satisfaction such that users prefer to use the new tool.

An issue with creating an automated behavioural classification system relates to transferability. When training a model for classification for a specific setting, it cannot be expected that this model also works in a different setting under different conditions. For each new setting, the model should be re-trained or adjusted to some extent. To create a model for a new dataset, new data is required. Information on the new data can be requested from the user. The users using the tool will generally be non-machine learning experts. Therefore, the tool can only request information on the behavioural aspects of the data. The design of the models for classification should be automated.

To increase the efficiency of the annotation process, limited information should be required from the user. Furthermore, the design should be such that models are obtained quickly. The tool should work in a way that most optimally leverages the user's time. Users should spend most of their time annotating, not waiting for a model to be trained.

However, as previously discussed, the tool should not result in a significant decrease in accuracy, compared to the manual scenario. While behavioural models for classification come in many shapes and sizes, it is unlikely that a single model for classification of sufficient quality can be created using only little labelled data, for any given dataset. For the Smart Annotation Tool to be useful it needs to be guaranteed to deliver quality for a range of datasets. Important to note is that the goal of the Smart Annotation Tool is to create a high-quality annotation, not necessarily a good model for classification.

A model that can predict the correctness of the classifications would allow for control over the annotation quality. Independently, in theory, of the actual quality of the model for classification. If it can be reliably estimated whether a sample is correctly predicted, these correct samples can be safely annotated by the machine learning back end. For the other samples, more user expertise is required. Note that in practice, the performance of the model for classification will likely influence the quality of the classification correctness prediction.

After automatically annotating the samples of which the system is certain about, the other samples should be labelled differently. Either by manually annotating them or finding another (semi-)automatic solution.

Users should be allowed to verify the automatically created annotations. They should also be able to correct any samples the tool still annotates undesirably.

1.4 Research questions

The problem statement leads to the following research question:

How can an interactive machine learning tool that assists researchers
in the annotation of behaviour improve usability, compared to fully manual annotation,
with a limited loss of quality?

For this research, the main focus will be on increasing the efficiency of the annotation process without losing effectiveness, rather than an in-depth exploration of increasing user satisfaction.

To answer this research question, several sub-questions are considered.

- RQ 1: How can machine learning be used to assist users who are not machine learning experts in the making of annotations?
  - RQ 1.1: Which model for classification can be used to classify behaviour from accelerometer data such that the annotation efficiency is high, while the classification accuracy is high enough to meet user expectations?
  - RQ 1.2: How can a model for classification be trained without the interference of

machine learning experts and what level of classification accuracy can be reached?

        - RQ 1.2.1: How can a model for classification be trained on any given dataset meeting the requirements (see Section 1.2) and what level of classification accuracy can be reached?

- RQ 2: How should the user interface of the tool work such that it does not obstruct the annotation process?
- RQ 3: How can it be ensured that while improving annotation efficiency, the quality of the annotation is preserved?
    - RQ 3.1: Can data points that are correctly classified be identified, with a precision meeting user expectations?
    - RQ 3.2: Can data points which are incorrectly classified by the model for classification be corrected to create an annotation of sufficient quality to meet user expectations?

1.5 Research methodology

This research is structured in two phases. The result of each phase is a prototype which is evaluated using automatic tests and small-scale user studies. All automatic tests are performed on multiple datasets. Ideally, the Smart Annotation Tool works on any feasible dataset that would potentially need to be annotated.

During the first phase, ‘Prototype I’ is developed by combining methods found in literature. Prototype I serves as a baseline and will be used to gather feedback to collect answers to RQ 1 and 2, besides answers already found in literature. Prototype I will not incorporate classification correctness prediction.

In phase two, RQ 3 is the focus of research. Experiments are conducted to compare methods for the identification of correctly classified data points. The result of this comparison is used to develop Prototype II. Prototype II is used for the final evaluation.

1.6 Contributions

In this research, a proof of concept of a semi-automatic tool for the behaviour of classification based on the various methods found in literature is presented. To the researcher’s knowledge, no system in literature exists that compares to the idea of the Smart Annotation Tool. The tool establishes a brand-new workflow through the combination of established methods in the state of the art. The tool presented in this research can also easily be extended to other areas, making it an interesting framework for a range of applications.

In this research, an extensive comparison of methods for the estimation of classification correctness is presented. The performance of several methods from active learning and out-of-distribution detection will be compared in a classification correctness prediction scenario. Specifically for datasets with few labelled data samples.

To summarize, this research offers the following contributions to the state of the art:

- A new tool for semi-automatic annotation unlike any found in literature.

- The design of a brand-new interactive machine learning framework utilizing neural networks, AutoML, classification correctness prediction and cascading classification.

- The design of a method where the quality of the classification can be controlled, utilizing user knowledge and uncertainty-based classification correctness prediction.

- A comparison of the performance of methods from active learning and out-of-distribution detection for classification correctness prediction on datasets with few labelled samples.

2 RELATED WORK

2.1 Classification of time-series

Time-series classification generally requires specifically dedicated classification algorithms. Time-series are sequential, the order of data points is not in an arbitrary order. To optimally utilize the data, their context should be taken into account. Examples of traditional time-series methods are Dynamic Time Warping [4] and Discrete Wavelet Transforms [5]. However, many of today's state-of-the-art machine learning methods use a form of deep learning.

2.1.1 RNNs

One such method is the Recurrent Neural Network (RNN) [6]. For RNNs, the same transformation is repeatedly applied to a series of input points. The transformation does not only use the new input, but also the output of the previous step. This allows RNNs to memorize the features it has seen before, giving RNNs the strength to deal with temporal dependencies.

RNNs are notoriously difficult to train. Backpropagation through RNNs often leads to exploding or vanishing gradients [7], causing parameters to either blow up or stagnate too early. Hochreiter and Schmidhuber [7] proposed a special kind of RNN that reduces these issues: the long short-term memory (LSTM) architecture. LSTM introduces additional connections between successive transformations to adapt the backpropagation process such that vanishing and exploding gradients occur less often.

Another type of RNN is the Gated Recurrent Unit (GRU) [8]. GRUs work comparable to LSTMs, but the number of gates is reduced. This means that GRUs contain fewer parameters that need to be trained. For the Smart Annotation Tool, the model for classification will need to be trained using a few data samples. Hence, to avoid overfitting [9] it is important to have a small model.

2.1.2 Transformers

Transformers [10] allow for parallel computation in sequential machine learning tasks, for RNNs this is not possible due to their recurrent nature. This allows for GPU support to, relatively, quickly train transformers. Furthermore, their performance is generally high. In the related area of natural language processing, the performance of transformers is the state-of-the-art [11], [12]. Transformers have also been successfully employed on motion data [13].

The issue with transformers is that they generally require the training of many parameters. As previously discussed, having many parameters requires a lot of training data. This does not make transformers a good choice for the Smart Annotation Tool.

2.1.3 (Mini)ROCKET

A type of deep learning method that can deal with context within data is the Convolutional Neural Network (CNN) [14]. CNNs are known for their high performance on tasks where the spatial or temporal context of a data sample matters. Due to the kernel transformations used, they can detect local features regardless of their global location [14].

The disadvantage of CNNs is that a high number of layers needs to be trained to obtain high performance. ROCKET (RandOm Convolutional KErnel Transform) [15] is a method which builds upon the principle of CNNs, but has been developed to reduce the training time. The key principle of ROCKET is that many diverse kernels should capture many different features of the input. The exact values used in the kernels should not matter much, as long as they are diverse enough to capture different features.

To this end, ROCKET creates a single convolutional layer by generating a large number of kernels of random hyperparameters and values. Since the kernels are randomly generated, they do not need to be trained. The maximum value and the proportion of positive values of the kernel output are used as output features. This means that, given k kernels, ROCKET only yields $2 \cdot k$ output features. These output features are used as input for a linear classifier, consisting of a single layer.

It is shown that ROCKET reaches state-of-the-art performance in a fraction of the time of its competitors on numerous datasets. Zerveas et al. [12] showed that ROCKET outperformed transformer architectures on some tasks with low dimensional data.

MiniROCKET [16] was published as an even faster and smaller version of ROCKET. Whereas ROCKET is based on the concept of random kernels, MiniROCKET is (almost) deterministic. The only random factor is the bias, which is determined based on a randomly picked data sample. The deterministic values of the parameters are based on experiments performed on 40 development datasets from the UCR archive [17]. The UCR archive is a collection of 109 widely different time-series datasets. Furthermore, where ROCKET used the maximum output value and proportion of positive values for each kernel as features, MiniROCKET uses only the proportion of positive values.

Due to its high speed without loss of accuracy, MiniROCKET will be used in the experiments for this thesis, instead of the previously discussed GRU model. While GRUs are smaller than LSTMs, MiniROCKET is even smaller. Dempster et al. [16] show that there are other methods that achieve slightly better results than MiniROCKET (HIVE-COTE/TDE [18] and TS-CHIEF [19]), however these are also more computationally expensive. Due to the user interaction in the Smart Annotation Tool, high speed is more important than a slight increase in accuracy.

(Mini)ROCKET was originally developed for univariate data series. However, a multivariate variant is also available through the Python package sktime [20]. The multivariate variant works by also including the additional dimension in the proportion of positive values.

2.1.4 Model input

ROCKET works with data segments of fixed input size. For each segment, a single label is predicted. By splitting a data stream into segments and classifying these, a data stream can be classified (in other words, annotated).

A method that can be used to split a data series into these segments is the sliding-window approach. The sliding-window approach works as the name suggests, by sliding a ‘window’, which contains a certain number of frames of the time-series, over the time-series [21]. These windows are generally overlapping, with different amounts of overlap. The advantage of this is that ‘more’ data is created. This does come at the cost of potentially biasing the dataset.

2.2 Reduction of labelling cost

In traditional machine learning, big sets of labelled data are supplied, which can be used for training a model. This approach is referred to as supervised learning. Creating such a set of labelled data is labour intensive. To reduce this effort, it can be tried to train a model using few labelled samples. If only few labels are available, the related samples do need to be representative of the dataset. The machine learning model may overfit [22] when trained on limited data points due to a lack of generalization.

Many approaches to training a model with few data samples exist that aim to mitigate this issue. Examples are: by leveraging not only the labelled data but also the unlabelled data [23], using a specialized model [24] or by making a smart selection of labels to supply. The latter option is also referred to as active learning [25] and will be discussed in the next section.

In unsupervised pre-training [23] a model is pre-trained using unlabelled data. This gives the model an understanding of the data structure. As a result, the model can be fine-tuned with less data than might be necessary for the fully supervised approach. In semi-supervised learning [26], the unlabelled data is used as support for the labelled data during training. A disadvantage of semi-supervised learning is that it slows training down since more samples need to be processed. Due to time constraints, these options will not be explored.

Few-shot learning is the field in machine learning that aims to train models for classes for which few data samples are available using prior knowledge [24]. The idea is to rely on other models or datasets to learn a structure which can be extended to the small dataset. This brings along the challenge of finding appropriate datasets to extract this prior knowledge. This challenge is considered out of the scope of this research.

The training of specifically high dimensional models, with few data samples, generally leads to overfitting [9]. Choosing a model with, relatively, few parameters is a way of enforcing generalization and allowing a reduced labelling cost. It is hypothesized that MiniROCKET will, due to its compactness, offer enough generalization to reach sufficient performance with less labelled samples.

2.3 Active learning

As previously discussed, in supervised learning, models are trained using a (big) set of preselected data. For the Smart Annotation Tool, only few samples should be labelled. Active learning is the field that investigates how to select these samples, such that high performance is reached with low labelling cost [25].

Data points that should be annotated are generally chosen based on features of the unlabelled data and a model that has been trained on the available (labelled) data [25]. The labels for these data points are queried to be labelled by the user. Using the newly supplied labels, the model can be trained or fine-tuned. An overview of such an active learning loop can be found in Figure 2.1. The Smart Annotation Tool will obtain labelled data in the same fashion.

The model can be retrained each time new samples are labelled or the same model can be fine-tuned. The latter option is referred to as online learning [27], which brings a whole range of new challenges and is considered out of the scope of this research.

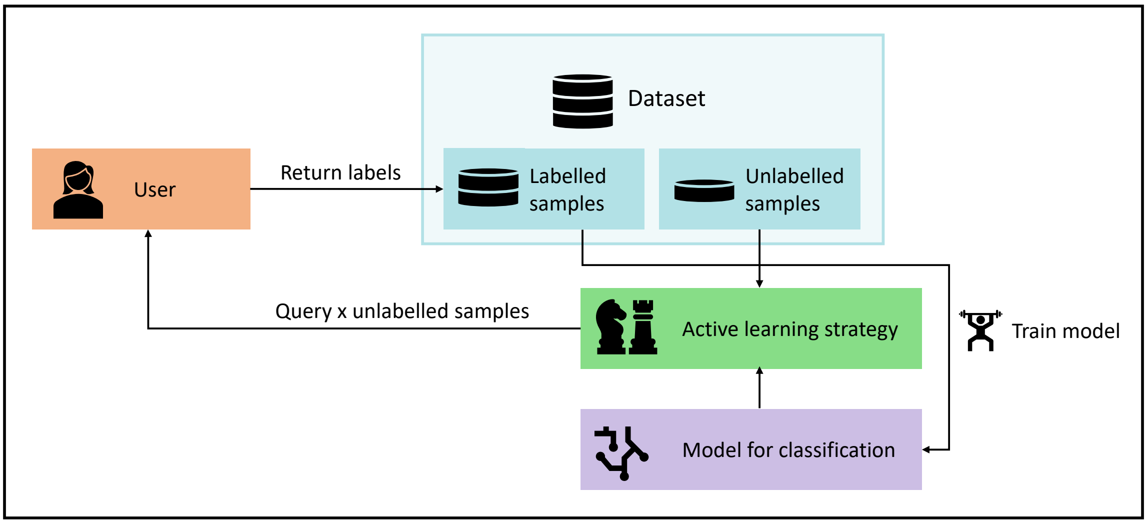


Figure 2.1: Overview of the active learning process. A model for classification is trained using the dataset of labelled and unlabelled samples. Unlabelled data samples are selected to query the user, possibly using the model output. Random sampling, as used in this research, does not use the model output.

Many active learning approaches select samples based on an estimation of their informativeness [25], this is often measured through model uncertainty. The sampling of data based on model uncertainty is referred to as uncertainty sampling [28]. Some active learning methods relating to uncertainty are explored in Section 2.6 since they closely relate to the classification correctness estimation used in this research.

Interestingly, in the literature, examples can be found where specialized active learning strategies perform not that much better than the random sampling of unlabelled samples [29]–[31]. Therefore, due to time constraints, choosing the best sampling strategy is considered out of the scope of this research and random sampling will be used.

2.4 AutoML

Once the system has received labelled data, this data can be used to train a model for classification. The selection of the proper machine learning model and hyperparameters [32] is generally crucial for reaching high, or even acceptable, performance. However, this selection requires an understanding of machine learning concepts. While the Smart Annotation Tool will get human assistance, the user will not necessarily be a machine learning expert. Automatic machine learning (AutoML) is the field in machine learning that focuses on building machine learning systems without human assistance [33].

For this research, the machine learning model will be kept the same across different datasets, see Section 2.1. However, the exact choice of hyperparameters is still important [34]. Automatically choosing hyperparameters is referred to as hyperparameter tuning [35] (also known as hyperparameter optimization).

There are many approaches to hyperparameter tuning. The general idea is to run several training trials with different hyperparameter configurations to find the best configuration. Grid search is the simplest approach, which simply tests all combinations of parameters within a search space. However, grid search is generally a slow approach [35]. Faster approaches incorporate the use of a smart sampling algorithm and/or a pruning algorithm [36]. The sampling algorithm is responsible for the hyperparameter configuration for each trial. Pruning cuts a trial short if it is unexpected to lead to good results.

2.4.1 Sampling algorithm

Examples of sampling algorithms used for hyperparameter tuning, which are faster than grid search, are: random search, Bayesian optimization, evolutionary methods, gradient descent-based methods and reinforcement learning [37]. There does not yet seem to be a single strategy that works best, which algorithm performs better varies between datasets and classifiers [38], [39]. For this research Bayesian optimization is chosen due to easily available implementations [36].

Bayesian optimization is used in situations where evaluating the objective function is costly [40], as is the case for hyperparameter optimization. Bayesian optimization evaluates points where the underlying model expects high performance or where the uncertainty of the expected performance is high. This allows for a balance between exploitation and exploration [40].

Wu et al. [41] reach comparable performance for Bayesian optimization and grid search, but Bayesian optimization reaches this performance faster. There are many different approaches to Bayesian optimization, the approach used by Wu et al. [41] is based on Gaussian processes [42]. The Sequential Model-based Algorithm Configuration (SMAC) [43] is also based on Gaussian processes. Other approaches include Tree-structured Parzen Estimators (TPE) [44] and Spearmint [45].

Eggensperger et al. [46] showed that the difference in performance between TPE, SMAC and Spearmint is not statistically different in many cases. Tsiakmaki et al. [29] use TPE for their AutoML method, based on these results. TPE is also considered a good approach for this research since it is less computationally expensive than SMAC and Spearmint.

2.4.2 Pruning

A very simple example of a pruning algorithm, is the use of the median stopping rule [47]. Median pruning stops a trial if the trial performs worse at the current epoch than the median of the previous trials. Starting after n_w warm-up trials. More advanced pruning algorithms include Successive Halving [48] and its successor Hyperband [49].

Successive Halving trains models for a fixed number of epochs n_h, evaluates their performances and only keeps the best half. While it has been shown that (the asynchronous variant of) Successive Halving can outperform median pruning [36], it requires the additional hyperparameter n_h. Hyperband extends on this idea, but inherently tunes the number of epochs n_h. This does mean that Hyperband requires a larger number of warm-up trials.

Since for the Smart Annotation Tool, the tuning process will be very short due to time constraints, it might not get out of the warm-up stage for Hyperband. Therefore, median pruning is used in this research due to its simplicity.

2.4.3 Model validation

To select the best hyperparameter configuration, model performance needs to be measured. In traditional machine learning, the data is split into a training, validation and test set. Where the validation set is used to make decisions regarding the hyperparameter configuration.

For the Smart Annotation Tool, only a small labelled set is available. This may result in potentially unreliable performance estimates. K-fold cross validation can be used to get a more reliable estimate, be it at a higher computational cost [50]. K-fold cross validation has often been successfully used in AutoML for model selection [29], [51], [52].

2.5 Semi-automated annotation

Once a model for classification has been obtained, it can theoretically be used for annotation. In literature, there are a few instances of other research where machine learning techniques are used to obtain annotations of behaviour, specifically in a joined effort with users. JABAA [53] is such a tool. Users can annotate any part of the dataset they are certain about. A classifier is then trained which best suits the existing annotation. Users can modify the predictions made by the classifier, after which the process repeats.

Lorbach et al. [30] used active learning (see Section 2.3) for the annotation of rodent behaviour. Spink et al. [31] used active learning for the annotation of horse behaviour. Both show that results close to the fully supervised scenario, where a big set of labelled data is supplied, can be reached using relatively few labelled samples. The question remains whether even the fully supervised scenario is good enough to immediately use the obtained model for annotations. There is no control over the resulting quality of the annotation.

Beugher et al. [54] presented a system for gesture analysis from video. They semi-automatically detect the location of a subject's hand. They require high performance of the detection, therefore detect uncertain samples and manually annotate these. They do this by assigning a confidence score to each automatic detection. The confidence score is based on pre-knowledge, namely the likelihood of any human subject positioning themselves that way. This allows for control over the quality of the resulting annotation. This very closely relates to the method presented in this paper. However, in this research the uncertainty is measured from the model for classification, see Section 2.6.

The big difference between the method by Beugher et al. [54] and active learning is that for active learning labelled samples are used to improve *model performance*, Beugher et al. [54] use labelled samples to improve *annotation quality*. For the Smart Annotation Tool, a good model is not the main goal, the main goal is a good annotation.

Another closely related method is DeepLabCut [55]. DeepLabCut obtains pose estimation of animals through transfer learning. Users only need to label a few samples to annotate a full dataset. Transfer learning would also be an option for this tool. However, this does require obtaining datasets that will allow the requested transfer to other datasets. This is considered out of the scope of this research.

2.6 Prediction of classification correctness

To control the quality of the annotation produced by the Smart Annotation Tool, classification correctness estimation to select correctly classified samples for annotation. Detecting incorrectly labelled samples is closely related to detecting out-of-distribution samples [56]. Both are generally based on the uncertainty estimation of models. In both cases, it is assumed that samples the model is uncertain about, correlate to samples that are either classified incorrectly or out-of-distribution. Therefore, out-of-distribution detection (OOD) methods can also be used for error detection tasks [56], [57].

Another field where uncertainty estimation is often used, is active learning, see Section 2.3.

An advantage of looking at methods in active learning is that these methods need to work for models trained with little data.

Therefore, both methods specifically designed for OOD and methods originally designed for active learning will be discussed in this section. Methods in both OOD and active learning generally work by selecting a metric that separates the data and thresholding this metric.

2.6.1 Maximum model output and entropy

Hendrycks and Gimpel [56] propose to use the softmax output of a neural network as a baseline for OOD. The softmax represents the inherently estimated uncertainty of the model. The maximum softmax output is also a metric often used in active learning [28]. A closely related method is using the entropy of the softmax output [28].

Opposing the previous methods, Gal and Ghahramani [58] have shown that while a network might have a high softmax output for a specific class, the model can still be uncertain of its prediction. This shows that while the softmax layer does simulate a probability distribution, the 'probabilities' produced do not need to correspond to actual model certainties. Softmax functions, by their exponential nature, often cause overly highly confident samples. Aigrain and Detyniecki [59] could more easily separate misclassified samples directly based on the logit output than the softmax output.

In this research, these three metrics, the maximum softmax output, the maximum logit output and the entropy will be considered.

2.6.2 ODIN

ODIN (Out-of-DIstribution detector for Neural networks) [60] is generally known as the state-of-the-art in out-of-distribution detection. ODIN improves on the baseline of the maximum softmax output through two steps: adding temperature to the softmax output and adding noise to the model input.

If a neural network is trained to classify classes $(1, 2, ..., N)$, an input is given by x and the logit output for class i is given by $f_i(x)$, then the softmax function with temperature is given by:

$$\tilde{p}_x(x;T) = \frac{\text{exp}\left(f_i(x)/T\right)}{\sum_{j=1}^{N} \text{exp}\left(f_j(x)/T\right)},$$

[60]. For $T > 0$, the temperature has a smoothening effect on the softmax output. This could potentially mitigate the extreme effects of the softmax function.

If the input data is given by x, the temperature by T, the noise magnitude by ϵ and the softmax transformation by S, then the noisy input $\hat{x}$ given by:

$$\hat{x} = x - \epsilon \cdot \text{sign}\left(-\nabla_x \log(\tilde{p}_x(x;T))\right),$$

where ∇_x can be calculated as the gradient of the model with respect to the cross entropy loss between the model output and the predicted labels [60]. The idea of this noise term is that by moving the x in the direction of the gradient, the softmax of all inputs is increased. It is shown by Liang et al. [60] that this has a bigger effect on in-distribution samples.

ODIN combines the temperature and noisy input into the metric $\tilde{p}_{\hat{x}}(\hat{x};T)$. ODIN was shown to outperform MC dropout (see the next section) on several datasets [61] for OOD. It should be noted that the datasets used by Shafaei et al. [61] are rather big and all contain images, which is rather different from the task at hand.

Not many instances of the use of ODIN for active learning can be found in literature. Chen et al. [62] compared different methods for active learning, including ODIN, for a semantic segmentation task. In their research the maximum softmax output and entropy metrics outperformed ODIN. This further raises the question of whether ODIN beats active learning methods for the 'small data' task at hand.

2.6.3 Monte Carlo Dropout

Gal et al. [63] propose using Monte Carlo dropout (MC dropout) as a measure in active learning for uncertainty estimation. Regular dropout is often used as regularization during the training of neural networks [64]. Random nodes are left out of the network ('dropped') when a sample is processed, these disruptions prevent overfitting. For MC dropout, dropout is not only used during training but also during inference.

Samples are repeatedly processed by the network, different nodes are dropped each time, giving different outputs. There are two approaches to using MC dropout in uncertainty estimation. The first category uses the different outputs as a form of ensemble learning [65]. The mean of the outputs is used, with the idea that the mean is a better estimation than a single output. The other approach is based on the idea that if the model is certain of its choice, the model should be robust to disruptions and return the same result every time (depending on the severity of the dropout). This is closely related to query by committee approaches [66], [67]. An advantage of MC dropout over both ensemble learning and query by committee is that only a single model needs to be trained.

Gal et al. [63] use the entropy as an ensemble learning-like method. It is important to note that the mean of probability distributions is again a probability distribution. After running the model R times, given $p_x^{i,r}$ as the softmax output for class i for model run t, then the MC dropout variant of the entropy is given by:

$$H[y|x] = -\sum_{i=1}^{N}\left(\frac{1}{R}\sum_{r=1}^{R} p_x^{i,r}\right)\cdot \log\left(\frac{1}{R}\sum_{r=1}^{R} p_x^{i,r}\right).$$

Gal et al. [63] also mention two methods of the other category. The first method takes the variance over the model outputs as a metric. The other approach uses the variation ratio, which is based on the occurrence of the mode in the output predictions. If f_m^x denotes the frequency of the mode in the output predictions. Then the metric is given by:

$$\text{Variation ratio} = 1 - \frac{f_m}{R}.$$

BALD [68] is a method that combines both the first and second categories. Theoretically, it is grounded in the mutual information criterion. If the mutual information is high, the model is still uncertain regarding a sample's ground truth. If y denotes the ground truth label, then the BALD metric (the mutual information) can be approximated as ([63]):

$$I[y;w] \approx H[y|x] + \frac{1}{R}\sum_{r=1}^{R}\sum_{i=1}^{N} p_i^{x,r}\cdot\log\left(p_i^{x,r}\right).$$

As can be seen, it uses both the entropy over the mean model outputs and the mean entropy over individual runs. For certain samples, the difference between these should be low since predictions should be consistent, while the entropy over the mean model outputs should be high

2.7 Cascading classification

Cascading (or hierarchical) methods can be used to split a difficult problem into simpler subproblems [69]. Heitz et al. [70] used cascading classifiers to improve classification accuracy by giving each classifier the output of the previous classifier as input, in addition to the original model input. Wang et al. [71] use a cascading model where increasingly complex models are applied to samples if the previous model was not certain enough on their prediction.

This idea will be briefly explored in this research. Once a model has been trained, the samples it can confidently annotate will be identified. For the classification of the other samples, it could be an option to train a different model that will focus on the leftover samples. This would automatically divide the classification task into simpler subproblems.

2.8 Interactive machine learning

Interactive Machine Learning (I-ML) is the field which aims to combine the strengths of human and machine [72]. The machine learning system is there to support the human, while the human is there to teach the machine learning system [73].

Traditional machine learning algorithms do not take the people providing the information to train models into account. This lack of user consideration is something that is also still prevalent in AutoML and active learning (AL), even though the purposes of these fields revolve around users. Little is known about how users interact with AutoML and active learning algorithms [74].

Some research has been done into I-ML in general. Dudley et al. [72] argue that the system task goals and constraints should be made explicit to users. Horvitz et al. [75] argue that it is important to let the user know what the system can do and how well the system can do what it does. This is further supported by Dudley et al. [72] indicating the importance of supporting user understanding of model uncertainty and confidence.

People value transparency in learning systems [76]. This is illustrated by the case of Kulesza et al. [77]. They gave users a tutorial on the workings of the system and found that user satisfaction increased.

During the training process, the user could or should have the option to inspect the current state of the model. Dudley et al. [72] specify that this model inspection step can only include metrics, but can also include a visual inspection of the model performance. Dudley et al. [72] suggest showing metrics such as coverage and confidence.

2.9 Conclusions

From the literature, some preliminary answers can be found to the research questions, see Section 1.4.

2.9.1 Model for behavioural classification (RQ 1.1)

Firstly, Mini-ROCKET is a method specifically designed for reaching high performance in little training time. To fully answer RQ 1.1, it needs to be tested whether MiniROCKET can reach high accuracy with little data.

2.9.2 Training of a model without human interference (RQ 1.2)

Several AutoML approaches exist to allow the training of machine learning models without the supervision of a machine learning expert. Using hyperparameter tuning, these approaches should theoretically be able to train models for classification for any dataset, partially answering RQ 1.2.1. The level of performance that can be reached should be explored experimentally to fully answer RQ 1.2 and RQ 1.2.1.

2.9.3 User interface (RQ 2)

In literature, discussions have been published regarding the best way to approach the user-side of interactive machine learning approaches. It is important to properly inform the user of the workings, limitations and progress of the machine learning back end. This partially answers RQ 2, Further answers for RQ 2 will be explored in the pilot user study for Prototype I.

2.9.4 Preserving effectiveness (RQ 3)

While some efforts have been made towards (semi-)automated annotation, most of these efforts do not allow the annotation quality to be controlled. Only the method by Beugher et al. was found [54]. They control the quality of their annotation using uncertainty estimation techniques, only annotating the most certain samples. This offers an answer to RQ 3.

2.9.5 Prediction of classification correctness (RQ 3.1)

Beugher et al. [54] base the certainty estimation of their data on pre-knowledge. Pre-knowledge may not be available for the general datasets that are to be annotated with the Smart Annotation Tool. However, various uncertainty estimation techniques exist that can be used for neural networks. Whether these work sufficiently well within the SAT will be explored for Prototype II.

2.9.6 Correcting misclassified points (RQ 3.2)

Cascading classification has been used in a few instances to simplify complicated machine learning problems. Whether they can be used to create a full annotation for a desired level of quality will be further explored using Prototype II.

2.9.7 Overall tool

No methods have been found which combine all the discussed concepts (annotation of behaviour, interactive machine learning, AutoML, classification correctness prediction and cascading classification). Therefore, whether such a tool will succeed is still an open question.

3 DATASETS

The Smart Annotation Tool should work for any dataset meeting the requirements (see Section 1.2). Therefore, to evaluate the quality of the tool, results should be presented on different datasets. A quick overview of the datasets used and their properties can be found in Table 3.1. The distribution of the data over the different classes can be found in Appendix A.

The number of samples, as reported in Table 3.1, results from the sliding-window segmentation with a window size of 2 seconds, as discussed in Section 4.1.1. Only data samples for which a ground-truth label is available from the original dataset are used throughout this research. Therefore, due to varying amounts of annotations being available for different datasets, the number of samples does not directly correlate with the dataset duration.

Data from different accelerometers is available for the datasets. Either all sensors or a single sensor is used during the experiments. This choice is based on the performance of Prototype I. This will be further discussed in Section 4.3.1.

Table 3.1: Overview of the datasets used in this research and their properties. Note, the sensor location denotes the sensor used when only one sensor is used.

Dataset	Type of activity	Duration	Number of samples	Number of classes	Number of subjects	Frequency	Sensor location
BAfitness [78]	Fitness exercises	20 min	2.768	6	1	64 Hz	Right knee
HCIgestures [78]	Arm gestures	6 min	1.134	5	1	98 Hz	Right wrist
OPPORTUNITY AR [79]	Locomotion	7 hours	6.375	4	4	30 Hz	Hip
Skoda [80]	Repairs	2 hours	16.533	10	1	98 Hz	Left wrist
Daphnet FoG [81]	Freezing of gait	8 hours	37.954	2	10	64 Hz	Ankle
Cow behaviour	Locomotion	19 hours	24.840	3	2	20 Hz	Neck

3.1 BAfitness dataset

The BodyAttack Fitness (BAfitness) dataset [78] contains the accelerometer data of six different activities performed by a single subject. The subject has ten accelerometer sensors placed on their left leg. The activities performed are: ‘Superman jumps’, ‘Knee lifts’, ‘Jumping jacks’, ‘High knee runs’, ‘Flick kicks’ and ‘Feet back runs’. The subject performs each of the activities sequentially over five runs. The total recording is about 20 minutes, where each activity is performed for a comparable amount of time.

3.2 HCIgestures dataset

The purpose of the HCIgestures dataset [78] is to recognize different arm gestures in an HCI scenario. The authors decided to reduce the variability in the movements by making the subject move their arm along a physical object [78].

All data is based on a single subject who performs five gestures over 60 runs. These gestures are of the following shapes: ‘Triangle (pointing up)’, ‘Square’, ‘Circle’, ‘Infinity’ and ‘Triangle (pointing down)’. Data is measured by 8 accelerometers attached to different places on the subject’s right arm.

3.3 OPPORTUNITY Activity Recognition dataset

The OPPORTUNITY Activity Recognition dataset [79] is a dataset containing sensor data from a range of activities performed by human subjects. The dataset contains data from 12 accelerometers. Data points are missing at different times for different sensors. To preserve data, only three sensors are considered, namely, the sensors attached to the subjects’ right knee, hip and left upper arm.

The data of four users over six runs has been made available. Runs last between 12 and 40 minutes. There are two levels of annotation, locomotion and finer-grained annotation. Only the locomotion data is used. For the locomotion annotation the classes ‘Walk’, ‘Stand’, ‘Sit’ and ‘Lie’ are available.

The originally published dataset did not contain video data, however recently the OPPORTUNITY++ dataset [82] has been published which extends the original dataset with video recording and subject tracking. The combination of available accelerometer and video data allows for the OPPORTUNITY datasets to be used in user studies.

3.4 Skoda Mini Checkpoint dataset

The Skoda Mini Checkpoint dataset [80] (from now on shortened as the ‘Skoda’ dataset) contains the data of a subject performing car maintenance. It contains the data of a single subject performing ten types of activities, for the exact activities see Appendix A. Data and annotations are separately available for the left and right arms of the subject. For this research, the data of the left arm is used, since more data is available. Data is available from ten accelerometers attached to various places on the subject’s arm.

3.5 Daphnet Freezing of Gait dataset

The Daphnet Freezing of Gait (Daphnet FoG) dataset [83] is a dataset on subjects with Parkinson’s disease. It was created as a benchmark for methods that automatically recognize gait freeze, for more on the freezing of gait see for example [81]. The dataset was recorded in a setting that was set up to generate freeze events in the subjects. The recorded data is annotated into two classes: ‘No freezing’ and ‘Freezing’.

Data of ten subjects was recorded. For some subjects, the data was recorded in one run, for others in two runs due to either technical difficulties or users needing to take breaks. Accelerometer data is recorded on the subjects’ left ankle, left knee and trunk.

3.6 Cow behaviour dataset

The Cow behaviour dataset has been developed at Noldus IT. The data contains observations of two cows in an indoor barn. The data has been annotated both for locomotion and eating behaviour. For this research, only the locomotion annotations are used. The locomotion behaviours observed are: ‘Stand’, ‘Walk’ and ‘Lie’. Data is recorded by a single accelerometer attached to the cows’ necks.

4 PROTOTYPE I: THE BASELINE

In this section, the baseline prototype is presented and discussed. The baseline serves as an initial proof of concept and to answer RQ 1 and RQ 2, see Section 1.4.

The baseline is evaluated through automatic tests and a small-scale user study. The purposes of Prototype I are as follows, the related research questions are referred to between brackets:

1. Verify the baseline system can train a model for behavioural classification for various datasets (RQ 1.1, RQ 1.2, RQ 1.2.1).
2. Verify that the performance can be accurately estimated (RQ 1.2.1, RQ 2).
3. Verify the AutoML system also works with a human user (RQ 1).
4. Identify any obstructive issues with the user interface (RQ 2).
5. Streamline the final user study for Prototype II.

The first two purposes are to be reached through the automatic tests, the others through the pilot user study.

In this section, the method used for Prototype I will first be discussed, followed by the experimental setup and the results of these experiments. The conclusions can be found in Section 4.5.

4.1 Method

4.1.1 Data preparation

To create an annotation, the Smart Annotation Tool needs data. The Smart Annotation Tool needs video data and accelerometer data. Furthermore, the offset between data streams and a set of predefined behaviours is requested. These predefined behaviours are a fixed set of labels users can choose from while annotating. Making this set variable is considered out of the scope of this research since this would require the same variability from the machine learning models.

After the data has been uploaded, the video and accelerometer data are segmented. The result of this segmentation is a set of ‘data samples’ consisting of a video clip and accelerometer segments. The accelerometer segments are transformed using MiniROCKET, see Section 2.1.

For the segmentation, sliding-window segmentation is used, see Section 2.1.4. For the video clips, a window size of 2 seconds is used. During the pilot user study, it should be verified whether this is an acceptable choice. For the accelerometer data, the window size is considered a variable hyperparameter, as will be discussed in Section 4.1.5. The segments are created immediately when the user uploads their data, such that the tool is fast at run time.

MiniROCKET is also applied to the accelerometer segments during preparation. The number of kernels for MiniROCKET is considered a hyperparameter. Therefore, the accelerometer segments of different durations are transformed by MiniROCKET with varying numbers of kernels, as will be discussed in Section 4.1.5. Dempster et al. [16] show that due to the choice of the kernel weights, it is not necessary to normalize the data before applying MiniROCKET.

For each segment, the ground truth behaviour is seen as the behaviour occurring at the (temporal) middle. The segments, of other durations than 2 seconds, are created such that the middle always corresponds to the middle of a window of 2 seconds. This makes it easy to transfer labels. If a window reaches into non-existing data, the sample is dropped only for that duration.

4.1.2 System overview

The goal of the initial prototype is to train a model for behavioural classification (MBC) in cooperation with the user. The MBC is trained using labels specified by the user. Users will annotate ('label') samples based on the video clips, while the model is trained using the accelerometer segments.

Users are repeatedly asked to label batches of x samples. Random sampling is used for selecting these samples, see Section 2.3. After a user has labelled a batch of samples, these samples are used to train the MBC. The training of the MBC includes the tuning of hyperparameters. The performance of the MBC is estimated using k-fold validation. While the estimated performance has not converged, users are repeatedly asked to label more data samples. This repeated process is referred to as the active learning loop.

When the performance has converged, the final MBC is trained. The final model is used to annotate the unlabelled samples. An overview of the full 'annotation system' can be found in Figure 4.1.

4.1.3 Model for behavioural classification

From the literature study, see Section 2.1, it was found that the MiniROCKET classifier [16] is the most suitable model for the aims of this project. While MiniROCKET is originally developed for univariate classification, a multivariate version of MiniROCKET is available [20]. Besides MiniROCKET, a classifier is needed to map the MiniROCKET transformation to output classes. For this, a fully connected layer with a softmax activation is used.

For the optimization of the fully connected layer, the Adam optimizer [84] is used with L2 regularization. The model is optimized based on the cross entropy loss.

In general, the Smart Annotation Tool will need to deal with imbalanced data. To mitigate issues that arise from training with imbalanced data during training, the data is resampled during model optimization. For each epoch, the data is resampled by weighting data samples based on the occurrence of their ground truth class. The number of data points sampled during an epoch equals the size of the dataset. This means that not all samples may occur during an epoch, but different samples may occur during different epochs.

During each iteration, models are trained from scratch, rather than continuing training with the newly added samples. As previously discussed, online learning is considered out of the scope of this research.

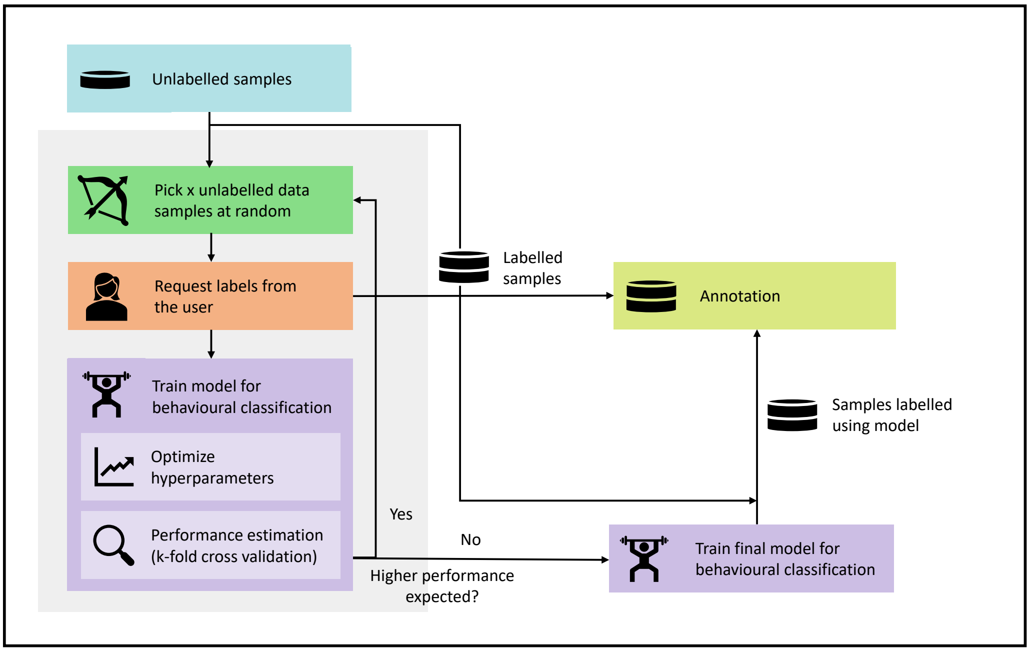

Unlabelled samples
Pick x unlabelled data samples at random
Request labels from the user
Train model for behavioural classification
Optimize hyperparameters
Performance estimation (k-fold cross validation)
Labelled samples
Annotation
Samples labelled using model
Yes
No
Higher performance expected?
Train final model for behavioural classification

Figure 4.1: Overview of the annotation system for Prototype I. The user is requested to label samples. These samples are used for hyperparameter tuning. This process is repeated until the performance has converged. A final model is trained which is used to create the annotation.

4.1.4 Data sampling

As previously discussed, during each iteration x random unlabelled data samples are selected to be labelled by the user. The resulting labelled dataset is used for training and validation purposes. In most machine learning research, where a model is being developed, data samples in the training, validation and test sets should be independent. Meaning the samples selected to be labelled should be independent somehow from samples used for evaluation purposes. However, the aim is to develop a model, the model should also work on unseen data. For the Smart Annotation Tool, the goal is to create an annotation for the unlabelled data, supported by the model.

As a result, when testing the performance of the tool, the tool should be evaluated on all unlabelled data and independence is not required. The performance estimated using the validation data should match the performance of the test data. This means that independence is also not required for the validation data.

4.1.5 AutoML

Using AutoML techniques found in literature, the model for behavioural classification is optimized without needing user interference. To properly train the model, hyperparameter optimization is used. The objective of the optimization is the weighted model accuracy as estimated by k-fold cross validation.

Validation

The performance of the MBC is considered as the weighted accuracy, if C is the set of classes in the dataset, n_c^c is the number of correctly classified samples of that class and n_c is the total number of samples in that class, then:

$$\text{weighted accuracy} = \frac{1}{|C|} \sum_{c \in C} \frac{n_c^c}{n_c}. \tag{4.1}$$

If $n_c = 0$, the associated term in the summation is set to 0.

To estimate the weighted accuracy, k-fold cross validation is used. k models are trained using k optimizers on k splits of the dataset. Each split forms a training and a validation set. The splits are chosen at random with the condition that all the data is split over the k validation sets and the sets are disjoined.

At a given epoch i, the model performance is estimated as the mean of the weighted accuracy over the k models, each trained up to epoch i. The standard deviation over the weighted accuracies can be used as a measure of confidence for the performance estimate. This estimate is used as feedback to the user, see Section 4.1.6.

Hyperparameters

The system contains several hyperparameters, including:

- Window size of sliding-window
- Step size of sliding-window
- Number of MiniROCKET kernels
- Learning rate
- Weight decay
- Batch size
- Number of labelled samples
- Number of training epochs
- Number of tuning trials
- Number of folds in the cross validation

There are some more hyperparameters used by the Adam optimizer [84], but these will not be considered.

There are three ways hyperparameters will be dealt with, they will be: fixed, tuned or estimated based on convergence. The number of folds in the cross validation and the step size of the sliding-window segmentation will be fixed. The number of folds in the cross validation cannot be tuned like the other hyperparameters, since it explicitly influences the performance estimation. Due to the implementation of the data preparation, as briefly discussed in Section 4.1.1, it is difficult to change the step size of the sliding window segmentation.

Tuning: Even though Dempster et al. [16] suggest always using 9996 kernels for MiniROCKET, it is still considered a tunable hyperparameter. For the Smart Annotation Tool, very little data may be available, in which case training 9996 parameters could lead to overfitting.

For the hyperparameter optimization, the Tree-structured Parzen estimation algorithm [44] is used for sampling. The TPE sampler is supported by a median pruner [47].

Convergence: The number of training epochs, tuning trials and labelled samples are estimated online based on the convergence of the estimated MBC performance. This performance is, again, the weighted accuracy as estimated using the k-fold cross validation.

The number of training epochs is estimated separately for each trial of the hyperparameter tuning. For each trial, the k cross validation models are trained until the performance, e.g. the average validation performance over all k models has converged. The models are considered 'converged' if the last improvement in the performance was n_e epochs ago.

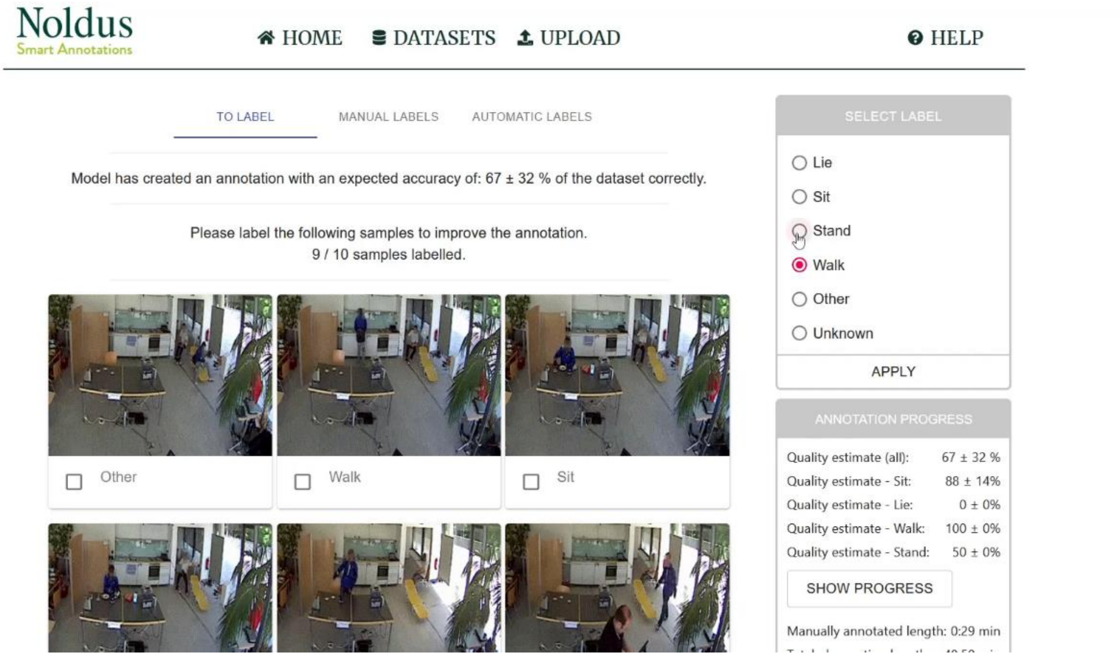
The performance is measured only every m_e epoch, to speed up training. To avoid situations where the tuning takes too long, training proceeds for a maximum of N_e epochs. For the training of the final MBC, the number of epochs for the best performing hyperparameter tuning trial is used.

The number of trials used for hyperparameter tuning is increased until no improvement has been found during the last n_t trials. There is a maximum of N_t, again to limit run time.

Samples are labelled by the user in batches of x samples. After each batch, the MBC is updated. If the last n_l active learning iterations did not find an improvement in the estimated performance, the iteration ends. A maximum of N_l samples are labelled since training time increases as more samples are labelled.

4.1.6 User interface

The user interface is adapted from an early prototype developed at Noldus IT. This prototype allows the labelling of video clips in batches. During each active learning iteration, x samples need to be labelled. These samples are shown to the user in a grid-wise fashion, see Figure 4.2. Videos of the same behaviour class can be selected and labelled using the 'Select Label' panel. The behaviour can be applied to the videos by pressing the 'Apply' button.



Model has created an annotation with an expected accuracy of: 67 ± 32 % of the dataset correctly.

Please label the following samples to improve the annotation.
9 / 10 samples labelled.

Quality estimate (all): 67 ± 32 %
Quality estimate - Sit: 88 ± 14%
Quality estimate - Lie: 0 ± 0%
Quality estimate - Walk: 100 ± 0%
Quality estimate - Stand: 50 ± 0%

Manually annotated length: 0:29 min

Figure 4.2: User interface for Prototype I. The videos are from the OPPORTUNITY++ dataset [82][1].

As discussed in Section 2.8, Dudley et al. [72] argue that it is important to allow users to inspect the model. They mention allowing the user to inspect the model by showing metrics regarding coverage and confidence and by allowing visual inspection.

[1] www.creativecommons.org/licenses/by/4.0

Metrics regarding coverage and confidence are shown in the ‘Annotation Progress’ panel. The k-fold cross validation estimation of the model quality and the corresponding standard deviation, as discussed in Section 4.1.5, are displayed. These statistics are also shown per class. Furthermore, how much of the dataset has been annotated by hand is displayed. The estimated overall quality of the MBC is also displayed on top of the page, such that this is emphasized.

There are three tabs available, the first tab ‘To Label’ is where users find the x samples of the current active learning iteration that they need to label. In the ‘Manual Labels’ tab, they can find the samples they have labelled by hand. In the ‘Automatic Labels’ tab, the predictions of the MBC trained so far can be found. In the ‘Automatic Labels’ and ‘Manual Labels’ tabs, video clips are grouped by behaviour. The ‘Automatic Labels’ tab can be used by the user for visual inspection of the performance. For Prototype I, incorrectly labelled video clips cannot be corrected yet.

4.1.7 Evaluation

For the system, the labelled samples are used as training and validation data. The system should be able to annotate the unlabelled data. The performance of the system is evaluated by the performance of the MBC on the unlabelled samples. This means that the unlabelled samples are considered to be the ‘test data’. Results will be reported on this test data in the form of the weighted classification accuracy, see Equation 4.1. The number of labelled samples it took to reach the performance of the MBC, is also considered part of the system performance.

4.2 Experimental setup

The machine learning back end is implemented in Python [85], supported by PyTorch [86]. The user interface is built upon a pre-existing prototype of Noldus IT. The user interface is a web-based application using a combination of Javascript [87], C# [88] and SQLite [89]. The pilot user study is run on an Intel Xeon Processor E5-1620 v3, with an NVIDIA GeForce GTX 1060 as GPU support.

4.2.1 Hyperparameters

The number of folds for the cross validation is set to $k = 5$, to balance speed and quality. The step size of the sliding-window segmentation is set at 1 second. The search space used by the TPE sampler during the hyperparameter tuning can be found in Table 4.1. Parameters are either sampled from a range or a set. The window size and number of kernels are treated as a set, see Section 4.1.1. Certain hyperparameters are sampled from a logarithmic scale, this can be found in the Table. For the implementation of the hyperparameter tuning, Optuna’s implementations of TPE and median pruning are used [36]. The median pruner is given a warm-up $n_w =$ 5 epochs.

For the number of epochs, training is stopped when the last improvement in performance was $n_e = 20$ epochs ago, with a maximum of $N_e = 150$ epochs, where the performance is measured every $m_e = 5$ epochs. For the number of trials, tuning is stopped when the last improvement was $n_t = 10$ trials ago, with a maximum of $N_t = 50$ trials. Finally, for the number of labelled samples, the active learning iteration is stopped when the last improvement in performance was $n_l = 3$ iterations ago, with a maximum of $N_l = 20$ iterations.

Finally, for the automatic experiments, the number of labelled samples gathered from the user is $x = 20$. For the user study $x = 10$ is used, to keep the samples manageable.

Table 4.1: An overview of the hyperparameters used for Prototype I. A hyperparameter is either a fixed value or tuned, in which case the range or set it is sampled from is displayed.

Hyperparameter	Type	Value/range/set
Step size (in seconds)	Fixed value	2
Number of folds	Fixed value	5
Learning rate	Logarithmic range	[1e-5, 1e-2]
Weight decay	Logarithmic range	[1e-5, 1e-2]
Window size (in seconds)	Set	[2, 5, 8]
Batch size	Range	[10, 100]
Number of kernels	Set	[500, 1000, 5000, 10000]

4.2.2 Automatic experiments

For the automatic experiments, the aim is to show the performance of Prototype I on each of the datasets, see Section 3. It is possible to select which sensors to use. For the HCIgestures and OPPORTUNITY AR datasets multiple sensors are used, for the others, a single sensor is used. This choice is based on early experiments, see Appendix B. In general, the differences were slight, except for the BAfitness dataset

The performance of the system is measured in terms of the weighted accuracy of the MBC on the data that has not yet been labelled (the *test accuracy*) and the number of labelled samples used to reach this accuracy. Furthermore, the mean absolute distance (MAD) between the weighted accuracy and the accuracy estimated by the k-fold cross validation (the *validation accuracy*) is measured to evaluate how well the cross validation estimates the test accuracy.

4.2.3 Pilot user study

A pilot user study is held to gather feedback on the baseline prototype. During the user study, users are observed while using the tool. To get more insight into how the users respond to using the tool, a think-aloud protocol will be employed.

The prototype testing will be followed up with a short interview. Participants are asked a series of questions. The questions concern the higher-level concept of the Smart Annotation Tool, but also more specific questions concerning the user interface, video clip duration and annotation time. The interview questions can be found in Appendix C.5.

Ethical approval for this user study has been granted by the Ethics Committee Computer & Information Science at the University of Twente under reference number RP 2022-14. The information brochure and informed consent form used can be found in Appendices C.2 and C.3.

The pilot user study took place in an early stage of the research. Afterwards, some small adjustments were made to the machine learning back end. These small adjustments should not have a big influence on the results. The changes can be found in Appendix C.1. Furthermore, it was found that the software did not cut out missing data. As a result, the beginning or end of some data samples contained measurements from earlier or farther along in the accelerometer signals. This means that the performance of the numeric results might be lower than the actual performance since this makes things more difficult for the machine learning back end.

User instructions

Participants are instructed to label 5 sets of 10 video clips. After each set, the machine learning back end is updated. In case multiple behaviours occur within a single video clip, participants are told to label the clip as the behaviour that happens in the temporal middle. If participants encounter a behaviour they cannot assign to any of the given behaviours, they are told to choose the ‘Other’ label. If they encounter a video where due to visual obstructions they cannot determine the behaviour, they are told to choose the ‘Unknown’ label.

Users are also given a short explanation as to how the tool works. Namely that the ‘AI’ will ask for labels and use these to get better at recognizing these behaviours. Furthermore, they are told that the ‘AI’ learns to estimate how well it is performing and that they will be shown this. The full user instructions can be found in Appendix C.4.

Metrics prototype test

To verify that the pipeline works, the model trained by the user during the prototype testing is tested on the ground truth data. It is tested by calculating the overall accuracy of the model on the unlabelled data. This accuracy is referred to as the test accuracy. Measuring the accuracy this way requires the user-made annotations to agree with the ground truth. To verify this, the rater-reliability is calculated between the user labels and the ground truth labels. For this Cohen’s kappa [90] is used. McHugh et al. [91] suggest that a Cohen’s kappa between 0.60-0.79 denotes moderate agreement, 0.80-0.90 denotes strong agreement and above 0.90 denotes almost perfect agreement.

Participants

Participants are chosen amongst employees at Noldus IT, for convenience reasons. Participants should have experience with behavioural annotation. Due to time constraints, a sample size of three participants is used.

Dataset

For the user study, the OPPORTUNITY++ dataset is used, due to the availability of video data and the relatively simple behaviours, see Section 3.3. The full OPPORTUNITY++ dataset contains about 7 hours of data. The prototype does not yet handle this much data. Therefore only a subset is used. The subset that is used contains 1 ADL (activities for daily living) run for 4 subjects. For the respective participants (1, 2, 3 and 5), runs 4, 2, 5 and 2 are randomly selected. The subset has a duration of about 50 minutes. For the pilot user study, only the single accelerometer on the subjects’ hip is used.

The user study should not be too time-consuming for the participants. Therefore, having enough time during the interview is prioritized over obtaining a large number of labels. Instead of letting users label until the performance has converged, a fixed number of 50 video clips will be labelled by the users. These clips are randomly generated but are the same for each user.

4.3 Results

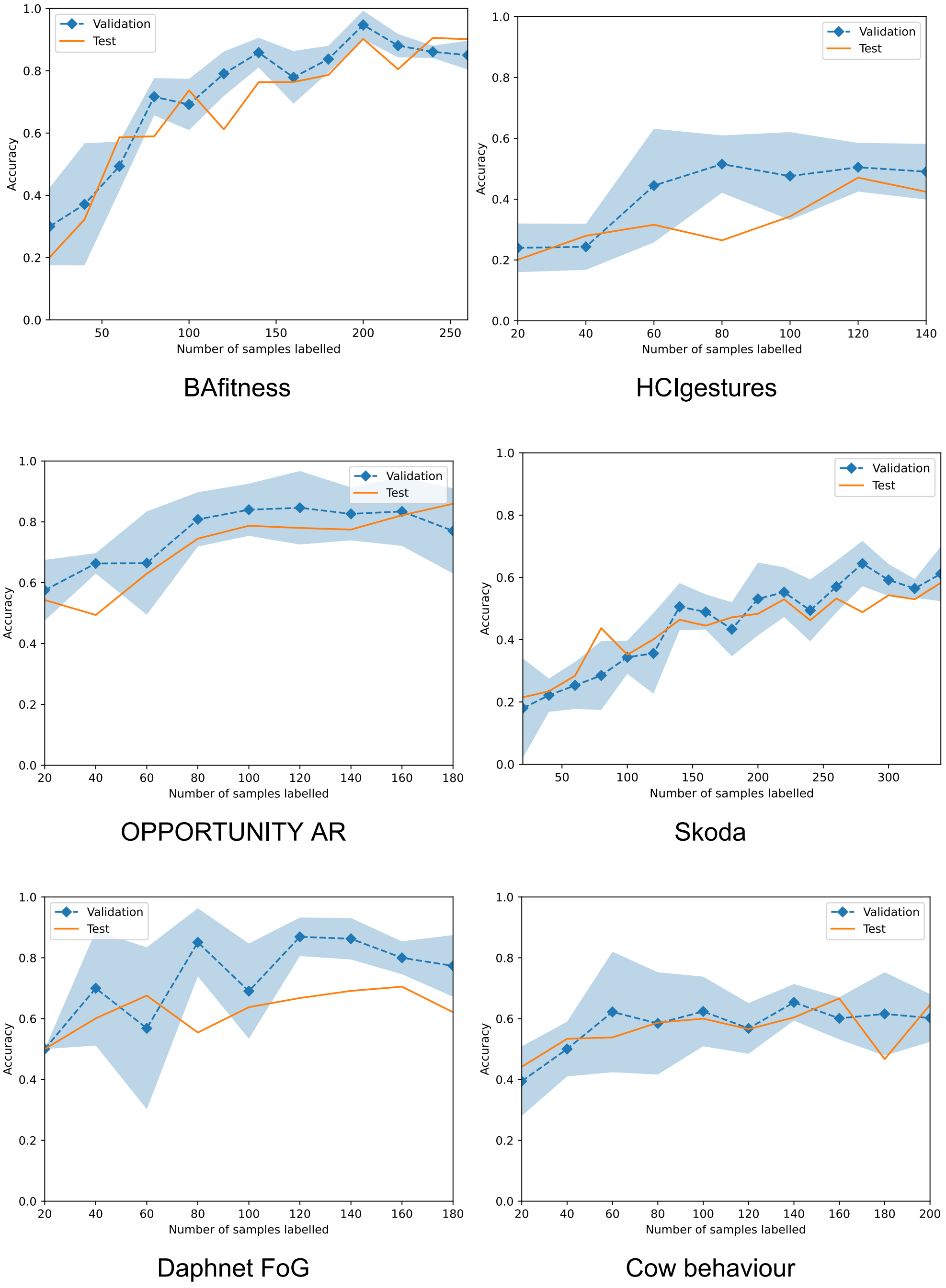
4.3.1 Automatic experiments

The results for each of the datasets, as the average and standard deviation over five runs, can be found in Table 4.2. Note, that the validation standard deviation and the MAD cannot be directly compared, as they are different metrics. However, they should both relate to the difference between the test accuracy and the validation accuracy. As can be seen, the system reaches relatively high accuracy for some datasets, while having more trouble with other datasets. The validation accuracy and test accuracy indeed seem to correlate, which is desired since the validation accuracy is used to estimate the test accuracy.

A single example run for each of the datasets can be found in Figure 4.3. Both the test and validation accuracy are shown. There seems to be an upwards trend in the test accuracy as the number of samples increases. The test accuracy generally falls within the standard deviation of the validation accuracy. For most datasets, the training process seems to have converged when the system finishes. However, it cannot be ruled out that the accuracy would not improve when adding more training samples.

Table 4.2: The results of using the annotation system for each of the datasets. In the first column, the expected random accuracy is shown, based on the number of classes for each dataset. In the other columns, the mean and standard deviation over five runs are given. The validation accuracy is the estimate from the k-fold cross validation, the standard deviation is calculated over the k folds. The MAD is between the test and validation accuracy of each separate run.

	Random accuracy	Test accuracy	Validation accuracy	Validation standard dev.	Number of samples	MAD
BAfitness	16.67%	84.35 ± 7.17%	89.78 ± 3.06%	5.86 ± 0.82%	212.00 ± 51.54	5.44 ± 4.47%
HCIgestures	20.00%	42.51 ± 13.89%	52.47 ± 3.96%	10.44 ± 6.05%	228.00 ± 118.39	11.93 ± 7.80%
OPPORTUNITY AR	25.00%	68.97 ± 11.72%	77.94 ± 4.60%	7.69 ± 2.80%	192.00 ± 65.24	8.97 ± 7.45%
Skoda	10.00%	46.32 ± 5.36%	53.00 ± 6.20%	8.56 ± 1.52%	272.00 ± 78.59	6.68 ± 6.60%
Daphnet FoG	50.00%	61.01 ± 5.80%	80.94 ± 2.38%	10.67 ± 6.81%	164.00 ± 61.19	19.93 ± 4.74%
Cow behaviour	33.33%	55.66 ± 6.77%	64.61 ± 6.63%	12.54 ± 4.13%	168.00 ± 24.00	11.67 ± 5.77%



BAfitness

HCIgestures

OPPORTUNITY AR

Skoda

Daphnet FoG

Cow behaviour

Figure 4.3: Example runs of the training process of Prototype I on each dataset. Both the test and validation accuracy is shown for the number of labelled samples. The shaded area depicts the standard deviation of the k-fold cross validation.

4.3.2 Pilot user study

The three users are referred to as 'user 1', 'user 2' and 'user 3'. Each user labelled 50 samples. The final obtained test accuracy for each of the users can be found in Table 4.3. As can the rater-reliability and the validation accuracy. The rater-reliability of the users is rather high, which means that the test accuracy based on the ground truth can be used, especially for user 1 and user 3. As can be seen, the resulting accuracies vary between users, but all accuracies are much higher than random (which would be 25%).

Table 4.3: Resulting rater-reliability, final test accuracies and final F1-scores from the three participants of the pilot user studies. The rater-reliability is calculated as Cohen's kappa. The test accuracy is based on the model trained after 50 video clips have been annotated.

	User 1	User 2	User 3
rater-reliability (no 'Other' or 'Unknown')	0.90	0.79	0.94
No. of samples labelled 'Other'	6	13	2
No. of samples labelled 'Unknown'	0	1	0
Test accuracy	48.95%	55.63%	63.88%

4.4 User feedback

In this section, answers, feedback and comments given during the user study are presented. In Table 4.4 the answers to a selection of questions can be found for each of the users. Some observations that were made for each user can also be found. In Table 4.5, general feedback and comments given by specific users can be found.

Uses all indicated that it was difficult to determine to which class samples belonged, but they also indicated that this is intrinsic to behavioural annotation. Especially if one is not annotating for their own research it can be difficult to determine to which class a sample belongs. The step of clicking the 'Apply' button was forgotten multiple times by multiple users. This caused mistakes to be made in the labelling.

Table 4.4: Answers to a selection of questions from the interviews and general observations applicable to users.

	User 1	User 2	User 3
Overall			
Would you want to use (a future version of) the Smart Annotation Tool?	Yes	Yes	Yes
How do you feel about the quality of the annotation?	Quality needs to increase		
Willing to label more samples to create a good annotation.	-	Yes	Yes
Tried to correct mistakes in the automatic labels tab.	Yes	Yes	Yes
Annotation process			
How do you feel about the length of the video clips?	Good	Prefer longer	Good
Used batch labelling.	No	No	Yes
Would you want to use batch labelling?	No	No	-
Difficult to determine classes corresponding to video clips.	Yes	Yes	Yes
User interface			
Forgot to click the ‘Apply’ button at least once.	No	Yes	Yes
Noticed the quality panel without instruction.	No	No	Yes

Table 4.5: Suggestions and points of feedback raised by specific users.

	User(s)
Overall	
Noticed mismatches between predicted accuracies and performance seen during visual inspection.	3
Wanted to know more about calculations going on behind the scenes.	1
Would want to know the confidence of the model for video clips.	1
Found the quality panel a useful feature.	1, 2, 3
Annotation process	
Liked being able to annotate per clip instead of from a data stream.	3
User interface	
Difficult to see the difference between labelled and unlabelled samples.	3
Would like the sidebar to scroll along.	3
Use a drop-down menu to switch between the classes for the automatic labels.	1
Difficult to see what is happening when the video jumps back to the beginning.	2
Noted having to scroll (a lot).	1, 2

4.5 Conclusions

As discussed at the beginning of this section, there were multiple purposes for the automatic tests and pilot user study. The conclusion for each of these purposes and possible corresponding research questions will be discussed. It should be noted that for the pilot user study only a small sample size was used, therefore conclusions concerning the user study should be regarded as indications.

1. Verify the baseline system can train the model for behavioural classification for various datasets (RQ 1.1, RQ 1.2).

The test accuracy on the unlabelled data for each of the datasets is presented in Table 4.2. The performance varies between different datasets. This is to be expected, some datasets are undoubtedly easier to classify than others. The performance on some datasets, specifically the HCIgestures and Cow behaviour dataset is rather low. For the HCIgestures and OPPORTUNITY AR datasets, there are big fluctuations in the obtained performance.

RQ 1.1 requires a classification accuracy that is high enough to meet user expectations. All participants expressed that they want to see an increase in accuracy. This shows that MiniROCKET does not reach a high enough classification accuracy by itself when only a small number of labelled samples is available. However, the main focus of the next research step will be to control the accuracy, starting from this baseline classification accuracy. Therefore, for the limits of this research, the initial performance of MiniROCKET is considered acceptable.

While improvements need to be made in reaching a high classification accuracy, the annotation efficiency is high. For most datasets, about 200 samples were used to train the models. For the Skoda dataset more samples were used. This makes sense since there are considerably more classes in this dataset.

Especially compared to the sizes of the datasets, see Table 3.1, the 200-350 samples needed are considered acceptable. Users indicated they had no issues with labelling higher numbers of samples than the 50 used in the user study, to create a good annotation.

The pilot user study did not explicitly focus on the computation time. However, none of the participants expressed that they noticed that the training of the model took too long. This aspect of RQ 1.1 will be further discussed for the final user study, see Section 8.2.

In conclusion, MiniROCKET seems to be an acceptable model to quickly classify behaviour using a relatively small amount of data, although the classification accuracy still needs to be improved.

To answer RQ 1.2, MBCs can be trained without interference by machine learning experts when utilizing hyperparameter tuning (with TPE sampling, Hyperband pruning and convergence estimation) combined with k-fold cross validation. To answer RQ 1.2.1, hyperparameter tuning and k-fold cross validation can also be used to optimize MBCs on different datasets. The quality of the performance estimation from the k-fold cross validation will be discussed for purpose 2.

For the convergence estimation, looking at the example run in Figure 4.3, some datasets may benefit from improvements. Especially for the Skoda dataset, the model accuracy is still increasing. For now, the current method will be accepted. The classification accuracy that is reached using hyperparameter optimization differs between datasets.

2. Verify the performance can be accurately estimated (RQ 2, RQ 1.2).

For all the averages presented in Table 4.2, the validation accuracy is higher than the test accuracy. This seems to indicate that the validation accuracy is an overly optimistic estimate. This disparity could have to do with the validation also having been used for the hyperparameter tuning. One of the participants in the user study also noticed a disparity between the performance estimate and what they saw for the automatically classified video clips.

The MAD denotes the difference between the validation and test accuracy. If the MAD is high, this means there is a mismatch between the validation and test data. Ideally, the validation standard deviation would represent this. Hence, if the MAD is high, it is also desired that the validation standard deviation is high. This seems to be the case for all datasets, except the Daphnet FoG dataset.

This can also be seen in Figure 4.3, the test accuracy is not properly estimated for the test run of the FoG dataset. For the other datasets, the test accuracy generally falls within the validation standard deviation with some outliers.

To conclude, the quality estimation as it currently works can keep users informed, but not by itself. At times a very incorrect estimate might be given. Therefore, the quality estimations should always be combined with visual inspection by the user.

In general, it seems like the convergence can be accurately estimated using the validation accuracy. In Figure 4.3, except for the HCIgestures dataset, the test and validation curves seem to be similar. Therefore, the quality estimation needed for the AutoML does seem good enough. This further answers RQ 1.2.

3. Verify the AutoML system also works with a human user (RQ 1).

It is difficult to compare the results obtained by the users with those from the automatic experiments. Small changes were made between the user studies and the final version of Prototype I. Furthermore, users only labelled 50 video clips and were allowed to use the ‘Other’ and ‘Unknown’ options. Users also made some accidental mistakes, specifically by forgetting to click on the ‘Apply’ button.

For users 1 and 3, Cohen’s kappa was above 0.9, denoting almost perfect agreement. For user 2 Cohen’s kappa was 0.79, denoting moderate agreement. Therefore, there is agreement between the users and the ground truth. As can be found in Table 4.3, the users reached accuracies between 48 and 66% with 50 labelled samples. This shows that even with the aforementioned issues, users were able to train the MBCs to some extent.

It is concluded, for RQ 1, that using the combination of MiniROCKET and AutoML methods presented in this section, non-machine learning expert human users can train a model for the classification of behaviour.

4. Identify any obstructive issues with the user interface (RQ 2).

Users accidentally skipped the step where they need to press the ‘Apply’ button. This resulted in mistakes being made. Therefore, the UI does not work well enough in its current form. Furthermore, users had trouble seeing what happens in the video clips. Users 1 and 2 did not pay attention to the annotation progress information without the researcher pointing it out. One of the users did not like the navigation in the ‘Manual Labels’ and ‘Automatic Labels’ tabs. Finally, users had to scroll quite a bit to apply labels, this could lead to a loss of overview for users.

One of the users did indicate that they would prefer longer videos, due to difficulty figuring out what was going on in the videos. The other two were fine with the 2-second videos. It could be the case that when clearing up the issues with seeing what is going on in the video, the 2-second videos will not be an issue anymore.

Even though two of the users indicated that they would prefer one-by-one labelling, the batch labelling interface is kept. Changing this is out of the scope of this project and one-by-one labelling is possible in the current UI, by simply only selecting one video at a time.

5. To streamline the final user study for Prototype II.

The last purpose is to streamline the final user study. For the pilot user study, all users found it difficult to determine the correct classes from the video clips. For the final user study, participants should be more clearly instructed on what kind of behaviour is expected for the various classes. Furthermore, users were interested in the workings of the system, this fits with the findings in literature, see Section 2.8. Therefore, more information will be shared during the introduction of the user study.

5 PROTOTYPE II: PRESERVING EFFECTIVENESS

In this section, classification correctness prediction is used to regulate the annotation quality. Annotation quality is regulated by only automatically annotating samples the MBC is expected to label correctly. This section aims to answer RQ 3, see Section 1.4.

In this section, the method used for Prototype II will be discussed, followed by the experimental setup and the results of these experiments. The conclusions of the experiments will be used to finalize Prototype II and answer RQ 3, the conclusions can be found in Section 5.5. The finalized version of Prototype II will be used for the final results and user study, in Section 6 and Section 7.

5.1 Method

5.1.1 Predicting classification correctness

To estimate which samples the MBC can classify correctly, a method is required to separate the correctly classified samples from the incorrectly classified samples. For this research, the samples will be separated based on the estimated certainty of the MBC.

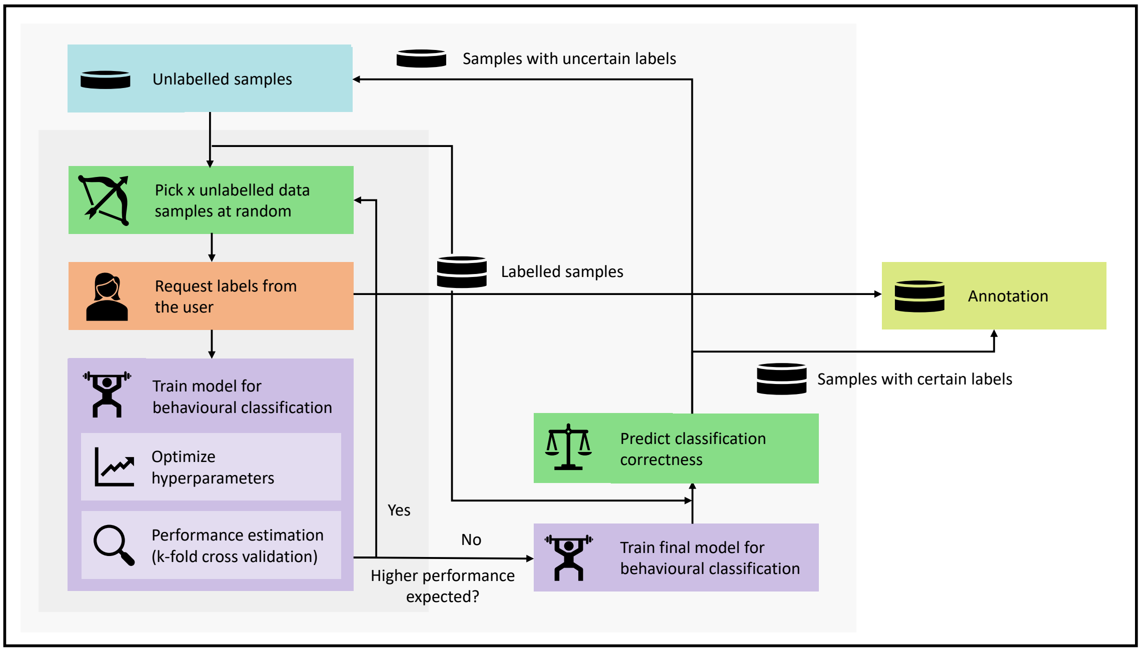
The estimation of whether a sample is correctly classified will be referred to as the classification correctness prediction (CCP). An overview of the system that combines the CCP with the model for behavioural classification (MBC) can be found in Figure 5.1. The MBC will always be a MiniROCKET classifier for this research. The combination of both models is referred to as the 'annotation system'.

The CCP classifies a sample as correctly or incorrectly classified by the MBC. To do so, it estimates the certainty of the MBC classification and imposes a threshold on the resulting value. To measure the certainty of the MBC, multiple metrics representing model certainty are considered, as will be discussed in Section 5.1.2. The threshold is set using k-fold validation based on the precision and the recall. This will be discussed in the next section.

Precision versus recall

The CCP should estimate a value m_c^x, signifying the MBC certainty, for each data sample x. This value can be converted into a classification by setting a threshold t_c. A sample that is correctly classified by the MBC will be considered a positive case, and a sample that is incorrectly classified will be considered a negative case.

The threshold is set based on a trade-off between the precision and the recall [92]. As per the problem statement, see Section 1.3, the Smart Annotation Tool should reduce annotation time (increase efficiency), but only at a limited cost of quality (maintain effectiveness). The quality of the annotation can be measured as the number of correctly classified samples added to



Unlabelled samples
Samples with uncertain labels
Pick x unlabelled data samples at random
Request labels from the user
Labelled samples
Annotation
Samples with certain labels
Train model for behavioural classification
Optimize hyperparameters
Performance estimation (k-fold cross validation)
Predict classification correctness
Yes
No
Higher performance expected?
Train final model for behavioural classification

Figure 5.1: Overview of the annotation system for Prototype II. For the unlabelled samples, the CCP estimates whether the MBC is certain 'enough' regarding its predictions. Samples of which the MBC is deemed certain, are labelled using the MBC. Other samples remain unlabelled. Hyperparameter tuning is used for setting the threshold for the CCP.

the annotation, compared to the total number of samples added to the annotation. Hence, the quality can be measured by the number of true positive samples TP, compared to all positive samples. This is given by the precision:

$$\text{precision} = \frac{TP}{TP + FP}.$$

Where FP denotes the number of false positive samples. Note, the precision will be considered 1.0 if none of the samples are classified as positive by the MBC.

The reduction in time can be measured by the number of samples classified as correct by the CCP. However, only the true positive samples will be considered, since more false positive samples should not be considered a win. The total number of positive samples is influenced by the performance of the MBC. Hence, the number of true positive samples depends on both the MBC and the CCP. Therefore, the CCP performance is measured by the number of true positive samples with respect to all positive samples, given the MBC's performance. This is given by the recall:

$$\text{recall} = \frac{TP}{TP + FN}.$$

Where FN denotes the number of false negative samples. Note, the recall will be considered 1.0 if none of the samples are predicted correctly by the MBC.

In an attempt to reach high precision, the system might favour majority class samples. It is undesired that the system cuts the quality of minority classes. Therefore, the precision and recall, as will be used during validation, will both be weighted by the occurrence of the (ground truth) behaviour labels. Note, that this means that the recall relates to the time reduction in the balanced case, the actual reduction in time may be higher or lower depending on the imbalance of the dataset. From now on, when the precision or recall are discussed, the weighted precision and recall are assumed.

Note that the weighted precision is the same metric for Prototype II as the accuracy was for Prototype I. For Prototype I, all samples were assumed to be positive cases since they were all added to the annotation, at once.

As discussed, an increase in efficiency may only have a limited impact on the effectiveness. The question remains, how much is the effectiveness allowed to decrease? This may depend on the application, this is why it is decided to leave the choice to the user. The user will be able to specify a target precision which the system aims to meet.

This target precision will be used to set the threshold t_c. The value of the recall will show how well the estimated metric m_c^x separates the data. Therefore, the performance of the CCP can be considered as the recall resulting from the threshold t_c, given the MBC's classification accuracy and the CCP's ability to meet the target precision.

Thresholding

The threshold t_c is set based on the target precision specified by the user. The threshold is set based on the labelled data. The same k models and data splits used in the k-fold cross validation used for the MBC are leveraged. During the validation, for each fold i a threshold t_c^i is set.

To set the threshold, the precision of the CCP is measured at different thresholds for each fold. The precision is measured for n_t equidistant values considered for t_c^i. These values are taken between the minimum and maximum value of the MBC certainty (m_c^x) in the fold. The least strict threshold matching the target precision is chosen.

The resulting k thresholds t_c^i are combined into a single threshold. Two methods of combining the thresholds are considered. The first is the **'extreme'** threshold setting, which sets the t_c as the strictest threshold. This is either the minimum or maximum value of t_c^i over all i, depending on whether a high or low value of m_c^x denotes a high certainty. The other is the **'mean'** threshold setting, which sets t_c as the mean value over t_c^i for all i. The 'extreme' threshold setting will be more conservative regarding the precision, the 'mean' threshold setting will have a higher recall.

One additional option is considered, instead of setting a single threshold for all data, a threshold can be set per class (and per threshold). The data is separated based on the predicted label for each sample. For each predicted class, the threshold is set to meet the precision target. This will be referred to as the **'per class'** setting, the regular method is referred to as the **'single'** setting. The 'per class' setting could allow the system to be more flexible regarding easier classes.

5.1.2 Metrics

Multiple metrics are considered for estimating the certainty of the MBC for the CCP. The metrics are (based on) the ones found in literature as discussed in Section 2.6.

Baseline

The baseline for the classification correctness estimation is the classification accuracy of the MBC. If the MBC is used to immediately label all unlabelled samples, as was the case for Prototype I, the resulting precision equals the classification accuracy. The recall is 1.0 in this case.

Basic metrics

Three metrics that directly use the MBC output are considered, these are referred to as the basic uncertainty metrics. These metrics are the maximum softmax output, maximum logit output and entropy. Since the entropy requires an (estimated) probability distribution, the entropy is always based on the softmax output. If the maximum output value is high (logit or softmax), the model is assumed to be certain about its prediction and a sample is assumed to be correctly classified by the MBC. For the entropy, this is the other way around.

ODIN

ODIN [60] is also considered. ODIN applies noise to the model input and a temperature to the softmax function, see Section 2.6. The maximum output of this modified softmax function is used as metric for uncertainty. It is again hypothesized to be high for certain samples. The noise and temperature are only applied when evaluating the classification correctness of samples, nothing is changed during the training of the MBC.

ODIN has two hyperparameters, the noise magnitude ϵ and the temperature T. These two hyperparameters are tuned after the MBC has been trained. They are tuned using the same k-fold partition of the data. The k MBCs trained during the trial corresponding to the 'best' hyperparameter configuration are used. The tuning objective is the recall. ODIN is tuned using grid search since each tune trial can be run relatively quickly.

Monte Carlo Dropout

Several metrics based on the principle of Monte Carlo Dropout are considered. Firstly, the entropy over the mean R model runs, as proposed by Gal et al. [63] is considered. This line of thinking, of applying metrics to the mean output, is extended to the maximum softmax output and the maximum logit output. The other metrics that will be considered are: BALD [68], the variation and the variation-ratio. Except for the maximum softmax output and maximum logit output, the metrics are assumed to be low for correctly classified samples.

MC dropout introduces a new hyperparameter to the system, the dropout-rate. The dropout-rate should be the same during training, hence changing it requires retraining of the model. It is possible to include the dropout-rate during the hyperparameter tuning of the MBC. However, this increases the computational complexity. Therefore, for this research, the model is limited to a fixed dropout-rate.

The number of model passes R is also a new hyperparameter. A higher value of R should result in a better approximation of the output. However, higher values will also make the system slow.

5.1.3 Cascading classification

The combination of the MBC and CCP is expected to confidentially label part of the dataset. Still, a set of samples will remain unlabelled afterwards. It would be an option to give all these unlabelled samples to the user so they can label them by hand. However, in the (highly) optimistic scenario where the MBC reaches a classification accuracy of 70% and the CCP reaches a recall of 70%, more than half of the dataset will still need to be labelled by hand. Therefore, the user likely still needs to do a lot of work.

While the user is labelling the remaining samples, new information is given to the system. This information will be readily available, ideally, it would be used to further increase efficiency. In this section, a method that repeatedly uses this user input is discussed.

When the CCP is used to create the automatic annotation, the MBC has been trained to confidently recognize part of the dataset. What is left unlabelled after the automatic annotation are (sub-)behaviours that the MBC has not yet learned to recognize. When the user labels more samples, more information becomes available regarding these (sub-) behaviours. These newly labelled samples can be used to train a new classifier to recognize some of the new behaviours.

For this research, the newly labelled samples are introduced in a cascading manner. At each 'level' of 'the cascade', an MBC is obtained, see Figure 5.1. After each level, the CCP is used to annotate the part of the dataset the MBC is expected to be certain about. The next level will do the same but only needs to focus on the remaining 'uncertain' samples.

Convergence objective

During each cascade level, the number of labelled samples needed to train the MBC is determined using convergence estimation. For Prototype I, the objective of this estimation was the MBC's accuracy. For Prototype II, not only the MBC's accuracy should increase as more samples are labelled, but also the recall of the CCP. Therefore, the system will be tested when using the MBC's accuracy as the objective and when using the mean between the MBC's accuracy and CCP's recall as the objective.

Discarding samples

At the end of a cascade level, choices can be made regarding what to do with the manually labelled samples that have been used for training and validation. They can be kept while newly labelled samples are added or discarded such that the level starts with a clean(-er) slate. If a sample is discarded, it remains in the annotation, but will not be included in the training and validation data. The following choices are considered:

1. **Discard all**: Discard all manually labelled samples and use only newly labelled samples in the next cascade level.
2. **Keep uncertain:** Keep manually labelled samples the CCP does not classify as correctly classified.
3. **Keep all:** Keep all samples that have been manually labelled.
4. **Pseudo-labelling:** Keep all manually labelled samples and add the automatically labelled samples [93].

The idea behind the first two options is that the system has already trained for the behaviours seen in the previous samples, hence discarding them could allow the focus to be on new behaviours. Furthermore, by removing samples from the training and validation sets, the system remains fast. This is the issue with the third and fourth options, unless the recall is sufficiently high they will lead to a big set of training samples and a slow system.

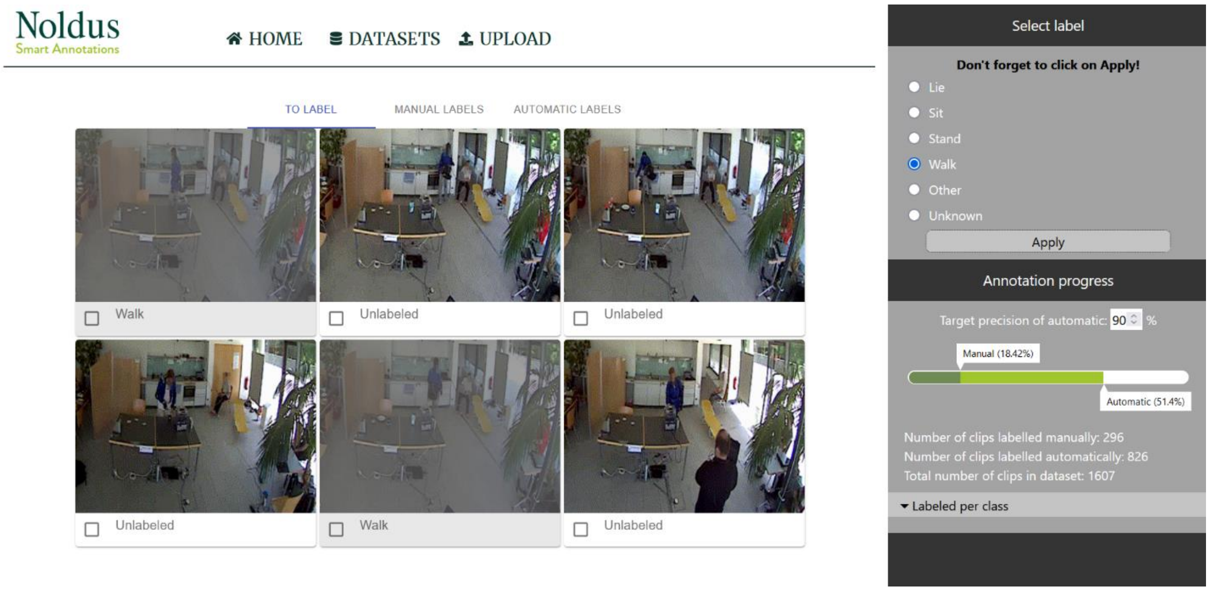
The 'keep uncertain' option could be advantageous since labels that have already been obtained of 'uncertain behaviours' remain available for training. A preliminary experiment was done where the 'keep uncertain' option did not lead to a notable increase in performance in general, see Appendix D.1. Note, this experiment was performed on the first 5000 samples in each of the datasets and with outdated code. During these preliminary experiments, the 'keep all' option was also considered. As expected, it led to a slight increase in performance but took a long time to compute.

The disadvantage of the 'keep uncertain option' is that besides potentially useful samples, outlier samples are also left in the training set. These samples may never leave the training set, resulting in a possibly skewed dataset. This also leads to more computation time.

The increase in computation time for the ‘keep uncertain’ option was obstructive during the preliminary experiments. Therefore, due to time constraints, this option is not further explored. Only the ‘discard all’ option is considered for the results.

5.1.4 User interface

For Prototype I, an estimation of the classification accuracy of the MBC was shown to the user. For Prototype II, due to the addition of the CCP, this is not the most useful information anymore. Instead, the number of samples that has been labelled so far, both by the user and automatically, is shown to the user. This information is shown numerically and in the form of a progress bar, see Figure 5.2. The number of samples labelled per class is also shown.



□ Walk
□ Unlabeled
□ Unlabeled
□ Unlabeled
□ Walk
□ Unlabeled

Don't forget to click on Apply!

○ Lie
○ Sit
○ Stand
● Walk
○ Other
○ Unknown

Target precision of automatic: 90 %

Number of clips labelled manually: 296
Number of clips labelled automatically: 826
Total number of clips in dataset: 1607

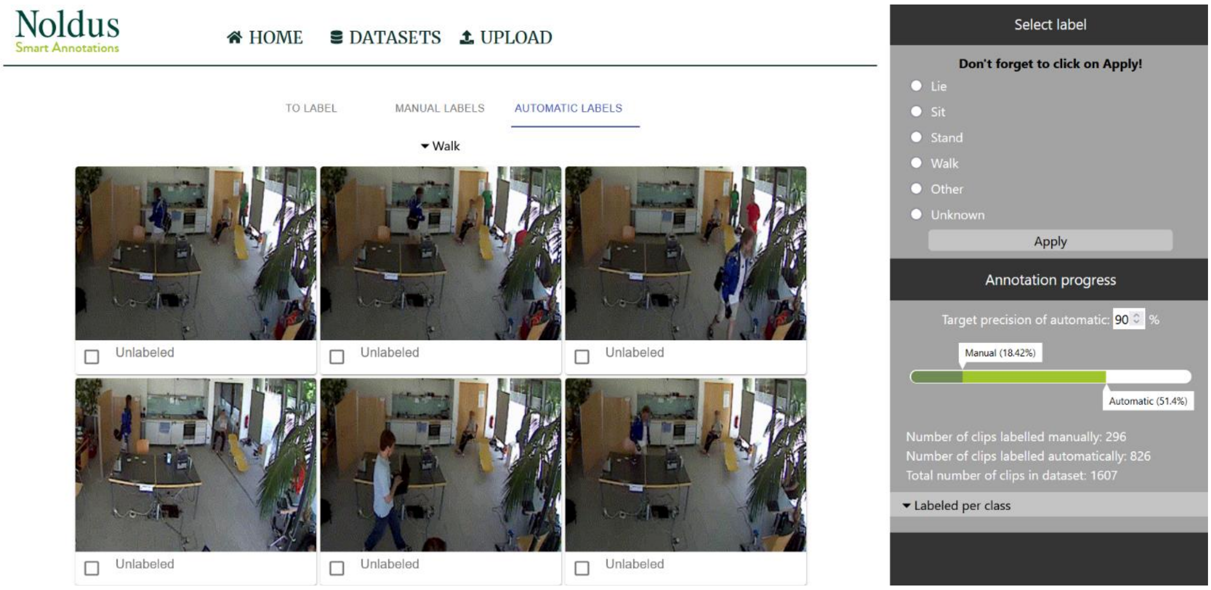
Figure 5.2: The ‘To Label’ tab of the user interface for Prototype II. The video clips are from the OPPORTUNITY++ dataset [82][1].

For the current implementation, the target precision used by the machine learning back end cannot be updated once annotation has been started. However, it is shown to the user as a way to make the concept clear to participants in the user study.

Further changes were made to the UI based on the issues found with the previous UI, see Section 4.5. The first issue was that users forgot to press the ‘Apply’ button. Therefore a reminder is displayed in the label panel. A more reliable solution is not considered due to time constraints. Users also had trouble seeing what happened in the video clips, therefore users are now given the option to enlarge video clips. When they click on a video clip it is shown in full-screen mode.

For Prototype I, users had to scroll too much. Therefore, the number of samples shown at the same time in the ‘To Label’ tab is lowered to 6, such that they all fit into the screen. The machine learning back end is still updated every x labelled samples. However, these samples are split into smaller batches.

To make it more clear which videos still need to be labelled by the user, labelled videos are



Don't forget to click on Apply!

Target precision of automatic: 90 %

Number of clips labelled manually: 296
Number of clips labelled automatically: 826
Total number of clips in dataset: 1607

Figure 5.3: The 'Automatic Labels' tab of the user interface for Prototype II. The video clips are from the OPPORTUNITY++ dataset [82][2].

shaded to make the others stand out.

The updating of the machine learning back end may take a while, especially when more samples have been labelled. Therefore, users are already presented with the next samples while the machine learning back end is still being updated. However, when users have labelled another batch and the back end is still updating they need to wait until the update is finished.

For the 'Manual Labels' and 'Automatic Labels' tabs, the sidebar is fixed such that users do not need to scroll back up to make changes to the labelling, as was the case for Prototype I. For Prototype II, the user can adjust incorrect labels in the 'Automatic Labels' tab. The adjusted samples are removed from this tab and moved to the 'Manual Labels' tab.

Furthermore, the navigation in the 'Manual Labels' and 'Automatic Labels' tabs is changed based on a participant's suggestion in the pilot user study. Instead of needing to click through the tabs to reach different classes, the desired class can be selected using a drop-down menu. In Figure 5.3, the interface for the 'Automatic Labels' tab is shown, the 'Manual Labels' tab looks similar.

5.2 Experimental setup

The hyperparameter settings for Prototype I are copied for the setup of Prototype II, see Section 4.2.1. Additional hyperparameters are added for ODIN and MC dropout. For ODIN, the temperature T and magnitude of the noise ϵ are tunable hyperparameters. For the MC dropout, the performance will be evaluated for fixed hyperparameters, see Section 5.1.2.

Liang et al. [60] choose the temperature T from the set: $[1, 2, 5, 10, 20, 50, 100, 200, 500, 1000]$. ϵ is chosen from 21 evenly spaced values in the range $[0, 0.004]$. For this research, the search space is made smaller to speed up tuning. The values used can be found in Table 5.1. The implementation of ODIN is modified from the published implementation[3].

[2] www.creativecommons.org/licenses/by/4.0

[3] www.github.com/facebookresearch/odin

Table 5.1: An overview of the new hyperparameters used for Prototype II. The hyperparameters for ODIN are tuned using grid search. The hyperparameters for MC dropout are fixed.

Hyperparameter	Type	CCP method	Value/range/set
T	Grid search	ODIN	$[1, 5, 10, 50, 100, 500, 1000]$
ϵ	Grid search	ODIN	$[0, 0.1, 0.4, 0.01, 0.04, 0.001, 0.004]$
Dropout-rate	Fixed	MC dropout	$[0.1, 0.2, 0.4]$
R	Fixed	MC dropout	$[1, 10, 50]$

For the MC dropout, three different dropout-rates will be considered: 0.1, 0.2 and 0.4. For the number of model passes $R = 1, 10$ and 50 and considered.

The number of thresholds considered for the CCP is set to n_t = 20.

Results are reported over five runs. For the basic and ODIN metrics five MBCs are trained to which the CCPs are applied. For the MC dropout metrics, different MBCs are trained due to the need for training with the dropout-rate. These MBCs are trained on the same samples as used for the basic and ODIN metrics.

Results for the CCP are reported for training and validating the model once after manually labelling 200 samples, rather than using convergence estimation. This is done to allow for a consistent comparison. Note, this means that the hyperparameter tuning used does not have knowledge of the best parameters over any previous runs, as is the case when using convergence estimation. For the cascading classification, convergence estimation is used.

5.3 Results

In this section the results for the separate CCP experiments and the full cascading classification are presented. Results are given as the mean and standard deviation over five runs for each dataset. The same colours will be used to refer to the different datasets in the remainder of this thesis. The legend that is used can be found in Figure 5.4.

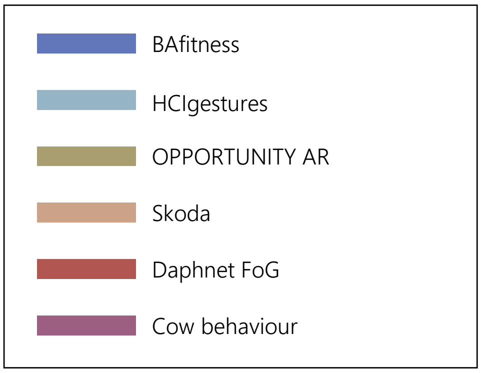


Figure 5.4: Legend for the different datasets, see Section 3.

5.3.1 CCP

This result section both aims to determine the best settings for the CCP and to analyze the performance. These results are necessary to answer RQ 3.1. First, the results used to determine which settings to use are presented. Afterwards,the CCP performance is further analyzed.

For the CCP experiments, MBCs are trained on 200 labelled samples. These MBCs are used to compare the performance of the CCP for different settings and metrics. For the basic metrics and ODIN the same MBCs are used. The test classification accuracy of these models can be found in Table 5.2.

For the MC dropout experiments, different MBCs are used, but these models are trained on the same samples as for the other metrics. The performance of the MBCs trained with a dropout-rate of 0.2 and number of model runs $R = 50$ can also be found in Table 5.2. The test classification accuracy is considered to be the mean over the R runs.

Table 5.2: Mean test classification accuracy of the MBC over five runs. Results are given for the scenario without dropout and the scenario with a dropout-rate of 0.2. The MBCs are trained on 200 samples. For the dropout-rate of 0.2, $R = 50$ is used.

	Dropout-rate = 0	Dropout-rate = 0.2
BAfitness	87.10 ± 3.88%	85.48 ± 9.20%
HCIgestures	55.81 ± 7.25%	50.67 ± 3.31%
OPPORTUNITY AR	68.67 ± 11.82%	76.62 ± 5.35%
Skoda	64.33 ± 5.7%	65.57 ± 3.11%
Daphnet FoG	67.94 ± 7.19%	61.47 ± 9.55%
Cow behaviour	59.84 ± 5.28%	56.88 ± 3.46%

To keep this section readable, only the most important results are presented in this section. The other results are presented in Appendix D.2. A summary of the results used to determine the CCP settings is given.

Initial results

For the CCP, two options need to be decided on: whether to use the 'extreme' or 'mean' setting and whether to set a single threshold or a threshold per class, see Section 5.1.1. In Appendix D.2.1, it is found that the extreme setting is more conservative than the mean setting, as expected. It is also found that the 'single' setting is more conservative than the 'per class' setting. This is likely due to the per class setting having to set the threshold with very little data. Since, as is shown in Section 5.3.1, the CCP needs to be treated conservatively, the extreme and single settings are used for the further results in this section.

Three different basic metrics are considered, see Section 5.1.2. Namely: the maximum softmax output, the maximum logit output and the entropy. In Appendix D.2.2, it is found that there does not seem to be one metric that notably performs better.

For the Monte Carlo dropout metrics, the dropout-rate and number of model runs remain to be chosen, see Section 5.1.2. In Appendix D.2.3, it is found that a dropout-rate of 0.2 seems to give the most consistent results. Using $R = 10$ or $R = 50$ model runs seemed to give slightly better results than using $R = 1$ model runs. However, the differences are slight.

The results for the different MC dropout metrics, see Section 5.1.2, can also be found in Appendix D.2.3. Again, the exact choice of metric does not seem to have a big influence. Only the variation-ratio performs notably worse, likely due to its discrete nature.

Metric comparison

In Figure 5.5 the precision for the maximum softmax output, ODIN and MC dropout version of the maximum softmax output are shown. The test accuracy of the MBC is also shown since the performance of the MBC may influence the performance of the CCP and the MBCs are different for the basic and ODIN metrics and the MC dropout metric. The precision target is set to 0.9. For the MC dropout, the dropout-rate is 0.2 and $R = 50$. The recall for the same metrics can be found in Figure 5.6. The recall when fixing the test precision at 0.9, can be found in Figure 5.7.

The differences in precision are not consistent between the different methods. For the recall at the fixed precision of 0.9, ODIN does seem to perform the best, specifically for the Skoda, Daphnet FoG and Cow behaviour datasets. This shows that ODIN is likely the best metric in general for separating the certain and the uncertain samples.

However, it is only the best metric in the ideal situation where enough data is available at all times. The actual system only obtains the recall as shown in Figure 5.6. For this recall, there is little difference between the different metrics.

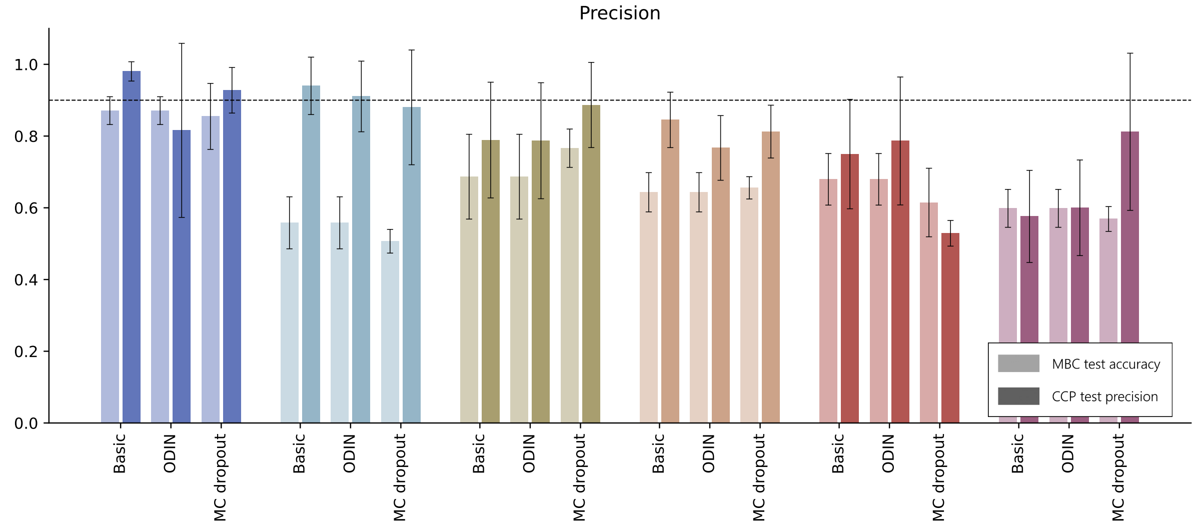


Figure 5.5: Comparison of the precision of the CCP for the basic, ODIN and MC dropout metrics. The test accuracy represents the baseline. For each metric the maximum softmax output is used. The target precision is 0.9, for the MC dropout the dropout-rate is 0.2 and R = 50.

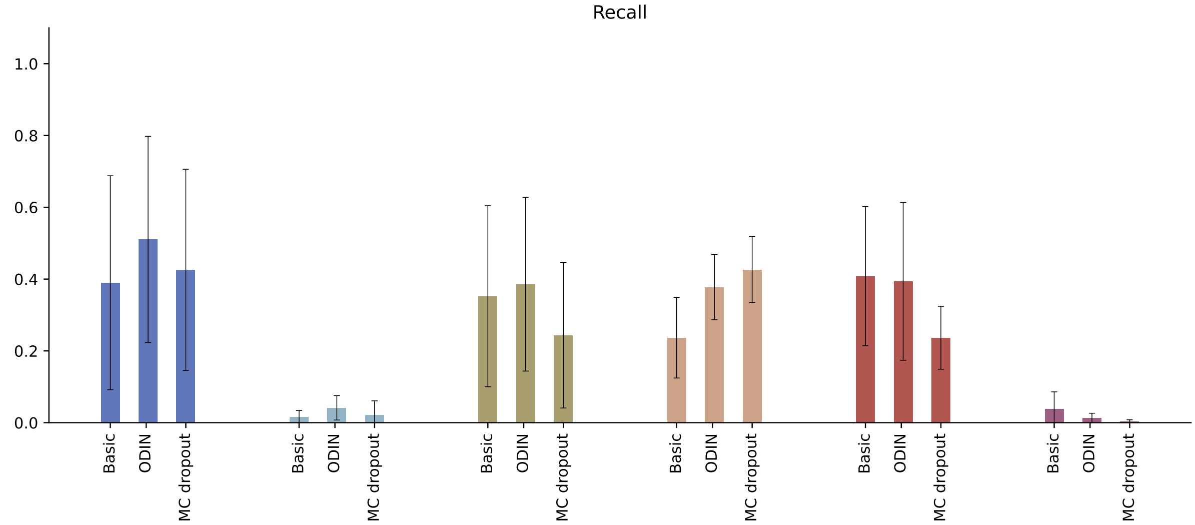


Figure 5.6: Comparison of the recall of the CCP for the basic, ODIN and MC dropout metrics. For all three the maximum softmax output is used. The target precision is 0.9, for the MC dropout the dropout-rate is 0.2 and R = 50.

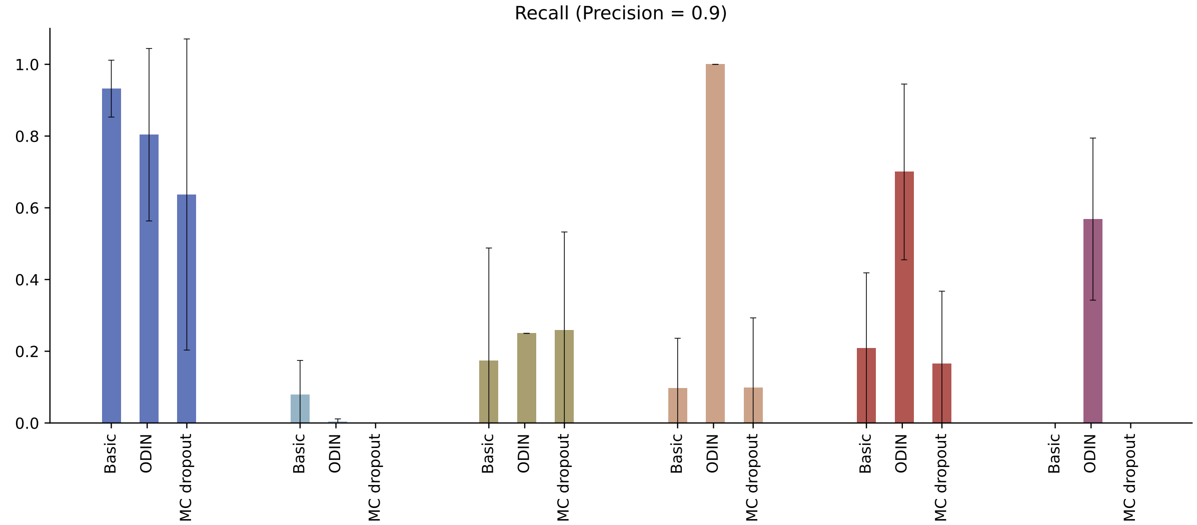


Figure 5.7: Comparison of the recall of the CCP for the basic, ODIN and MC dropout metrics. For all three the maximum softmax output is used. The target precision is 0.9, for the MC dropout the dropout-rate is 0.2 and R = 50.

CCP analysis

In Figure 5.8, the precision for the validation, training and test scenarios is shown. The result for the training scenario is the performance when applying the CCP to the data used for training and validation. The result for the validation scenario results from the k-fold cross validation used to set the CCP threshold. The result for the test scenario is the actual performance.

The difference between the validation and training results is considerably smaller than between the validation and test results. This shows that any differences between the MBCs used during k-fold cross validation and the final MBC have a small influence on the dip in performance for the test scenario.

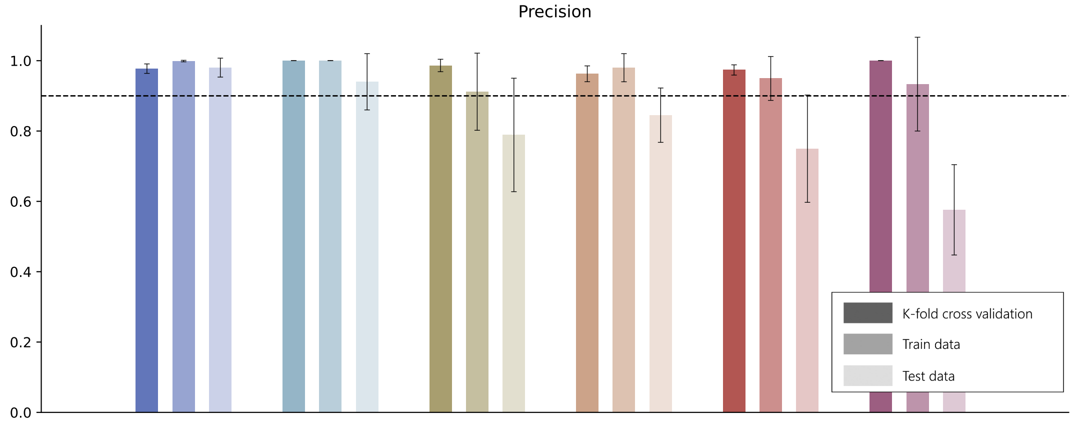


Figure 5.8: The obtained precision for the k-fold cross validation scenario, the training scenario for the final MBC and the test scenario on the unlabelled data. The CCP metric used is the basic maximum softmax certainty, the target precision is 0.9.

In Figure 5.9, the precision when using the **same** data samples for the MBC training and validation as for the CCP validation is compared with the precision when using **separate** samples. The former is what currently happens in the system. In both cases, 200 samples are used for the CCP validation. For the measuring of both performances, the unlabelled samples from the separate scenario are used. As can be seen, there are no notable differences. This shows that the bias towards the validation data is minimal.

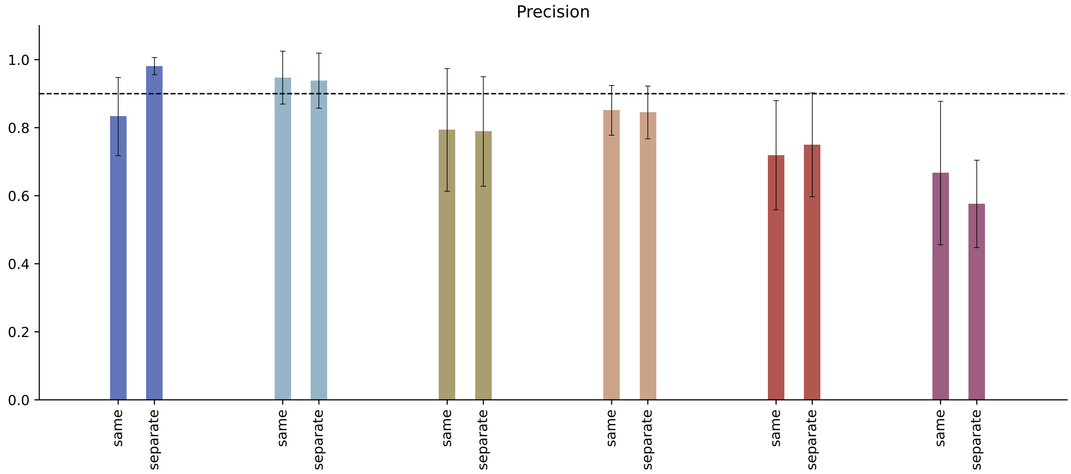


Figure 5.9: The obtained precision when using the same samples as used for the MBC training for the CCP validation and when using a separate set of samples. The CCP metric is the basic maximum softmax certainty, the target precision is 0.9.

In Figures 5.10 and 5.11 the results can be found when increasing the number of labelled samples. The newly labelled samples (after the initial 200) are only used when setting the CCP threshold. The additional samples are added to each fold. For each scenario, the same test samples are used. This means that for the BAfitness, HCIgestures and OPPORTUNITY AR dataset the unlabelled samples from the scenarios with 1000 labelled samples are used. For the other datasets, the unlabelled samples from the scenarios with 10.000 labelled samples are used. All results are shown for the basic maximum softmax output metric.

It can be seen that as the number of samples increases, the precision target is met more closely. This shows that the lower numbers of labelled samples may not contain sufficient information.

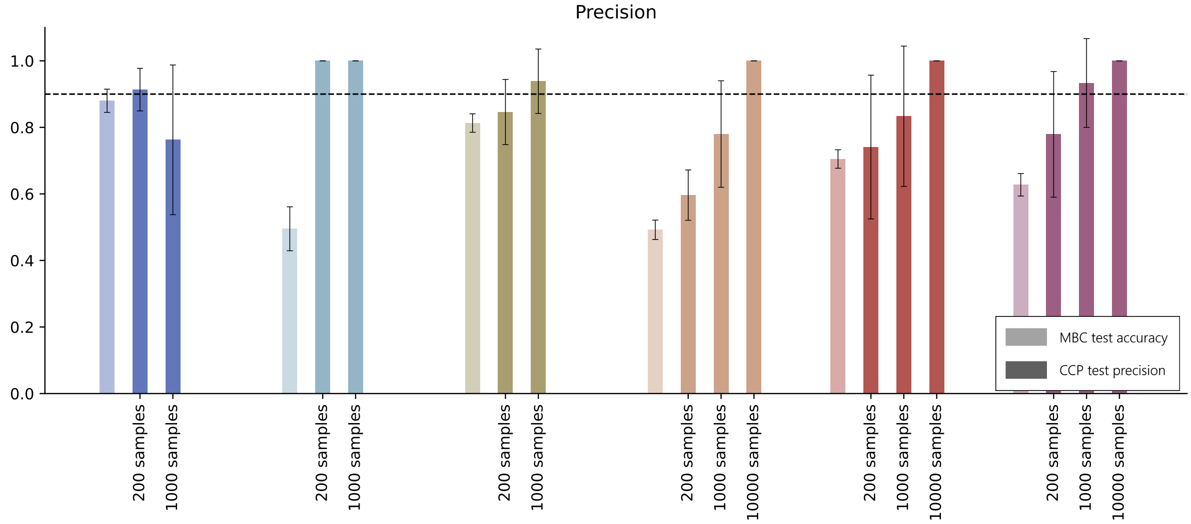


Figure 5.10: Test accuracy and precision when increasing the number of labelled samples, additional samples are only used as validation data for setting the CCP threshold. The CCP metric is the basic maximum softmax output, the target precision is 0.9.

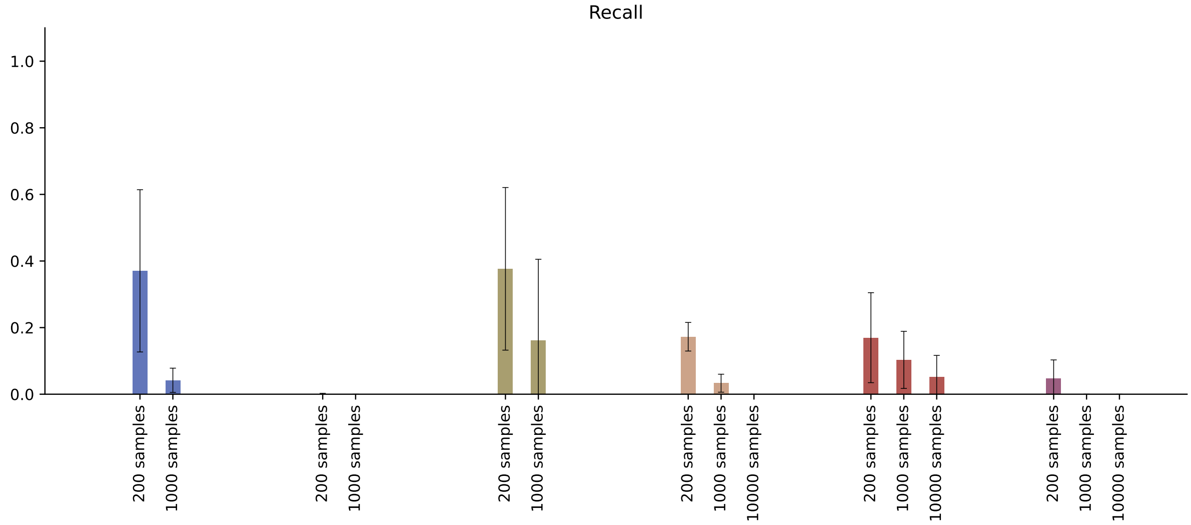


Figure 5.11: Recall when increasing the number of labelled samples, additional samples are only used as validation data for setting the CCP threshold. The CCP metric is the basic maximum softmax output, the target precision is 0.9.

The sample principle is repeated, but now the newly labelled samples are used throughout the system, also for the training and tuning of the MBC. The results for this can be found in Figures 5.12 and 5.13. It can be seen, in comparison to only adding the newly labelled samples for the CCP, that the recall is much higher, while the precision is comparable.

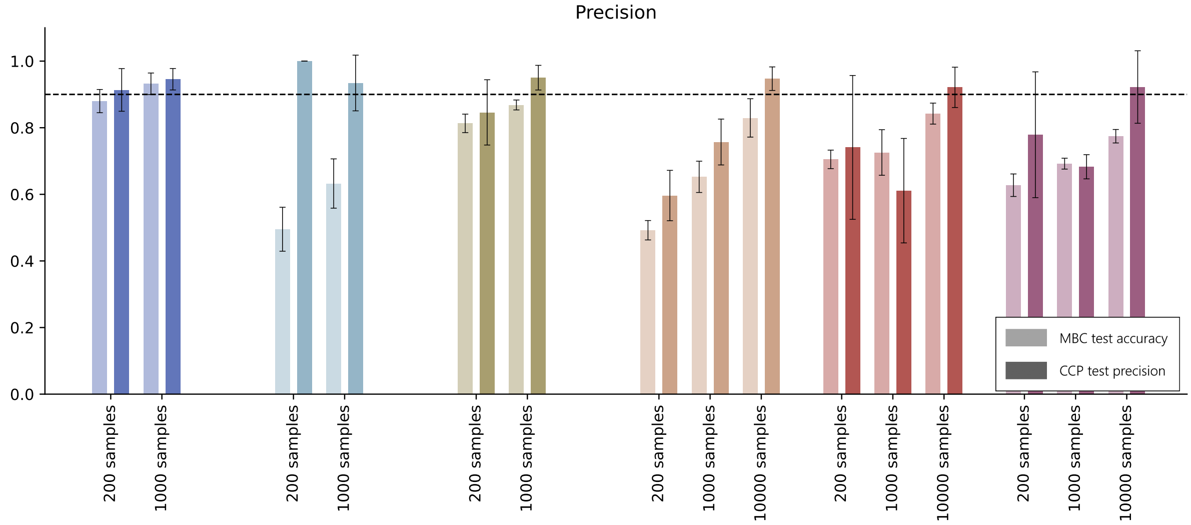


Figure 5.12: Precision and test accuracy when increasing the number of labelled samples. The CCP metric is the basic maximum softmax output, the target precision is 0.9.

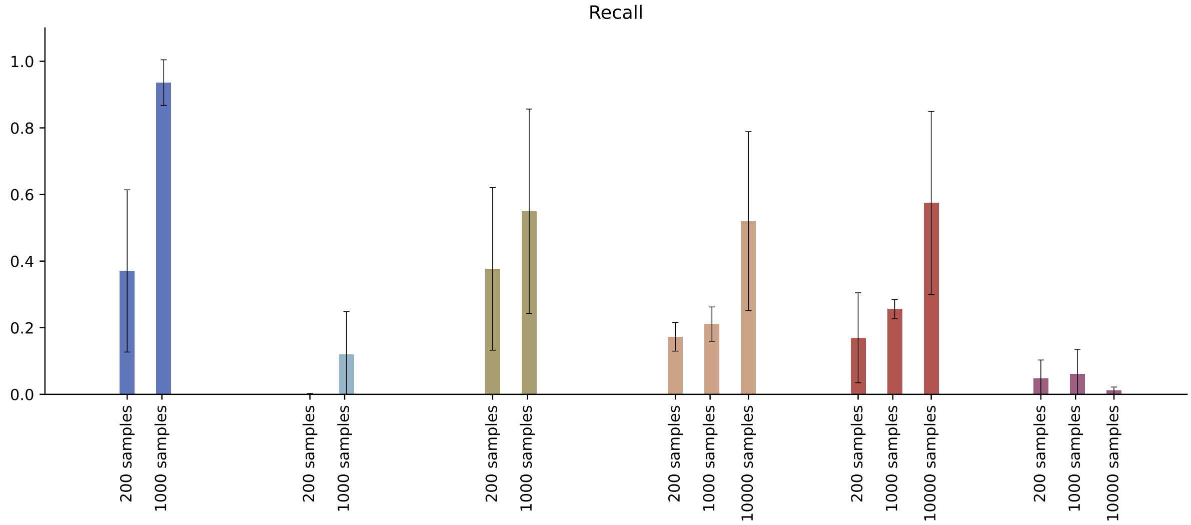


Figure 5.13: Recall when increasing the number of labelled samples. The CCP metric is the basic maximum softmax output, the target precision is 0.9.

5.3.2 Cascading classification

The CCP can be used to annotate the full dataset using a cascading approach, as discussed in Section 5.1.3. The results for this can be found in Figure 5.14. The performance of an annotation is shown in terms of the precision of the automatically labelled samples and the fraction of the dataset that was automatically labelled. The results are based on the basic maximum softmax output metric for a target precision of 0.9. The objective used for the convergence estimation of the number of labelled samples is the MBC accuracy.

As can be seen, the system annotates part of the datasets automatically. The system does have trouble meeting the precision target of 0.9 for most datasets.

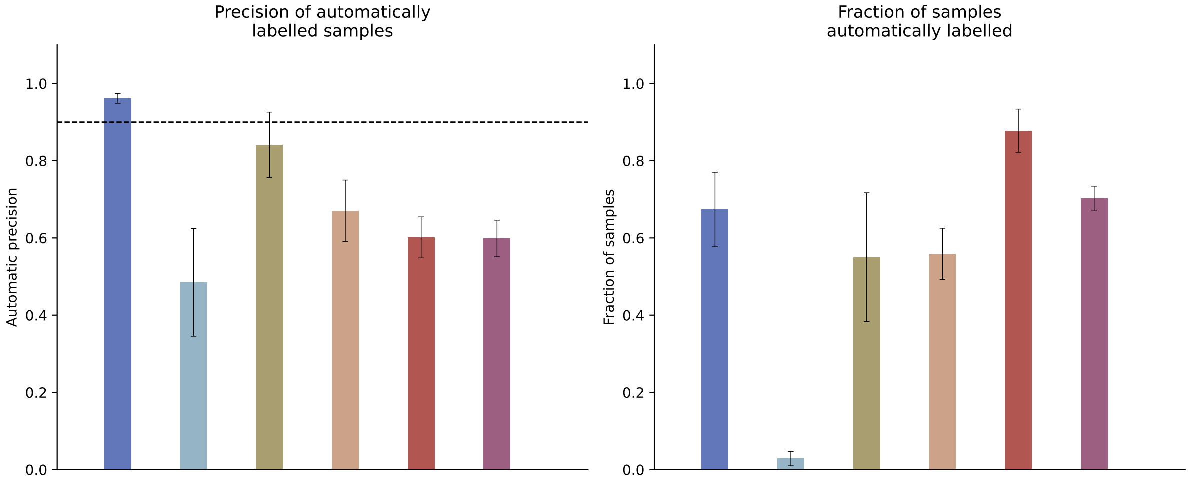


Figure 5.14: Precision of the automatically labelled samples and the fraction of samples automatically labelled for the cascading classification. Note, the precision is weighted per class but the fraction of automatically labelled samples is not.

For the cascading classification, it was hypothesized that including the recall in the objective for the convergence estimation (with respect to the number of labelled samples) would increase the performance, see Section 5.1.3. For Prototype I, only the MBC's accuracy was used. However, as discussed in Appendix D.2.4, using the mean between the MBC's accuracy and the CCP's recall is detrimental to the results. Either the system is not able to correctly estimate the recall or the recall does not behave consistently.

5.4 Discussion

5.4.1 CCP

In this section, the comparison of the metrics used for uncertainty estimation will first be discussed. Afterwards, issues meeting the precision target will be analyzed.

Comparison

Ideally, the different metrics for the CCP would be compared based on the resulting recall. However, meeting the target precision has a higher priority. It was found that reaching this target precision is an issue in general. As can be seen in Figure 5.5, the CPP does not consistently meet the target precision for any of the metrics. Especially for the Skoda, Daphnet FoG and Cow behaviour datasets, the CCP comes short. The exact choice of metric does not seem to have a big effect.

As can be seen in Figure 5.7, when setting the CCP threshold using the test data, ODIN seems to generally reach the highest recall. Most of this advantage is lost when setting the recall using the validation data, as can be seen in Figure 5.6. As a result, in the CCP, the exact choice of the metric is not expected to make a big difference.

The basic metrics do not require any additional tuning, which ODIN does require. MC Dropout requires multiple model runs, making it also slower than the basic metric. Therefore, the basic maximum softmax output will be used in the remainder of this report.

Analysis

No matter the choice of metric, the CCP has trouble meeting the precision target for some datasets. The precision not being met is a sign that the validation scenario used to set the CCP threshold is not representative of the test scenario. The following reasons are hypothesized for this mismatch between the validation and test scenarios:

1. The k MBCs trained during k-fold cross validation variate too much from the final MBC trained on all data.
2. The MBC is biased towards the validation data since the validation data is used for choosing the best hyperparameter configuration.
3. The validation data is not sufficiently representative of the test data.

As a consequence of the first reason, one would expect a higher precision for the k-fold cross validation than for the CCP with the final MBC on the labelled data (as used for training). As can be seen in Figure 5.8, for the OPPORTUNITY AR and Aeres datasets there are some deviations between the validation and training scenarios. However, for the training scenario, the precision target is generally met. This shows that any differences between the k validation models and the final MBC indeed have some influence on the results. However, since there is a big dip in precision between the training and test data, the second or third reason likely has a bigger impact.

The second reason suggests that the MBC may be biased towards the validation data. However, as can be seen in Figure 5.9, the system does not seem to have any advantage when using different validation data for the CCP than for the training and validation of the MBC. Therefore, the second reason is rejected.

As can be seen in Figure 5.10, adding more samples to the validation set used for setting the CCP threshold, generally increases the precision. This shows that when extending the validation set the precision target is met more closely. This is a consequence of the third reason. The CCP has access to more validation data, hence more representative data may be available and the performance increases.

Hence, while the difference between the models used during cross validation and the final MBC seems to have some effect on the performance, the third reason seems to have the biggest impact: the validation dataset does not contain enough representative data for all datasets.

When increasing the number of labelled samples and only using them for the CCP validation, the recall gets small, see Figure 5.11. Especially for the HCIgestures, Skoda and Aeres datasets. When adding the additional labelled samples also to the training dataset, the recall is much higher see Figure 5.13. This is likely due to the increase in the test accuracy for the MBC as the number of samples increases. If the test accuracy is higher, the CCP starts with a higher precision, making its task easier. The precision remains comparable in this scenario, see Figure 5.10.

5.4.2 Cascading classification

Using the cascading classification approach, annotation using CCP can be completed for the full dataset. However, as can be seen in Figure 5.14, the approach has issues meeting the target precision. This is a logical consequence of the CCP not performing well enough, as previously discussed.

Interestingly, for the CCP experiments the HCIgestures dataset did reach the target precision, see Figure 5.5, while it does not come close for the full annotation. This difference could be due to the convergence estimation for the number of labelled samples, as was also discussed for Prototype I, see Section 4.5. For the cascading experiments, convergence estimation is used, but for the CCP experiments, the number of labelled samples is fixed at 200. The difference could also be due to the system having to deal with more difficult samples as the annotation proceeds. During the CCP experiments, only the first annotation round is considered.

5.5 Conclusion

The research done for Prototype II revolved around RQ 3.1 and 3.2, see Section 1.4. The former questions whether it is possible to identify the data points the MBC classifies correctly. The latter questions whether these misclassified data points can be corrected.

5.5.1 Prediction of classification correctness (RQ 3.1)

The CCP is not able to identify correctly classified samples for every dataset and desired precision, in the scenario where 200 samples are labelled. The CCP has trouble reaching the target precision, especially for the FoG and Aeres datasets. Different choices of metrics do not seem to have a big effect.

The difficulties relating to meeting the precision target seem to be mainly due to a lack of representativeness of the test data in the validation data. As the size of the validation set increases, the precision target is more easily met. However, as a consequence, the system becomes slower. Therefore, in future research, ways to obtain a more representative labelled dataset, through more advanced active learning methods, should be explored.

The recall is quite low for the current CCP setup regardless of the metric. When setting the CCP threshold using the test data, the recall is higher for the ODIN metric. Hence, if the issues with meeting the target precision are solved, it is recommended to use ODIN.

In answer to RQ 3.1, the system is currently not able to identify correctly classified data points at all desired precision levels. For some datasets, the precision is still very low and is not expected to meet user expectations.

5.5.2 Correcting misclassified data points (RQ 3.2)

Using the cascading classification approach, the full dataset can be annotated while attempting to control the quality with the classification correctness prediction. The advantage of using the cascading system is that the MBC only needs to train on limited labelled samples at a time. This makes the system faster, which is needed for the user interactions. Currently, there are still issues with obtaining the desired quality.

For most datasets the system still faces issues. This is mainly due to the CCP not being able to meet the desired precision. The datasets for which the system does perform well are also the datasets for which the MBC reaches the highest accuracy. Hence, improving the performance of the MBC on the other datasets may increase the overall performance. Furthermore, based on the conclusions for the CCP, the system performance will likely increase when specifically selecting representative samples for manual labelling.

The same conclusion can be drawn for RQ 3.2 as for RQ 3.1. Too many incorrectly classified data points are allowed through the CCP. For most datasets, this results in a precision that does not meet all desired levels of quality.

6 RESULTS

In Figures 6.1 and 6.2 the comparison between Prototype I and Prototype II is shown as the averages and standard deviations over five runs. For the legend of the datasets, see Figure 5.4 in Section 5.3. For Prototype I, the hyperparameters discussed in Section 4.2.1 are used, for Prototype II, the hyperparameters discussed in Section 5.2 are used. The additional settings used for Prototype II can be found in Table 6.1.

Table 6.1: Settings used for Prototype II in the final results.

Option	Setting
CCP metric	Basic maximum softmax output
Threshold setting	Extreme
Per class/single threshold	Single
Convergence objective	MBC accuracy
Discarding samples	Discard all

The manual effort is higher for Prototype II, less samples are automatically labelled, see Figure 6.2. However, the goal for Prototype II was to restore the quality lost for Prototype I. As can be seen in Figure 6.1, the precision has indeed increased for Prototype II. However, the target precision is still not always met.

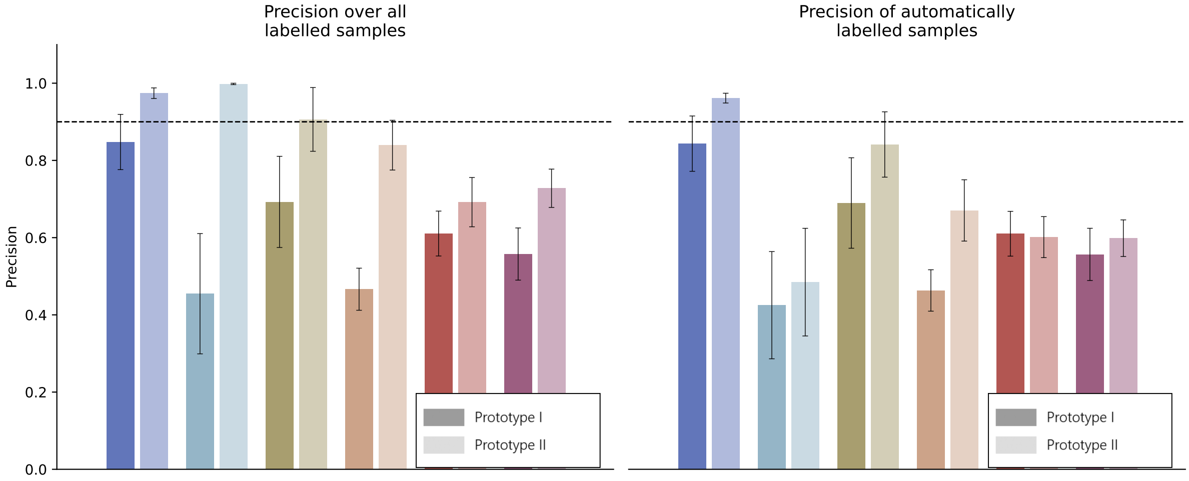


Figure 6.1: Annotation precision reached for Prototype I and Prototype II. The overall and automatic precision are shown. Note, the precision is weighted per class.

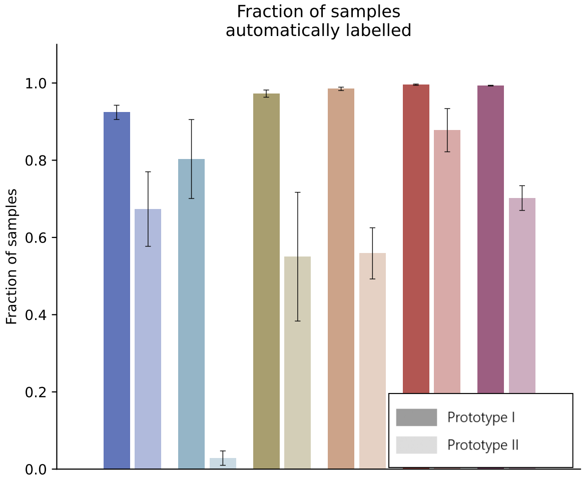


Figure 6.2: Fraction of samples automatically labelled for creating the annotation for Prototype I and Prototype II. Note, the number of samples is **not** weighted per class.

7 USER STUDY

A user study is held to gather feedback on Prototype II. Ethical approval for this user study has been granted by the Ethics Committee Computer & Information Science at the University of Twente under reference number RP 2022-103. The information brochure and informed consent form used can be found in Appendices E.1 and E.2.

7.1 Method

The user study presented in this section is similar in methodology to the pilot user study for Prototype I. Users are again observed while they are using the tool, followed up by an interview. While users are testing the tool, they are asked to follow the think-aloud protocol. The interview questions can be found in Appendix E.5 and are similar to those used for Prototype I.

7.1.1 Participants

Participants are again recruited from employees at Noldus IT. The sample size is three, due to time constraints. Participants should have experience with behavioural annotation.

7.1.2 Instructions

Since users had trouble determining the behaviours of subjects in the OPPORTUNITY++ datasets during the pilot user study for Prototype I, users are given more instructions regarding the annotation. They are told that if the subject moves their legs, but stays in place, the sample should be labelled as ‘Stand’. If they move out of place, it should be labelled as ‘Walk’. Crouching should be considered to be ‘Stand’. Users are also shown a few examples of difficult behaviours.

Users are told about the ‘Other’ and ‘Unknown’ options but are also told they should try to choose one of the other categories if at all possible. This choice is made since all video clips shown do have ground truth labels in the OPPORTUNITY++ dataset. Hence, it should be possible to label all video clips using the four behaviours.

In the pilot user study for Prototype I, it was found that users would like some more information regarding the workings of the tool, see Section 4.5. Therefore, more information was given, the full instructions can be found in Appendix E.3.

7.1.3 Experimental setup

The number of labelled samples gathered from the user during each active learning iteration is $x = 18$. Since it would be unfeasible to ask participants to label big amounts of data for this user study, the total number of samples is limited. The number of labelled samples users need to label is fixed at 108. The CCP step is fixed to take place only once, after the user has labelled these 108 samples.

While the number of labelled samples is fixed, the active learning iterations do take place and the hyperparameters from the best performing active learning iteration are used. This is done to give a fair experience of the possible slowing down that occurs as more samples are labelled.

The target precision is set to 0.8, to avoid situations where no automatic samples are labelled. If such a situation does occur, users will be shown an example result, created based on manual labels by the researcher, these may be different from the labels in the OPPORTUNITY++ dataset.

For the user study, the tool is run on an Intel Core i7 CPU without GPU support. The current implementation does not support parallel hyperparameter tuning. Therefore, the main overhead, especially when the model has few samples available, is sequential.

Users again use a subset of the OPPORTUNITY++ dataset due to size restrictions of the current implementation of the tool. ADL (activities for daily living) runs 2, 5 and 2 were randomly selected for subjects 2, 3 and 4.

7.2 Numerical results

A detailed overview of the numerical results of the user studies can be found in Appendix E.4. Interestingly, the samples that were automatically labelled for each user were all annotated with a precision of 1.0, more on this is discussed in Section 8.2. The rater-reliability varied between 0.71 and 0.85, which is considered moderate to strong agreement.

For user 2, no samples were automatically labelled. Therefore, such that they would understand how the tool is supposed to work, they were shown an example result, see Appendix E.4.

7.3 Feedback

In this section, the feedback gathered during the final user studies is summarized. Feedback is based on observations made by the researcher, statements made by users during the think-aloud protocol and the interviews. Feedback is described in the form of quotes or paraphrased opinions. Transcripts of the think-aloud protocol and the interviews can be found in Appendix E.6.

Feedback regarding the general workings of the tool can be found in Table 7.1. Feedback on the efficiency and effectiveness can be found in Tables 7.2 and 7.3. Feedback on the video clips of the data samples can be found in Table 7.4. There was also feedback regarding details for the UI, details can be found in Appendix E.7. The main response regarding the UI is that while the interface is found to be quite straightforward, it can work more efficiently.

Interestingly, user 1 mentioned that they noticed they were influenced by thinking about what the model would expect. “With the knowledge of humans I think, he will take a step. But I already take into account that I think the model will not understand it if I classify it as Walk.’

Table 7.1: General feedback

Positive points	User
“The idea is perfect.”	3
Found the tool (quite) straightforward.	1, 2
Thought the tool works well.	3
Liked the interactive aspect.	1
Issues	
Noticed they were interpreting samples keeping the AI in mind.	1
“I am not convinced with the results as they are now.”	2
“I would start doubting how efficient the tool is when it starts making mistakes.”	3
Suggestions	
Would like to change the precision target, to see what happens and decide what to do.	1, 2
“I would like to browse myself.”	1
“Easier if transitions could be put into one pile.”	1
Sort data samples based on confidence.	1
Option to add labels while annotating.	3
General observations/statements	
Did not use batch labelling.	3
If waiting time between batches gets long it would take me out of the flow.	1, 3

Table 7.2: Efficiency

Issues	User
Would get discouraged if they would still need to label this many samples.	2
“If I need to label 70-80%, it is better than nothing, but it also needs to feel like it’s better.”	3
If there is too much data you cannot inspect everything and need to trust the system.	3
“When would you trust the model to be good enough? Do I need to verify everything?”	2
Once there are mistakes you need to look at everything to get 100% accuracy.	1
The more there is labelled, the more is expected to be labelled automatically.	1, 2, 3
Would not expect to see videos from the same sequence for labelling.	2
Expects to only be shown the uncertain samples, others should be labelled correctly.	2
General observations/statements	
Willing to label 50% of the dataset (but depends on the size of the dataset).	1, 3
Would not want to label more than 10% of the dataset.	2
Verification is faster than manual annotation (to some extent).	1

Table 7.3: Effectiveness

Issues	User
In the beginning, you start trusting the automatic labels, later samples might get more difficult and mistakes slip in.	1
The precision target of 80% does not tell me much.	2, 3
Expects people to set the precision target to 100% and wonder why it is not working.	3
“If I label one more sample I would expect more to be labelled.”	3
“100% accuracy is not possible, but keeping 95% would be nice. It needs to be accurate”	3

Table 7.4: Video clips

Positive points	User
Appropriate clip length for this dataset.	1, 2, 3
Issues	
Difficult to determine whether ‘Stand’ or ‘Walk’.	1, 2
Worked with videos of 2 seconds for other datasets in the past, which was difficult for rodents.	2
Suggestions	
Show the middle of videos as thumbnail images.	3
Use the start of the clip as the point to label [user indicated context beforehand is not needed].	3

8 DISCUSSION

8.1 Results

The purpose of Prototype II was to regain the quality lost in Prototype I. As can be seen in Figure 6.1, although the target precision is not generally met yet, the overall precision for Prototype II is indeed higher than for Prototype I for all datasets. This comes at the cost of the user generally having to label more samples by hand, see Figure 6.2.

For the overall precision, Prototype II reaches the precision target for the BAfitness, HCIgestures and OPPORTUNITY AR datasets. For the precision of the automatically labelled samples, the precision target is only generally met for the BAfitness dataset. If the CCP would work as desired, the target precision should be met for the automatic case since it is only applied for the automatic labels. The overall precision of the HCIgestures and OPPORTUNITY AR datasets meets the target precision due to high numbers of manually labelled samples, see Figure 6.2.

As discussed in Section 5.3.1, the main cause of the precision target not being met seems to be that the validation data is not representative (enough) of the test data. In the future, more advanced active learning methods should be employed to increase the representativeness of the labelled data. This will be further discussed in Section 8.3.

A related issue is the convergence estimation. As discussed in Section 5.3.1, the CCP performs better when more data is available than is currently estimated. A disadvantage of having more labelled samples per cascade level is that the system slows down. Another option could be to select a fixed number of labelled samples based on a trade-off between the increase in performance and the increase in computational cost. This will be discussed in Section 8.3.

In the current approach, users still need to label a sizeable proportion of the dataset. While anything that is labelled automatically can be seen as a win, higher efficiency is desired. As also came up during the user study (see Section 7.3), users have high expectations of the efficiency that should ideally be matched.

The efficiency could also be improved by increasing the performance of the MBC. Such an improvement was also seen in Section 5.3 when more data was made available for the MBC. It is expected that when more advanced active learning methods are used, the MBC performance will also be improved. It might be possible to even further improve the performance. For example, a method from few-shot learning [24] could perhaps reach higher performance than MiniROCKET since few-shot learning specifically aims to train models with very little data, as is the case for this research.

From the results in Section 5.3, it was found that ODIN has the highest recall, given that the precision threshold can be accurately set. Therefore, it is recommended that when the issues regarding the meeting of the precision target are solved, ODIN is used for the CCP instead of the basic maximum softmax output metric.

8.2 User study

8.2.1 Deviations from Prototype II

After the user study was performed it was found that there was an issue with the code, causing the beginning or end of some segments to contain measurements from earlier or farther along in the accelerometer signals. However, the difference in performance was slight when comparing automatic simulations. For the tool with the bug, a precision of 83.54 ± 13.65 and a recall of 33.91 ± 23.95 were measured over 5 runs. For the correct tool, a precision of 83.79 ± 10.93 and a recall of 20.44 ± 12.81 was measured.

Furthermore, differences in performance should not have a big impact on the results of the user study. The user study mainly revolved around the overall concept of the tool and the users' future wishes regarding efficiency and effectiveness. Therefore, just giving a preview of the tool should be enough.

8.2.2 Numerical results

During the actual user studies, the precision was 1.0 for all users and the recall was rather low. This is unexpected compared to the automatic simulations. To verify the workings of the prototype, in Appendix E.4 the results of two annotations made by the researcher are presented. For one of these annotations, the precision is not 1.0 and the recall is higher.

The rater-reliability between the users and the ground truth varied between 0.71 and 0.85. This shows there is no perfect match. For the ground truth, the original annotations for the OPPORTUNITY++ dataset were used, rather than annotations made by the researcher. A difference in labelling has likely led to the difference in performance. The difference in labelling could be due to the different approach to labelling: labelling clips rather than full data streams. It could also be due to a different interpretation of the behaviours.

8.2.3 Feedback

Users expressed concerns regarding obtaining a fully correct annotation. User 1 noted that mistakes might slip in later during the annotations. This could indeed be the case since as the annotation progresses, the difficult samples remain. As a consequence, the system might start making more mistakes. At this point, a user might have already started trusting the system and stopped verifying the automatic labels. In future research, it would be interesting to analyze whether this drop in precision actually occurs as the annotation proceeds.

Users noted or worried that when a precision of 100% is required, you need to verify all automatic annotations. For the tool in its current state, this is indeed the case. The system aims to reach the precision target, but no guarantees can be made. User 3 doubted that the tool would still be efficient when it starts making mistakes. However, as user 1 also noted, verification should always be faster than manual annotation.

Users 2 and 3 found it difficult to interpret the target precision and thought it would lead to difficulties for other users. Therefore, instead of referring to the target precision in the UI, it could be an idea to give users a slider between 'high efficiency' and 'high quality'. This would make the trade-off easier to interpret for users.

Users may have high expectations of the efficiency of the tool. User 2 expressed that they would not want to label more than 10% of the dataset, however the other users would be willing to label half of the dataset. For the datasets where the overall precision meets the precision target, the fraction of automatically labelled samples is on average about 67% for the BAfitness dataset, 55% for the OPPORTUNITY AR dataset and 3% for the HCIgestures dataset. Hence, the BAfitness and OPPORTUNITY AR datasets do meet the expectations of user 1 and user 3, while not coming close to the expectations of user 2.

The scale of the user study is small. Therefore, no conclusion can be drawn regarding user expectations. However, in general, it is expected that there will be users with high expectations regarding the increase in efficiency of a semi-automatic tool. Furthermore, the higher the efficiency, the higher the general usability is expected to be. Therefore, the system would still benefit from an increase in efficiency, this will be discussed in Section 8.3.

Whether the computation time of MiniROCKET is fast enough is a question that remains open for RQ 1.1, see Section 4.5. For Prototype II, none of the users mentioned having to wait. However, users only labelled 108 samples, the lag might become more noticeable as more samples are labelled. To further answer RQ 1.1, more detailed research is required. Nonetheless, users not seeming to notice any waiting time up to these 108 samples, does seem to indicate that the machine learning update is fast enough.

8.3 Future research

User interaction

For the user interface, users are only shown progress when a new batch of samples is automatically labelled. This occurs at the end of the active learning iteration, for example after 200 samples. This means that it takes a long time before users get any sense of progress on the automatic annotations. During the user study, it was not tested how users would respond to not knowing when a new machine learning update occurs. Since users were informed of this. Furthermore, it already occurred after 108 labelled samples.

For Prototype I, users expressed that they liked the quality estimates. For Prototype II comparable estimations cannot be used. The accuracy of the MBC is of limited relevance due to the extensions with the CCP. The validation precision of the CCP does not correspond to the test precision, see Figure 5.8. It could be explored whether the recall can be estimated at earlier stages during the active learning loop. Furthermore, it might also be possible to inform the user about the convergence of the active learning iteration.

It could be informative for users to show confidence scores for the automatically labelled samples. This informs them of the model's understanding. An issue would be that the certainty estimation values do not necessarily carry inherent meaning. For one model a certain maximum softmax output may be high, while that value is low for another model. It might be possible to scale the metrics based on values in the validation set.

Users expressed they would like more interaction regarding the samples that are automatically labelled. For example, by changing the precision target and seeing what happens or choosing for themselves which samples to label. Users could be shown video clips ranked by certainty and they could correct clips that go wrong. Cueflik [94] is an example of a system that works comparably. Such an approach would likely increase user satisfaction. However, it comes with the danger of including bias in the labelled data, due to the user being in charge of choosing which samples to label.

The way samples are currently labelled, with multiple video clips at a time, may need to be reconsidered. Both during the pilot user study and the final user study, multiple users specified that they are not interested in batch labelling. If only one video is shown at a time, the user will be able to have a better look at this video and be less distracted. It could be an option to let users switch between two modes, batch labelling and single clip labelling, based on their preferences.

Fixed number of labels

Instead of using convergence estimation, it could be possible to train models using a fixed number of labelled samples per cascade level. The user could make a trade-off between the effort of labelling more samples and having a higher performance.

When a fixed number of samples N is chosen, the machine learning back end will be faster during the labelling of batches. No intermediate active learning iterations would need to be performed, models only need to be trained every N samples. Hence, bigger values of N could be chosen without continuously disrupting the labelling process. The user could be suggested to take a break after labelling N samples and come back later to label the next N. Unfortunately, this advantage could be lost if, in future research, a different active learning approach than random sampling would be used.

Active learning

If in the future active learning is incorporated into the tool, this should be done carefully, keeping the k-fold cross validation in mind. The cross validation requires a representative distribution of the labelled data. Many active learning methods, like uncertainty sampling, do not necessarily choose samples that best represent the data distribution.

Instead, it is recommended to research representativeness-based active learning. This could work by pre-training an embedded space and clustering the resulting representations [95]. The clustering could be used for the active learning sampling. The embedded space could be trained using unsupervised learning techniques, like autoencoders. An additional advantage would be that the pre-training also accelerates the training of the MBC and generally reduces the number of samples required for training it.

Transfer learning

Instead of, or combined with, active learning and pre-training, the MBC performance could be improved using transfer learning. An approach similar to DeepLabCut [55] could be used. For this to work, research regarding the best choices for supporting datasets needs to be done. Since the tool needs to support a wide range of subjects, conditions and behaviours it might be possible to deliver pre-trained models for different purposes. For example by preparing human or rodent-specific networks.

Sequentiality

As came up during the user study, ideally, a sequence of video clips of the same behaviour would be labelled at once. The difficulty here is that this requires knowledge regarding the transitions. Without knowing whether a transition occurs, it is uncertain which neighbours can be labelled.

It would be an option to use data-based segmentation as opposed to sliding-window segmentation. This would ideally mean that a video clip corresponds to a single sequence. Meaning, that labelling one video clip inherently leads to classifying the whole sequence. The issue here is that this would require additional tuning of the granularity of the segmentation based on user input.

Another option would be to automatically recognize transitions. This would also require an additional model and a new type of user input. It could also be possible to include the label predictions and model certainty of neighbouring samples in the CCP prediction.

In the current method, there is no independence between the training, validation and testing data. This means that video clips from the same sequence may be used to recognize other clips in this sequence. It has not yet been explored how many video clips from data sequences of which no samples have been labelled can be annotated with high certainty. It would be interesting to verify how well the system works when the unlabelled samples are independent from the labelled samples.

Efficiency

For measuring an actual increase in efficiency compared to regular manual annotation, more extensive user studies should take place. In these user studies manual annotation, for example, using The Observer[1] should be compared with semi-automatic annotation using the Smart Annotation Tool. Manual annotation currently takes place in a rather different way from the annotation process in the Smart Annotation Tool. Generally, annotation is done by labelling segments directly from a video stream instead of by labelling batches of video clips. Therefore, it is difficult to make any estimation of the actual increase in efficiency provided by the Smart Annotation Tool.

8.4 Further application

The use of the methods presented in this report is not limited to behavioural annotation. The system can be adapted to any classification task that requires high precision but has a limited labelling cost. The MiniROCKET classifier can be used for any numeric time series data. Furthermore, the MiniROCKET classifier can be switched with a different model for classification to make the system applicable for different classification tasks.

Annotations created with the tool can also be used to accelerate the development of AI in general. The tool can be used to quickly annotate datasets. These datasets can in turn be used to train new AI models. Note, annotations obtained through the methodology presented in this report are not to be used for the development of any malicious AI outside the EU legislation.

8.5 Limitations of the current tool

The Smart Annotation Tool currently only uses accelerometer data while users use the video data. This could cause discrepancies between what users see and what the machine learning model ‘sees’. Currently, the solution is to inform the user of this such that they can keep this in mind. In future work, it might be an option to include video data in the machine learning back end. The disadvantage is that this could slow training down.

Currently, the video clips shown to users are all 2 seconds in duration, while users indicated that this duration was fine for the OPPORTUNITY++ dataset, see Section 4.5, this might not be ideal for some datasets. Sometimes more temporal context might be necessary. While in other cases the two seconds might make it too ambiguous to identify the behaviour occurring, due to

[1] www.noldus.com/observer-xt

transitions. Therefore, the clip duration should be further explored. It could be a good option to allow the user to adjust it.

At the beginning of this research, the choice was made to use sliding-window segmentation, see Section 2.1.4. This segmentation results in overlapping windows. However, for the Smart Annotation Tool, this is actually not be desirable. The advantage of the overlapping windows for sliding-window is an increase in data (at the cost of biasing the data). However, for the Smart Annotation Tool, the data available for training is limited by the labelled clips sampled from the pool of unlabelled data. Increasing the pool with overlapping segments does not increase the available labelled data, but does decrease the informativeness of sampled clips. Not using sliding-window segmentation would also increase the computation speed, since fewer samples would need to be processed.

As also came up during the user studies, an inherent limitation of the tool is that at the beginning of the annotation progress, users will likely be more focused to verify the automatic annotations made by the tool. As the annotation process progresses, users may start trusting the performance of the tool. However, as the process progresses, the samples encountered may also become more difficult. Hence, the tool might start making more mistakes while the user is not aware of this. This should be mitigated by informing the user they should stay vigilant.

A possible limitation of the tool is that users are influenced in their annotation process by what they expect the model 'understands', as mentioned by one of the users in the user study. For example, a user might see a subject in a video making a slight walking motion. However, they feel like the machine learning model would not recognize it as such. Therefore, they decide to label it as 'Stand'. This could impact the resulting annotation.

8.6 User recommendations

To safely employ the Smart Annotation Tool, users should always be informed of the limitations of the tool. Users should be informed that they should always verify the automatic annotations created using the tool. It could be the case that the tool makes mistakes regarding crucial annotations. Using the 'Automatic labels' tab it should be relatively easy to identify and correct any mistakes made by the Smart Annotation Tool. The user is also recommended to analyze the automatic annotations to identify possible biases or imbalances in the data.

Users should always be informed that machine learning back end only uses accelerometer data, not video data. Accelerometers should be placed on such that all relevant information is captured. However, the preliminary results in Appendix B for Prototype I suggest that using more sensors is not always better. Therefore, the placing of the accelerometers is a task that should be carefully considered.

9 CONCLUSION

This research aims to answer the research question:

> How can an interactive machine learning tool that assists researchers
> in the annotation of behaviour improve usability, compared to fully manual annotation,
> with a limited loss of quality?

The research question requires this tool to lead to an improvement in usability compared to fully manual annotation, with a limited loss of accuracy. Usability is defined as the combination of efficiency, effectiveness and user satisfaction.

The result of this research is the Smart Annotation Tool (SAT). The SAT combines a model for behavioural classification (MBC) with classification correctness prediction (CCP). The MBC should lead to an increase in efficiency, while the CCP should preserve effectiveness (e.g. quality). The level of quality that should be preserved can be controlled by the user.

For the MBC, the MiniROCKET classifier is used due to its speed and low number of parameters. MBCs are trained without the supervision of a machine learning expert, utilizing k-fold cross validation and hyperparameter optimization methods, specifically Bayesian optimization and pruning. For the CCP, thresholding is used based on uncertainty estimates of the data samples.

The CCP needs further improvement before it can be trusted to preserve quality for most datasets. However, the precision is generally higher when including the CCP. It does come at the cost of more manual labour. Including more representative data leads to better results. In future research, the CCP could be improved by incorporating more advanced active learning approaches that focus on the sampling of representative data.

The final SAT was able to label part of the dataset automatically for all datasets. As a result, the efficiency is expected to increase. To measure the actual increase in efficiency, further user studies are required. These user studies should compare manual annotation as it is currently done (for example using The Observer) with the SAT.

Users found the tool straightforward to use, however, the user satisfaction can still be improved. Users expressed concerns regarding mistakes being made by the machine learning back end. In the current stage of the tool, users should always verify the automatic annotations. In future research it would be interesting to see what happens to the user satisfaction and the performance of the MBC and CCP when users are given more control over the annotation process.

To conclude, the Smart Annotation Tool presented in this research has the potential to increase usability while preserving a controlled level of quality. With some improvements, the tool will be a valuable tool for the annotation of diverse behaviour data.

REFERENCES

[1] S. J. Pan and Q. Yang, “A Survey on Transfer Learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[2] J. Brooke, “SUS: A ’Quick and Dirty’ Usability Scale,” in *Usability Evaluation In Industry*, CRC Press, 1996.

[3] ISO, “ISO 9241-11:1998 Ergonomic requirements for office work with visual display terminals –Part 11: Guidance on usability,” 1998.

[4] D. Berndt and J. Clifford, “Using Dynamic Time Warping to Find Patterns in Time Series,” *undefined*, 1994.

[5] K.-P. Chan and A. W.-C. Fu, “Efficient time series matching by wavelets,” in *Proceedings 15th International Conference on Data Engineering (Cat. No.99CB36337)*, Mar. 1999, pp. 126–133.

[6] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[7] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[8] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, “On the Properties of Neural Machine Translation: Encoder-Decoder Approaches,” *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pp. 103–111, Oct. 2014.

[9] I. Horenko, “On a Scalable Entropic Breaching of the Overfitting Barrier for Small Data Problems in Machine Learning,” *Neural Computation*, vol. 32, no. 8, pp. 1563–1579, Aug. 2020.

[10] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, “Attention is All you Need,” *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[11] T. Wolf, L. Debut, V. Sanh, *et al.*, “Transformers: State-of-the-Art Natural Language Processing,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45.

[12] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, “A Transformer-based Framework for Multivariate Time Series Representation Learning,” in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, Virtual Event Singapore: ACM, Aug. 2021, pp. 2114–2124.

[13] H. Xu, P. Zhou, R. Tan, M. Li, and G. Shen, “Limu-bert: Unleashing the potential of unlabeled data for imu sensing applications,” in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, 2021, pp. 220–233.

[14] Y. LeCun, B. E. Boser, J. S. Denker, *et al.*, “Handwritten Digit Recognition with a Back-Propagation Network,” *Advances in neural information processing systems*, vol. 2, p. 9, 1989.

[15] A. Dempster, F. Petitjean, and G. I. Webb, “ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels,” *Data Mining and Knowledge Discovery*, vol. 34, no. 5, pp. 1454–1495, Sep. 2020.

[16] A. Dempster, D. F. Schmidt, and G. I. Webb, “MiniRocket: A Very Fast (Almost) Deterministic Transform for Time Series Classification,” in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, Virtual Event Singapore: ACM, Aug. 2021, pp. 248–257.

[17] H. A. Dau, A. Bagnall, K. Kamgar, *et al.*, “The UCR time series archive,” *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 6, pp. 1293–1305, Nov. 2019.

[18] M. Middlehurst, J. Large, G. Cawley, and A. Bagnall, “The Temporal Dictionary Ensemble (TDE) Classifier for Time Series Classification,” in *Machine Learning and Knowledge Discovery in Databases*, F. Hutter, K. Kersting, J. Lijffijt, and I. Valera, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2021, pp. 660–676.

[19] A. Shifaz, C. Pelletier, F. Petitjean, and G. I. Webb, “TS-CHIEF: A scalable and accurate forest algorithm for time series classification,” *Data Mining and Knowledge Discovery*, vol. 34, no. 3, pp. 742–775, May 2020.

[20] M. Löning, A. Bagnall, S. Ganesh, V. Kazakov, J. Lines, and F. J. Király, “Sktime: A Unified Interface for Machine Learning with Time Series,” *arXiv:1909.07872*, Sep. 2019.

[21] L. Ye and E. Keogh, “Time series shapelets: A new primitive for data mining,” in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’09*, Paris, France: ACM Press, 2009, p. 947.

[22] T. Dietterich, “Overfitting and undercomputing in machine learning,” *ACM Computing Surveys*, vol. 27, no. 3, pp. 326–327, Sep. 1995.

[23] D. Erhan, A. Courville, Y. Bengio, and P. Vincent, “Why Does Unsupervised Pre-training Help Deep Learning?” In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, JMLR Workshop and Conference Proceedings, Mar. 2010, pp. 201–208.

[24] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, “Generalizing from a Few Examples: A Survey on Few-shot Learning,” *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–34, May 2021.

[25] B. Settles, “Active Learning Literature Survey,” University of Wisconsin–Madison, Computer Sciences Technical Report 1648, 2010, p. 67.

[26] X. Zhu, “Semi-Supervised Learning Literature Survey,” *world*, no. 10, p. 10, 2005.

[27] S. Shalev-Shwartz, “Online Learning and Online Convex Optimization,” *Foundations and Trends® in Machine Learning*, vol. 4, no. 2, pp. 107–194, Mar. 2012.

[28] D. D. Lewis and J. Catlett, “Heterogeneous Uncertainty Sampling for Supervised Learning,” in *Machine Learning Proceedings 1994*, Elsevier, 1994, pp. 148–156.

[29] M. Tsiakmaki, G. Kostopoulos, S. Kotsiantis, and O. Ragos, “Fuzzy-based active learning for predicting student academic performance using autoML: A step-wise approach,” *Journal of Computing in Higher Education*, vol. 33, no. 3, pp. 635–667, Dec. 2021.

[30] M. Lorbach, R. Poppe, and R. C. Veltkamp, “Interactive rodent behavior annotation in video using active learning,” *Multimedia Tools and Applications*, vol. 78, no. 14, pp. 19 787–19 806, Jul. 2019.

[31] S. Spink, J. Kamminga, and A. Kamilaris, “Improving the Annotation Efficiency for Animal Activity Recognition using Active Learning.docx,” *Volume 2 of the Proceedings of the joint 12th International Conference on Methods and Techniques in Behavioral Research and 6th Seminar on Behavioral Methods*, May 2022.

[32] J. N. van Rijn and F. Hutter, “Hyperparameter Importance Across Datasets,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, London United Kingdom: ACM, Jul. 2018, pp. 2367–2376.

[33] X. He, K. Zhao, and X. Chu, “AutoML: A Survey of the State-of-the-Art,” *Knowledge-Based Systems*, vol. 212, p. 106 622, Jan. 2021.

[34] F. Hutter, H. Hoos, and K. Leyton-Brown, “An Efficient Approach for Assessing Hyperparameter Importance,” *International Conference on Machine Learning*, pp. 754–762, 2014.

[35] M. Feurer and F. Hutter, “Hyperparameter Optimization,” in *Automated Machine Learning*, F. Hutter, L. Kotthoff, and J. Vanschoren, Eds., Cham: Springer International Publishing, 2019, pp. 3–33.

[36] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A Next-generation Hyperparameter Optimization Framework,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD ’19, New York, NY, USA: Association for Computing Machinery, Jul. 2019, pp. 2623–2631.

[37] T. Elsken, J. H. Metzen, and F. Hutter, “Neural Architecture Search,” in *Automated Machine Learning: Methods, Systems, Challenges*, ser. The Springer Series on Challenges in Machine Learning, F. Hutter, L. Kotthoff, and J. Vanschoren, Eds., Cham: Springer International Publishing, 2019, pp. 63–77.

[38] T. Hinz, N. Navarro-Guerrero, S. Magg, and S. Wermter, “Speeding up the Hyperparameter Optimization of Deep Convolutional Neural Networks,” *International Journal of Computational Intelligence and Applications*, vol. 17, no. 02, p. 1 850 008, Jun. 2018.

[39] H. Alibrahim and S. A. Ludwig, “Hyperparameter Optimization: Comparing Genetic Algorithm against Grid Search and Bayesian Optimization,” in *2021 IEEE Congress on Evolutionary Computation (CEC)*, Jun. 2021, pp. 1551–1559.

[40] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, “Taking the Human Out of the Loop: A Review of Bayesian Optimization,” *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, Jan. 2016.

[41] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, and S.-H. Deng, “Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimizationb,” *Journal of Electronic Science and Technology*, vol. 17, no. 1, pp. 26–40, Mar. 2019.

[42] C. E. Rasmussen, “Gaussian Processes in Machine Learning,” in *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures*, ser. Lecture Notes in Computer Science, O. Bousquet, U. von Luxburg, and G. Rätsch, Eds., Berlin, Heidelberg: Springer, 2004, pp. 63–71.

[43] F. Hutter, H. H. Hoos, and K. Leyton-Brown, “Sequential Model-Based Optimization for General Algorithm Configuration,” in *Learning and Intelligent Optimization*, C. A. C. Coello, Ed., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2011, pp. 507–523.

[44] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for Hyper-Parameter Optimization,” *Advances in neural information processing systems*, vol. 24, p. 9, 2011.

[45] J. Snoek, H. Larochelle, and R. P. Adams, “Practical Bayesian Optimization of Machine Learning Algorithms,” in *Advances in Neural Information Processing Systems*, vol. 25, Curran Associates, Inc., 2012.

[46] K. Eggensperger, M. Feurer, F. Hutter, *et al.*, “Towards an Empirical Foundation for Assessing Bayesian Optimization of Hyperparameters,” *NIPS workshop on Bayesian Optimization in Theory and Practic*, p. 5, 2013.

[47] D. Golovin, B. Solnik, S. Moitra, G. Kochanski, J. Karro, and D. Sculley, “Google Vizier: A Service for Black-Box Optimization,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Halifax NS Canada: ACM, Aug. 2017, pp. 1487–1495.

[48] K. Jamieson and A. Talwalkar, “Non-stochastic Best Arm Identification and Hyperparameter Optimization,” in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, PMLR, May 2016, pp. 240–248.

[49] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, “Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization,” *The Journal of Machine Learning Research*, vol. 18, no. 1, p. 52, 2017.

[50] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, Nov. 2016.

[51] C. Yang, Y. Akimoto, D. W. Kim, and M. Udell, “OBOE: Collaborative Filtering for AutoML Model Selection,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage AK USA: ACM, Jul. 2019, pp. 1173–1183.

[52] E. LeDell and S. Poirier, “H2O AutoML: Scalable Automatic Machine Learning,” *7th ICML Workshop on Automated Machine Learning*, p. 16, 2020.

[53] M. Kabra, A. A. Robie, M. Rivera-Alba, S. Branson, and K. Branson, “JAABA: Interactive machine learning for automatic annotation of animal behavior,” *Nature Methods*, vol. 10, no. 1, pp. 64–67, Jan. 2013.

[54] S. D. Beugher, G. Brône, and T. Goedemé, “A semi-automatic annotation tool for unobtrusive gesture analysis,” *Language Resources and Evaluation*, vol. 52, no. 2, pp. 433–460, Jun. 2018.

[55] A. Mathis, P. Mamidanna, K. M. Cury, *et al.*, “Deeplabcut: Markerless pose estimation of user-defined body parts with deep learning,” *Nature Neuroscience*, 2018. [Online]. Available: `https://www.nature.com/articles/s41593-018-0209-y`.

[56] D. Hendrycks and K. Gimpel, *A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks*, Oct. 2018.

[57] T. DeVries and G. W. Taylor, *Learning Confidence for Out-of-Distribution Detection in Neural Networks*, Feb. 2018.

[58] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning,” *33rd International Conference on Machine Learning, ICML 2016*, vol. 3, pp. 1651–1660, Oct. 2016.

[59] J. Aigrain and M. Detyniecki, “Detecting Adversarial Examples and Other Misclassifications in Neural Networks by Introspection,” *arXiv:1905.09186*, May 2019.

[60] S. Liang, Y. Li, and R. Srikant, “Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks,” *6th International Conference on Learning Representations, ICLR*, 2018.

[61] A. Shafaei, M. Schmidt, and J. J. Little, *A Less Biased Evaluation of Out-of-distribution Sample Detectors*, Aug. 2019.

[62] W. Chen, R. Salay, S. Sedwards, V. Abdelzad, and K. Czarnecki, “Accelerating the Training of Convolutional Neural Networks for Image Segmentation with Deep Active Learning,” in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, Sep. 2020, pp. 1–7.

[63] Y. Gal, R. Islam, and Z. Ghahramani, “Deep Bayesian Active Learning with Image Data,” *34th International Conference on Machine Learning, ICML 2017*, vol. 3, pp. 1923–1932, Aug. 2017.

[64] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research 1*, p. 30, 2014.

[65] O. Sagi and L. Rokach, “Ensemble learning: A survey,” *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, e1249, 2018.

[66] H. S. Seung, M. Opper, and H. Sompolinsky, “Query by committee,” in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory - COLT ’92*, Pittsburgh, Pennsylvania, United States: ACM Press, 1992, pp. 287–294.

[67] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby, “Selective Sampling Using the Query by Committee Algorithm,” *Machine Learning*, vol. 28, no. 2/3, pp. 133–168, 1997.

[68] N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel, "Bayesian Active Learning for Classification and Preference Learning," *arXiv:1112.5745*, Dec. 2011.

[69] D. Munoz, J. A. Bagnell, and M. Hebert, "Stacked Hierarchical Labeling," in *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2010, pp. 57–70.

[70] G. Heitz, S. Gould, A. Saxena, and D. Koller, "Cascaded Classification Models: Combining Models for Holistic Scene Understanding," in *Advances in Neural Information Processing Systems*, vol. 21, Curran Associates, Inc., 2008.

[71] X. Wang, Y. Luo, D. Crankshaw, A. Tumanov, F. Yu, and J. E. Gonzalez, *IDK Cascades: Fast Deep Learning by Learning not to Overthink*, Jun. 2017.

[72] J. J. Dudley and P. O. Kristensson, "A Review of User Interface Design for Interactive Machine Learning," *ACM Transactions on Interactive Intelligent Systems*, vol. 8, no. 2, pp. 1–37, Jul. 2018.

[73] R. M. Monarch, *Human-in-the-Loop Machine Learning: Active Learning and Annotation for Human-centered AI*. Simon and Schuster, Jul. 2021.

[74] B. Ghai, Q. V. Liao, Y. Zhang, R. Bellamy, and K. Mueller, "Explainable Active Learning (XAL): An Empirical Study of How Local Explanations Impact Annotator Experience," *arXiv:2001.09219*, Sep. 2020.

[75] E. Horvitz, "Principles of mixed-initiative user interfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems the CHI Is the Limit - CHI '99*, Pittsburgh, Pennsylvania, United States: ACM Press, 1999, pp. 159–166.

[76] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, "Power to the People: The Role of Humans in Interactive Machine Learning," *AI Magazine*, vol. 35, no. 4, pp. 105–120, Dec. 2014.

[77] T. Kulesza, S. Stumpf, M. Burnett, and I. Kwan, "Tell me more?: The effects of mental model soundness on personalizing an intelligent agent," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Austin Texas USA: ACM, May 2012, pp. 1–10.

[78] K. Forster, D. Roggen, and G. Troster, "Unsupervised Classifier Self-Calibration through Repeated Context Occurences: Is there Robustness against Sensor Displacement to Gain?" In *2009 International Symposium on Wearable Computers*, Sep. 2009, pp. 77–84.

[79] D. Roggen, A. Calatroni, M. Rossi, *et al.*, "Collecting complex activity datasets in highly rich networked sensor environments," in *2010 Seventh International Conference on Networked Sensing Systems (INSS)*, Jun. 2010, pp. 233–240.

[80] P. Zappi, C. Lombriser, T. Stiefmeier, *et al.*, "Activity Recognition from On-Body Sensors: Accuracy-Power Trade-Off by Dynamic Sensor Selection," in *Wireless Sensor Networks*, R. Verdone, Ed., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2008, pp. 17–33.

[81] N. Giladi, T. A. Treves, E. S. Simon, *et al.*, "Freezing of gait in patients with advanced Parkinson's disease," *Journal of Neural Transmission*, vol. 108, no. 1, pp. 53–61, Jan. 2001.

[82] M. Ciliberto, V. Fortes Rey, A. Calatroni, P. Lukowicz, and D. Roggen, "Opportunity++: A Multimodal Dataset for Video- and Wearable, Object and Ambient Sensors-Based Human Activity Recognition," *Frontiers in Computer Science*, vol. 3, 2021.

[83] M. Bachlin, M. Plotnik, D. Roggen, *et al.*, "Wearable Assistant for Parkinson's Disease Patients With the Freezing of Gait Symptom," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 2, pp. 436–446, Mar. 2010.

[84] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980*, Jan. 2017.

[85] G. van Rossum, *Python reference manual*, Jan. 1995.

[86] A. Paszke, S. Gross, F. Massa, *et al.*, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” in *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019.

[87] D. Flanagan and G. M. Novak, “Java-Script: The Definitive Guide, Second Edition,” *Computers in Physics*, vol. 12, no. 1, p. 41, 1998.

[88] A. Hejlsberg, S. Wiltamuth, and P. Golde, *C# Language Specification*. USA: Addison-Wesley Longman Publishing Co., Inc., 2003.

[89] R. D. Hipp, *SQLite*, version 3.31.1, 2020. [Online]. Available: `https://www.sqlite.org/index.html`.

[90] J. Cohen, “A Coefficient of Agreement for Nominal Scales,” *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, Apr. 1960.

[91] M. L. McHugh, “Interrater reliability: The kappa statistic,” *Biochemia medica*, vol. 22, no. 3, pp. 276–282, 2012.

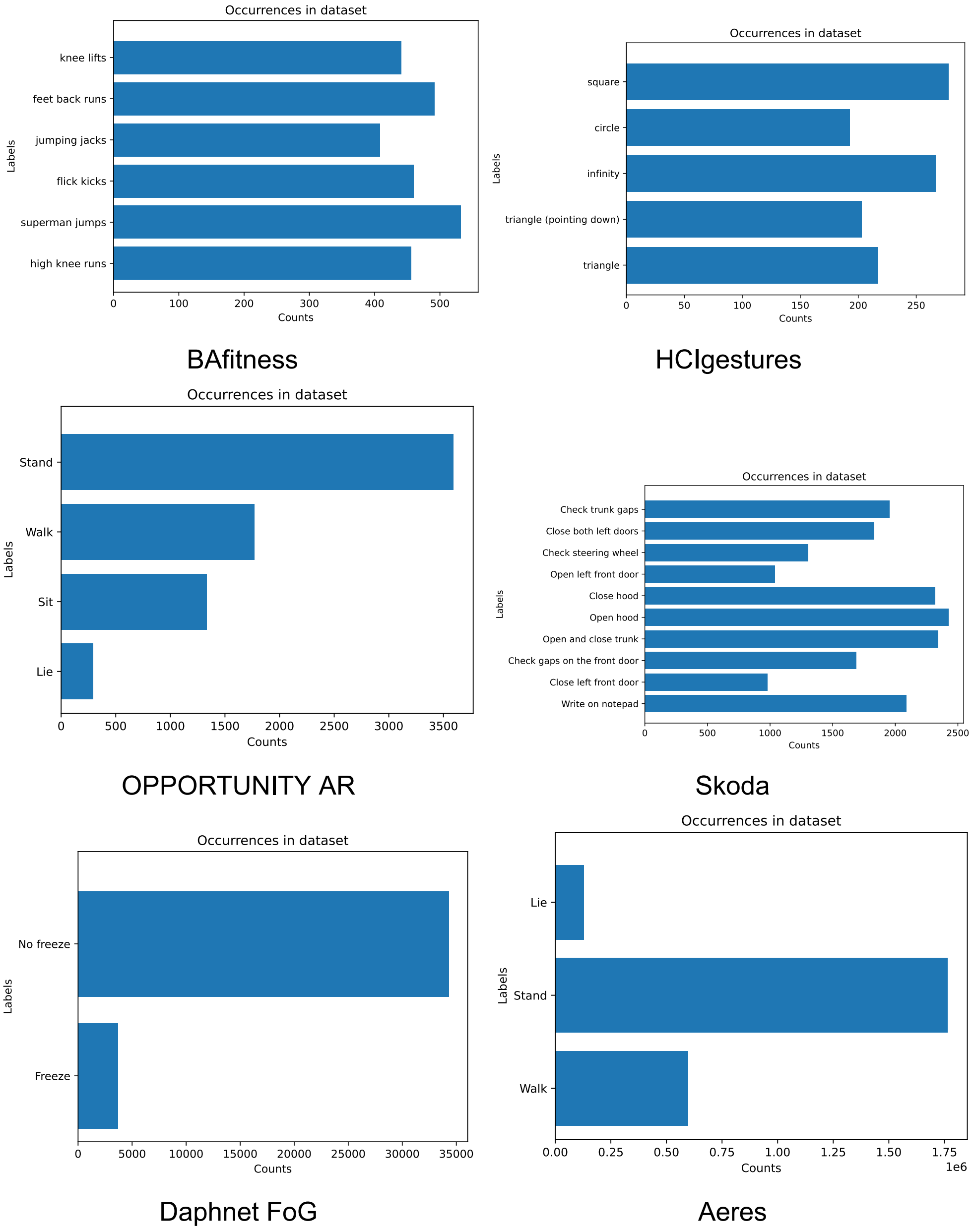
[92] M. Buckland and F. Gey, “The relationship between Recall and Precision,” *Journal of the American Society for Information Science*, vol. 45, no. 1, pp. 12–19, 1994.

[93] D. H. Lee, “Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks,” *Workshop on challenges in representation learning, ICML*, vol. 3, no. 2, p. 896, Jul. 2013.

[94] J. Fogarty, D. Tan, A. Kapoor, and S. Winder, “CueFlik: Interactive concept learning in image search,” in *Proceeding of the Twenty-Sixth Annual CHI Conference on Human Factors in Computing Systems - CHI ’08*, Florence, Italy: ACM Press, 2008, p. 29.

[95] C. Wan, F. Jin, Z. Qiao, W. Zhang, and Y. Yuan, “Unsupervised active learning with loss prediction,” *Neural Computing and Applications*, Sep. 2021.

A DATASET DISTRIBUTION



BAfitness

HCIgestures

OPPORTUNITY AR

Skoda

Daphnet FoG

Aeres

Figure A.1: The distribution of the datasets. Occurrences are based on the sliding-window segmentation with a step size of 1 second and a window size of 2 seconds.

B DIFFERENCE SENSORS

The results of an early version of Prototype I on the different datasets when using data from a single or multiple sensors can be found in Table B.1. As can be seen, for all datasets except the BAfitness dataset, the differences are slight. Note, the Aeres dataset only contains the data from a single sensor.

Table B.1: The test accuracy of an early version of Prototype I on the different datasets, where the data from either one or multiple sensors is used.

	Test accuracy	Number of labelled samples
BAfitness (1 sensor)	**85.69 ± 3.69%**	**311.80 ± 67.35**
BAfitness (multi sensor)	62.04 ± 18.05%	267.80 ± 56.88
HCIgestures (1 sensor)	38.52 ± 9.77%	162.80 ± 32.21
HCIgestures (multi sensor)	**47.68 ± 9.01%**	**199.80 ± 75.45**
OPPORTUNITY AR (1 sensor)	61.63 ± 6.16%	195.00 ± 43.83
OPPORTUNITY AR (multi sensor)	**75.20 ± 8.13%**	**198.00 ± 36.38**
Skoda (1 sensor)	**64.22 ± 7.36%**	**167.20 ± 50.96**
Skoda (multi sensor)	65.89 ± 5.93%	164.00 ± 42.71
Daphnet FoG (1 sensor)	**53.76 ± 5.00%**	**238.20 ± 55.73**
Daphnet FoG (multi sensor)	54.44 ± 4.12%	347.20 ± 57.49
Aeres (1 sensor)	**53.75 ± 6.86%**	**210.80 ± 115.10**

C PILOT USER STUDY

C.1 Changes Prototype I

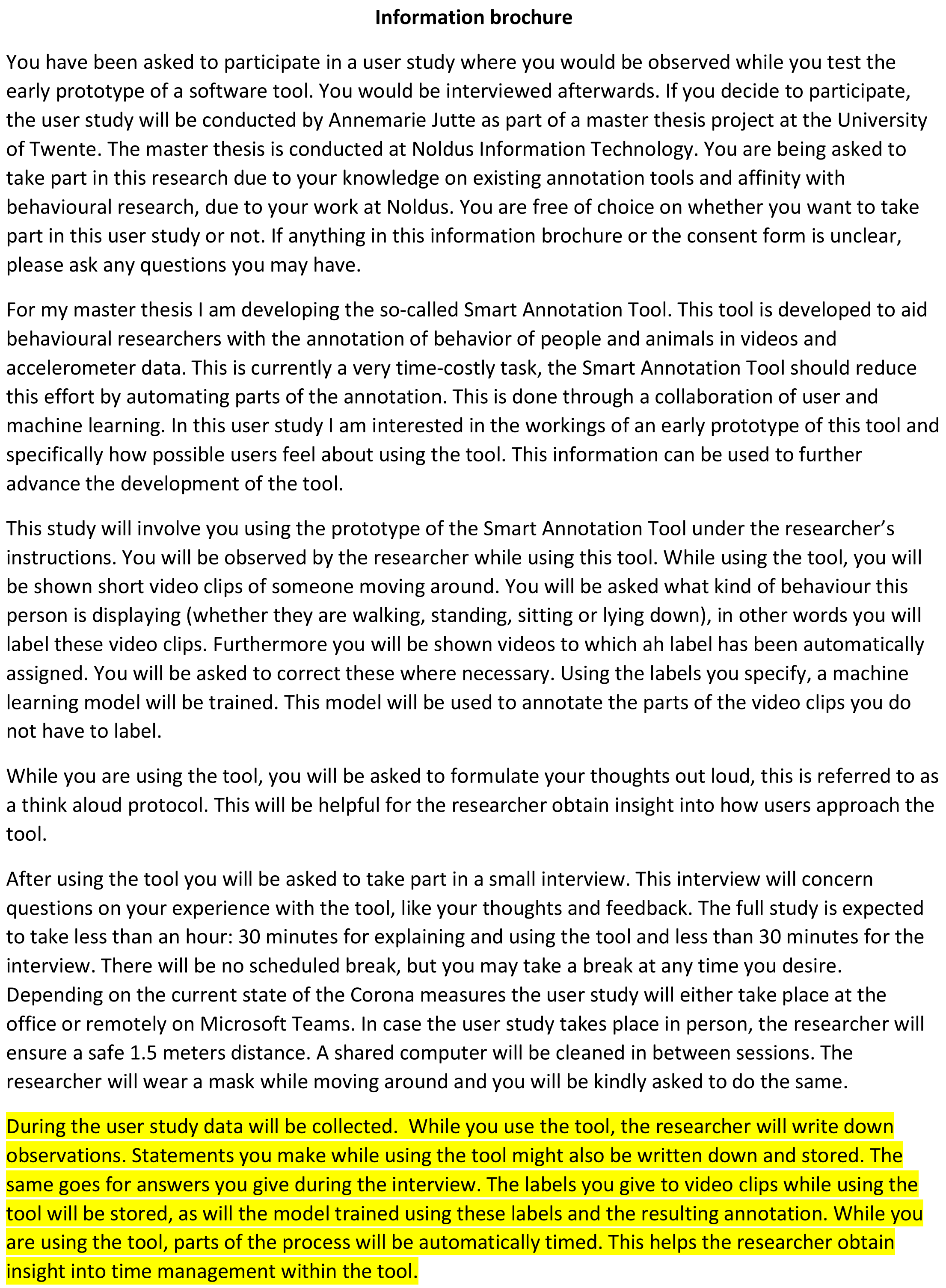
For the pilot user study conducted for Prototype I, there are some small differences to the workings of the annotation system compared to the methodology presented in Section 4.1. This is due to the pilot user study having taken place in an earlier stage of the research.

For Prototype I, when a class is not present in a fold during cross validation, the associated accuracy in the weighted accuracy is set to 0, see Section 4.1.5. For the user study, the class was skipped entirely, this gives a too optimistic estimate.

For the number of labelled samples, the performance was considered converged when the mean over the past 10 epochs was not higher than the 10 epochs before, for 5 epochs in a row. However, this can lead to states where convergence is never reached due to constant fluctuation. Furthermore, the for the user study performance was measured every epoch, instead of every 5.

For the user study, the number of tuning trials is fixed at 25, instead of being measured online based on convergence. Furthermore, Hyperband tuning is used instead of the median stopping rule.

C.2 Information brochure



Information brochure

You have been asked to participate in a user study where you would be observed while you test the early prototype of a software tool. You would be interviewed afterwards. If you decide to participate, the user study will be conducted by Annemarie Jutte as part of a master thesis project at the University of Twente. The master thesis is conducted at Noldus Information Technology. You are being asked to take part in this research due to your knowledge on existing annotation tools and affinity with behavioural research, due to your work at Noldus. You are free of choice on whether you want to take part in this user study or not. If anything in this information brochure or the consent form is unclear, please ask any questions you may have.

For my master thesis I am developing the so-called Smart Annotation Tool. This tool is developed to aid behavioural researchers with the annotation of behavior of people and animals in videos and accelerometer data. This is currently a very time-costly task, the Smart Annotation Tool should reduce this effort by automating parts of the annotation. This is done through a collaboration of user and machine learning. In this user study I am interested in the workings of an early prototype of this tool and specifically how possible users feel about using the tool. This information can be used to further advance the development of the tool.

This study will involve you using the prototype of the Smart Annotation Tool under the researcher’s instructions. You will be observed by the researcher while using this tool. While using the tool, you will be shown short video clips of someone moving around. You will be asked what kind of behaviour this person is displaying (whether they are walking, standing, sitting or lying down), in other words you will label these video clips. Furthermore you will be shown videos to which ah label has been automatically assigned. You will be asked to correct these where necessary. Using the labels you specify, a machine learning model will be trained. This model will be used to annotate the parts of the video clips you do not have to label.

While you are using the tool, you will be asked to formulate your thoughts out loud, this is referred to as a think aloud protocol. This will be helpful for the researcher obtain insight into how users approach the tool.

After using the tool you will be asked to take part in a small interview. This interview will concern questions on your experience with the tool, like your thoughts and feedback. The full study is expected to take less than an hour: 30 minutes for explaining and using the tool and less than 30 minutes for the interview. There will be no scheduled break, but you may take a break at any time you desire. Depending on the current state of the Corona measures the user study will either take place at the office or remotely on Microsoft Teams. In case the user study takes place in person, the researcher will ensure a safe 1.5 meters distance. A shared computer will be cleaned in between sessions. The researcher will wear a mask while moving around and you will be kindly asked to do the same.

During the user study data will be collected. While you use the tool, the researcher will write down observations. Statements you make while using the tool might also be written down and stored. The same goes for answers you give during the interview. The labels you give to video clips while using the tool will be stored, as will the model trained using these labels and the resulting annotation. While you are using the tool, parts of the process will be automatically timed. This helps the researcher obtain insight into time management within the tool.

Furthermore, you will be asked for your area of work and experience with behavioural annotation, this data will be stored with the other data. You will be asked whether your screen may be recorded. This would be useful for processing the results. You are free to deny this. Your personal data will not be shared with any third parties.

From the data collected, the following may appear in the researcher's master thesis and related reports on the Smart Annotation Tool: quotes from the interview, statements made by you, observations made by the researcher, labels you gave video clips, statistics on the annotation created, durations of parts of the annotation process and anonymized screen grabs (the latter only in case you consent to having your screen recorded). This data will only appear fully anonymized.

In case you allow recording of your screen, the recordings would be useful for making more detailed observations on possible issues with the tool. Furthermore, (anonymized) screen grabs could be extracted for the thesis report. After processing the recordings, the recordings will be deleted.

All data collected during the user study will be removed before the end of my master thesis, besides the data included in reports (see the previous paragraph) and used to further the design of the Smart Annotation Tool. The removal will take place at least before 31-12-2022, but earlier if the thesis is already finished.

You may withdraw from this study at any time, with no further questions asked. After the study your data can be used for the further development of the Smart Annotation Tool. The data can be used in the master thesis report relating to this research. Furthermore, any data published in the master thesis report may be used in any reports relating to the development of the Smart Annotation Tool. Only anonymized data will be published. After the data has been published it may not be removed from reports or products upon withdrawal. However, any other data stored will be removed when requested (and before finishing the thesis).

If you have any questions at any point do not hesitate to contact the researcher of this study right now or later, using the contact information below.

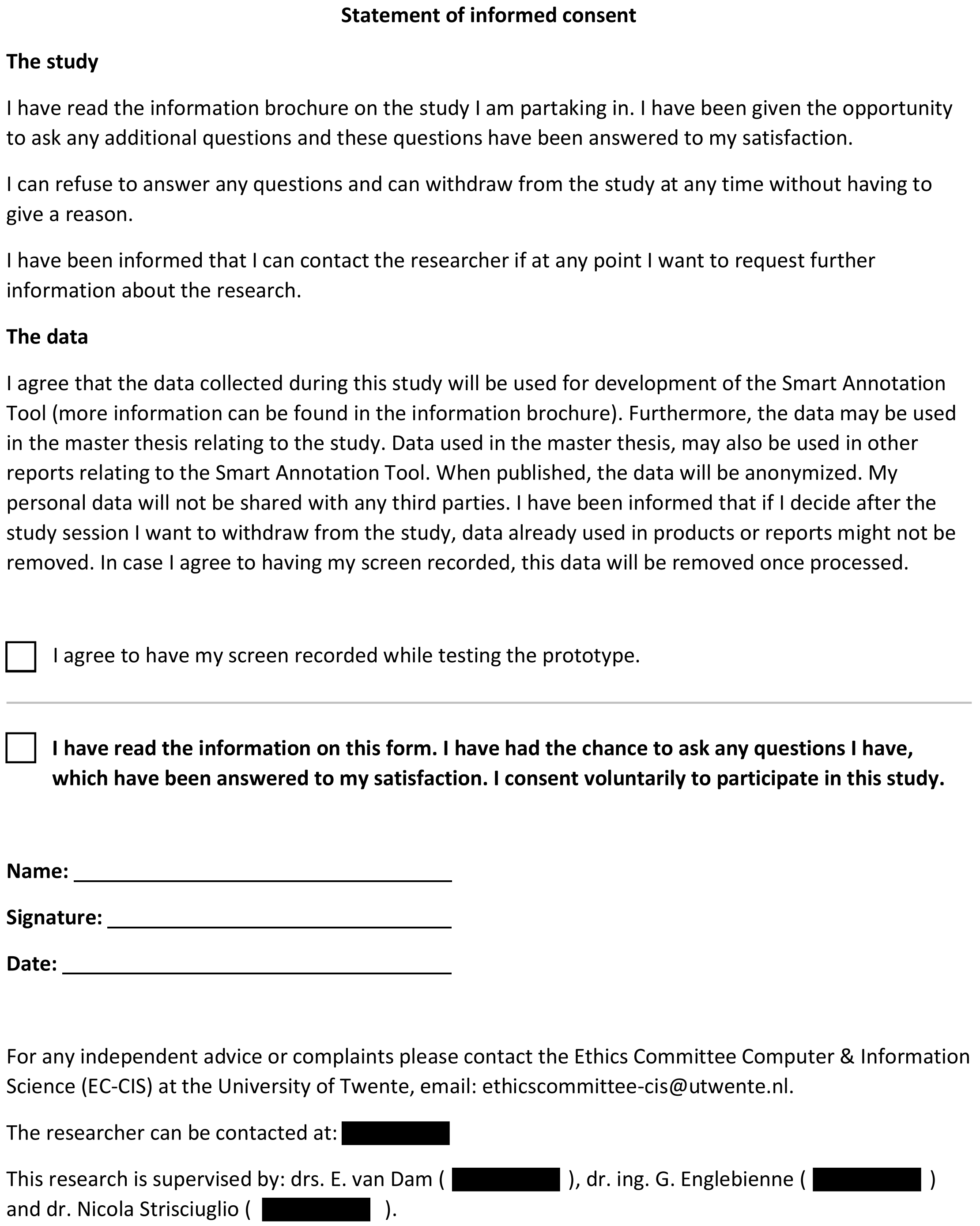
Thank you for your interest in this study.

Annemarie Jutte,
[redacted]

For any independent advice or complaints please contact the Ethics Committee Computer & Information Science (EC-CIS) at the University of Twente, email: ethicscommittee-cis@utwente.nl.

This research is supervised by: drs. E. van Dam ([redacted]), dr. ing. G. Englebienne ([redacted]) and dr. Nicola Strisciuglio ([redacted]).

C.3 Informed consent form



Statement of informed consent

The study

I have read the information brochure on the study I am partaking in. I have been given the opportunity to ask any additional questions and these questions have been answered to my satisfaction.

I can refuse to answer any questions and can withdraw from the study at any time without having to give a reason.

I have been informed that I can contact the researcher if at any point I want to request further information about the research.

The data

I agree that the data collected during this study will be used for development of the Smart Annotation Tool (more information can be found in the information brochure). Furthermore, the data may be used in the master thesis relating to the study. Data used in the master thesis, may also be used in other reports relating to the Smart Annotation Tool. When published, the data will be anonymized. My personal data will not be shared with any third parties. I have been informed that if I decide after the study session I want to withdraw from the study, data already used in products or reports might not be removed. In case I agree to having my screen recorded, this data will be removed once processed.

□ I agree to have my screen recorded while testing the prototype.

□ **I have read the information on this form. I have had the chance to ask any questions I have, which have been answered to my satisfaction. I consent voluntarily to participate in this study.**

Name: ___________________________________

Signature: ________________________________

Date: ____________________________________

For any independent advice or complaints please contact the Ethics Committee Computer & Information Science (EC-CIS) at the University of Twente, email: ethicscommittee-cis@utwente.nl.

The researcher can be contacted at: [redacted]

This research is supervised by: drs. E. van Dam ([redacted]), dr. ing. G. Englebienne ([redacted]) and dr. Nicola Strisciuglio ([redacted]).

C.4 Instructions

Note, the Dutch translation of the instructions is used during the user study.

We are here today to test the first prototype of the Smart Annotation Tool. The idea of the Smart Annotation Tool is to speed up the traditional method of behavioural annotation, where you need to go from front to back through all the data and precisely annotate the behaviour. Instead of labelling all this data by hand, only a small part of the data needs to be labelled. All the other data will be done using artificial intelligence.

When using the tool you will be presented by short video clips, to which you need to assign the appropriate labels. The AI behind the SAT will use the labels you have given and learn to give similar data the same labels. This is a repeated progress. The tool will ask you for more labels, you will give more labels and the AI will get better at recognizing behaviours. At the end of this progress the AI you have trained will be used to create the final annotation of the dataset. The AI will also learn for itself to estimate how good its annotation is, this estimate will get better as you have labelled more samples.

Today we are going to label a dataset for which various locomotion tasks need to be annotated. The subject of which we want to create the annotation is the person with the blue jacket. You will be presented with short video clips of this person. You can assign one of 4 behaviours for each video clip: walking, standing, sitting and lying. Try to choose the best fitting behaviour where possible. If multiple behaviours occur in a single clip, try to choose the behaviour that occurs in the middle of the clip. If this is not an option or if a different behaviour occurs from the four that we are annotating, please select the ‘Other’ behaviour. If you cannot see what behaviour occurs, due to, for example, objects blocking the view, please select the ‘Unknown’ behaviour.

There will be three tabs in the annotation program: ‘to label’, ‘manual labels’ and ‘automatic labels’. The ‘to label’ tab is where you are presented with the video clips the program wants you to label. This is the most important tab. The ‘manual labels’ tab is where video clips previously labelled by you will be shown. You can make adjustments here, if you have made any mistakes. The ‘automatic labels’ tab shows the current state of the annotation. You can use this tab to inspect the current quality of the annotation for yourself. While using the tool you will be asked to follow a so-called think aloud protocol. The idea is that you say any thoughts that come up during the annotation process aloud. This way I can get into your thought process while you interact with the tool and get a better idea of what is going on.

After trying out the tool I will ask you some questions on how you have experienced using the tool. It is important to keep in mind that this prototype really still is a first prototype. It will not work perfectly. The idea is to test the concept of the tool and verify that the basics work, to avoid issues in the final prototype. Furthermore, the focus of this research is really on creating the annotation and not necessarily on the specifics of the user interface.

Do you have any questions before we begin?

C.5 Interview questions

Note, the Dutch translation of the questions is used during the user study.

1. What area of work do you operate in?
2. How much experience with the annotation of behaviour do you have?
3. Overall, how did you feel about using the tool? What did you like/dislike?
4. How did you feel about determining which behaviours were occurring in the various video clips?
5. How did you feel about assigning labels to the video clips once you had decided on behaviours?
6. How did you feel about the length of the video clips?
7. How do you feel about the quality of the annotation? Are the behaviours labelled as you want them to? Why/why not?
8. Do you feel like the tool gives you sufficient information to determine the quality of the annotation you are creating? Do you feel like the information you were shown was correct?
9. How do you feel about the time it took you to create the annotation?
10. How many samples would you be willing to label to create a good annotation?
11. If/when you have to annotate a dataset in the future, would you consider using this tool? Why/why not?
12. Is there an important aspect of annotating in general that the tool currently overlooks?
13. Is there anything you would like to add?

C.6 Transcripts

The answers to the first two questions of the interview are not in the transcripts, to preserve the privacy of the participants. Transcripts have been translated from Dutch to English. Note: comments by researcher are in italics.

C.6.1 User 1

Think-aloud protocol

It is not possible to add labels.

[Showing] 10 images [on a 3 column grid] is ugly.

User first tried to click on the video to label it instead of the selection box.

User marks clips where the subject bends as 'Other'.

The 'Apply' button should be clearer. [After struggling to find this button.]

Make the difference between labelled and unlabelled clips more explicit. Give them a colour.

I would put the 'Apply' button to the right, it is easier to work from left to right.

I would like it if the panels scroll with me.

I want to label as many clips as possible with the smallest number of mouse clicks as possible.

User does not seem to notice the progress information that is given in the application.

It differs between studies how important which behaviour I choose is.

In the Automatic Label section:

For the switching between behaviours I would like to know which behaviour I will go to. Maybe use a drop down.

How can I correct behaviours?

[Clip of subject standing still] I understand why it doesn't do this correctly, but this is not sitting.

I want an interaction to pick out [the clips] that are not correct.

Interview

1. Overall, how did you feel about using the tool? What did you like/dislike?

   I know the history of the tool. I am excited to see it in the next stage. I think this stage is going to help, but the tool does need some improvements. However, it is a good step towards a usable tool. The Annotation Progress panel is really cool.

2. How did you feel about determining which behaviours were occurring in the various video clips?

   If you land in the middle of an annotation progress it takes some time before you know how to exactly annotate the behaviour. If you conduct the study yourself, you know exactly what behaviour you are looking for.

I could imagine that for each set of labels you make a set of example videos available. Also show example videos of behaviour you do not want to annotate as such. With 'Stand' there are many uncertain cases. When is a behaviour 'Walk'? How many steps before it is 'Walk' instead of 'Stand'?

3. How did you feel about assigning labels to the video clips once you had decided on behaviours?

   Difficult that you have to scroll so much. I want to have as many videos as possible on the screen, without scrolling.

4. How did you feel about the length of the video clips?

   That was fine.

5. How do you feel about the quality of the annotation? Are the behaviours labelled as you want them to? Why/why not?

   I would like to see a percentage per clip. What is the expectation for each video clip? What is the expected chance? Rank videos based on the accuracy percentages.

6. Do you feel like the tool gives you sufficient information to determine the quality of the annotation you are creating? Do you feel like the information you were shown was correct?

   Some I find pretty impressive. However, less than 80% correct is not yet good enough. I wonder what went wrong with 'Stand', why is it that low? Give me some videos of 'Stand' that I can label such that it hopefully does get above 80% percent when training on that specific behaviour.

7. How do you feel about the time it took you to create the annotation?

   Not long. I think usability improvements would be nice, such that it goes even more efficient.

8. How many samples would you be willing to label to create a good annotation?

   [Did not get to this question.]

9. If/when you have to annotate a dataset in the future, would you consider using this tool? Why/why not?

   Yes, definitely. Annotation goes more easily and more accurately. More accurate, because you get feedback on the accuracy. [Why would it be more accurate?] Because you can make an extension where you can look at what you're searching for and what you're not searching for. It will be easier to annotate with other people, due to the data being cut into short clips. You can show definitions of those clips. The possibility of sharing the videos plus the setting of the tool will make the quality of the annotation higher.

10. Is there an important aspect of annotating in general that the tool currently overlooks?

    The creation of an annotation scheme. Starting the annotation from an ethogram. If there are too many behaviours, it will be difficult to manage them in this environment.

11. Is there anything you would like to add?

    Asks how the quality estimate is calculated. [Do you think an explanation of this should be part of the tool?] I think you should add a question mark with an explanation of how the quality estimate is calculated. If you explain it, someone can read it if they want to and hopefully understand it.

C.6.2 User 2

Think-aloud protocol

User does not notice the 'Apply' button at first.

User labels the samples one by one instead of in groups, user works the samples off in order.

User has to scroll a lot.

User is very precise, as a consequence labels many behaviours as 'Other'.

New information on model quality is missed.

Some mistakes are made where the subject is seen as 'Stand' while it should be 'Sit'.

Apply button is sometimes not clicked, resulting in incorrect labels.

Lie is not recognized as lie.

I find it very hard to see what is happening, videos are small.

I cannot see whether he is sitting or walking.

In the Automatic Label section:

There are no lie videos, because I did not label any as such.

This is not really 'Stand' the subject takes a step.

This is more sitting.

The quality is nice.

Interview

1. Overall, how did you feel about using the tool?

   I thought it was difficult to do. I thought let's just do that, but I had to think a lot. I do think it is good to partly automatize the process. It is a very time-consuming process to do manually. Some people do not have the patience for it. The annotations do need to be really good, that is what people expect. This is something I see with Ethovision (another product at Noldus IT), that does automatic recognition, people immediately point out things that go wrong.

2. How did you feel about determining which behaviours were occurring in the various video clips?

   I found it very difficult to determine which behaviour occurs. The video clips were very small. I would not consider taking a step walking, but other people would. Longer videos would have been easier. When the video jumps back to the beginning I find it hard to determine what happens. It is difficult to determine whether someone is walking or standing still. This also happens with normal coding.

3. How did you feel about assigning labels to the video clips once you had decided on behaviours?

   See previous.

4. How did you feel about the length of the video clips?

   See previous.

5. How do you feel about the quality of the annotation? Are the behaviours labelled as you want them to? Why/why not?

   I thought I had to label quite a lot, but that needs to be done, I have to give enough input. But it is a time investment. You want to do it as precisely as possible.

6. Do you feel like the tool gives you sufficient information to determine the quality of the annotation you are creating? Do you feel like the information you were shown was correct?

   [Points to the annotation progress panel] That is really useful to have, but I would also want to verify for myself. Is this correct? I also do not know where these numbers come from. I would trust them. With 50% accuracy I would think, that is not good enough. Based on these numbers I think the annotation is good. When I look at the videos, I think this needs to be better. These are easy behaviours. More difficult behaviours also exist and then it becomes even more difficult.

7. How do you feel about the time it took you to create the annotation?

   I thought I had to label quite a lot, but that needs to be done, I have to give enough input. But it is a time investment. You want to do it as precisely as possible.

8. How many samples would you be willing to label to create a good annotation?

   I have a lot of patience. If I had to label 50 videos to get 200 videos correctly then I would be willing to do that.

9. If/when you have to annotate a dataset in the future, would you consider using this tool? Why/why not?

   Yes, definitely. It will really help me. If the videos are this small, it is really difficult for me to see. I would want to have them full screen. I would also have my own videos in high quality and really zoom in. Then I can really see what I am interested in. I would want to see the videos one by one. Label one and then the other.

10. Is there an important aspect of annotating in general that the tool currently overlooks?

    I would want to do more than stand, lie, walking. I also want to look at facial expression by going through the videos a second time. I would want to go through the entire dataset again. Let's first do the legs, then the hands.

11. Is there anything you would like to add?

    I think the interface is very clear. Very good application.

C.6.3 User 3

Think-aloud protocol

It is not immediately clear to user that they have to click on Apply.

Having to click on the checkboxes seems to be obstructive.

Goes through the videos one by one, instead of batch selection.

User forgets about the 'Apply' button.

I have some trouble with choosing the right label.

I think it is 'Other', but it is on the edge.

User moves close to the screen to be able to see what is happening.

Now I'm used to it, it is quite easy to annotate like this.

User does not notice that the model is updating or the annotation progress on the top.

User does immediately see that the Annotation Progress panel is updated

'Here I see all kinds of things about the quality'.

[*User assumes they are shown videos the computer finds difficult.*] I can imagine the AI finds these [samples] difficult.

User forgets to click 'Apply', but does correct themselves.

I have not yet seen 'Lie'.

User tries to select text instead of checkbox.

I see the quality estimates increasing, 'Sit' is quite good. Only 'Lie' needs more data.

It would be nice if the right side of the screen moves along [while scrolling].

Still 0% 'Lie', which is unfortunate, since it does have 1 'Lie' sample.

It is already very good, but I want to see more 'Lie' samples in the annotation.

In the Automatic Label section:

[*Looking at 'Lie' automated labels*] For 'Lie' it all looks good. Can I make changes? Most are good, there are a few that a not correct. It should be 70% I think.

[*Looking at 'Sit'*] There are many things here that are not correct.

[*'Sit'*] That is correct. I don't believe 'Sit' is correct, that cannot be 83% accuracy.

User looks at bar with overall accuracy, 17% is a high stand deviation. 50 percent is a lot worse.

[*'Stand'*] I see a mistake over here, but other than that it is all correct.

[*'Walk'*] Also good, definitely most are clearly walking.

I am impressed, especially that lie worked that good while I have only labelled a single sample.

Interface is nice, it all works well.

Interview

1. Overall, how did you feel about using the tool? What did you like/dislike?

   See next.

2. How did you feel about determining which behaviours were occurring in the various video clips?

   It was very doable. The interface worked alright. Generally it was difficult if a clip was on the edge of two behaviours. What is the definition of standing? This is something that is always difficult with annotating. [Do you also experience these issues with manual annotation?] Also with The Observer XT, the first frames are very difficult. With The Observer you are really trying to make the annotation accurate to the exact frame. Here you can simply decide per clip. For The Observer you have to specify exactly from when to when a behaviour occurs. Takes a lot of time to get that right.

3. How did you feel about assigning labels to the video clips once you had decided on behaviours?

   I had to get used to that you can select multiple things. I thought I had to annotate per video. It was not that difficult. [Would you rather select multiple things or label clip by clip?] I think I would prefer labelling by clip.

4. How did you feel about the length of the video clips?

   I thought that was alright. If they were longer, the behaviours would become more ambiguous, there will be more behaviours in a clip. Ideal length.

5. How do you feel about the quality of the annotation? Are the behaviours labelled as you want them to? Why/why not?

   I thought it was good. Especially ‘Lie’, since I labelled only 1 video. I was surprised that ‘Sit’ was not the best one, because I did label a lot of those.

6. Do you feel like the tool gives you sufficient information to determine the quality of the annotation you are creating? Do you feel like the information you were shown was correct?

   Yes the percentages are useful. I would not consider that enough information. I especially appreciate that I can see examples. I do think it is strange that Lie is still at 0%. Because I see a lot of examples where it goes correctly, it should intuitively be higher. Percentages should agree more. The total accuracy could be correct.

7. How do you feel about the time it took you to create the annotation?

   I was content. I did not feel it took me too much time.

8. How many samples would you be willing to label to create a good annotation?

   Depends on how important it is. I think if I could get this even better, I would have no problem doing 100 more. If it is very important you can go even further than that. These 50 were quite fast, especially because you can easily click them.

9. If/when you have to annotate a dataset in the future, would you consider using this tool? Why/why not?

   Yes, I definitely think so. I like the way the interface works, with selecting per clip. What would be ideal is if you can correct automatic labels. To see where it goes wrong and interfere. The data is different than when you annotate the full video, you have labelled clips instead of frames.

10. Is there an important aspect of annotating in general that the tool currently overlooks?

    What I miss is the correction of automatic labels.

11. Is there anything you would like to add?

    I would like it if the right side of the page scrolls with you.

D PROTOTYPE II

The legend, as used for the figures in this section, can be found in Figure 5.4 in Section 5.3.

D.1 Discarding samples

Preliminary results for the different choices of what to do with the labelled samples from the previous cascade level ('discard', 'keep uncertain' or 'keep all') are shown in Figures D.1 and D.2. For all these results the MBC's classification accuracy is used for convergence estimation. The MC dropout variation of the maximum softmax output metric, with $R = 10$ and a dropout-rate of 0.2 is used. The target precision is 0.9.

Note, these results are based on the first 5000 samples of each dataset, due to time constraints. Furthermore, the results are based on old code. This means that a bug where some data segments contained data from events further or earlier along the time series was still present. However, this should have a minimal effect on the difference in performance between the methods.

As to be expected, the option of keeping all labelled samples throughout the process seems to result in the highest precision. This makes sense since this option has the most data available for the MBC and CCP.

The keep uncertain option does not necessarily seem to improve the results compared to the discard option, even while also having more data available. It is likely the case that only keeping the uncertain samples skews the labelled data distribution too much. As a result, the model may focus too much on out-of-distribution samples. A comparable mismatch in representation could be expected for the 'keep all' option, however since this option keeps all samples, also the ones the system is certain about, the data may remain more representative.

The discard option starts fresh at every level of the cascade, creating a new dataset representing the leftover data. This means it is the only option that does not inherently carry the possibly problematic bias. This could be the reason why it does not perform much worse than the keep all option.

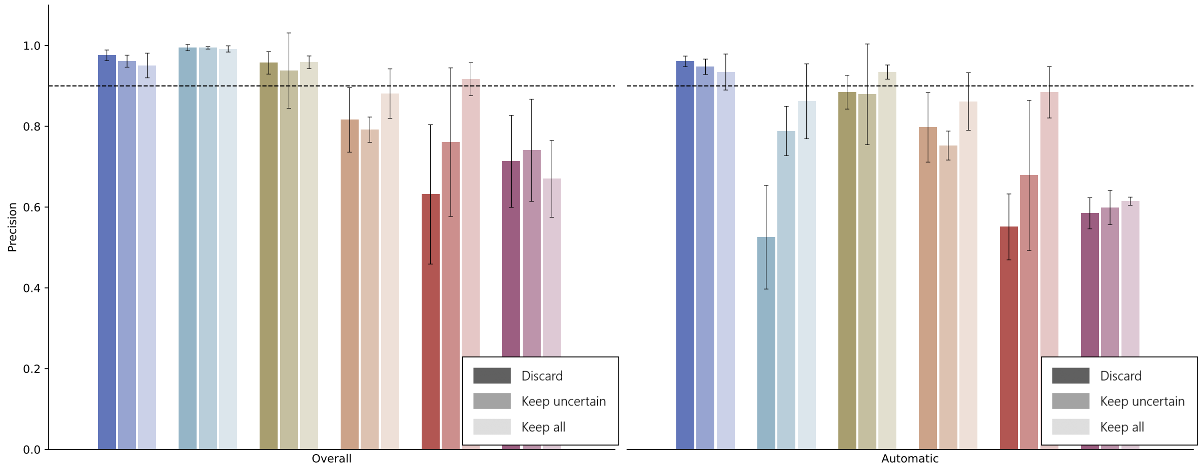


Figure D.1: Precision of the annotation for the different options of handling labelled samples from the previous cascade level (discard, keep uncertain or keep all). Both the overall (both the automatic and manual labels are considered) precision and the automatic precision are shown. The MBC's accuracy is used as the objective for convergence. The precision is shown for all samples and only the automatically labelled samples. Note, that the precision is weighted per class.

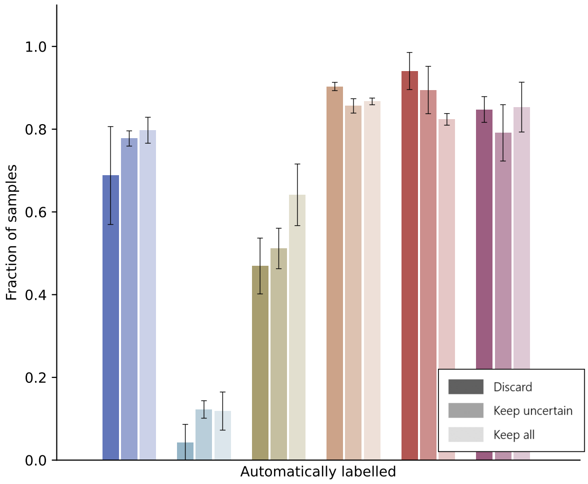


Figure D.2: Fraction of automatically labelled samples for the different options of handling labelled samples from the previous cascade level (discard, keep uncertain or keep all). The MBC's accuracy is used as objective for convergence. Note, the number of samples is **not** weighted per class

D.2 Results

D.2.1 CCP settings

In Section 5.1.1 different settings for determining the threshold used in the CCP are discussed. The difference in precision when setting the threshold using the ‘extreme’ setting and the ‘mean’ setting is shown in Figure D.3. Results are shown for a precision target of 0.9, the basic maximum softmax output metric and a single overall threshold for all classes. As to be expected, the extreme setting is more conservative than the mean setting. It can also be seen that even the extreme setting has trouble meeting the target precision.

In Figure D.4 the difference when setting a single overall threshold or setting a threshold per class can be found. Results are again shown for a precision target of 0.9 and the maximum softmax output metric. It can be seen that the ‘per class’ setting negatively impacts the ability to meet the target precision. This is likely due to the ‘per class’ setting having to set the separate thresholds with even less data, resulting in a less stable trade-off.

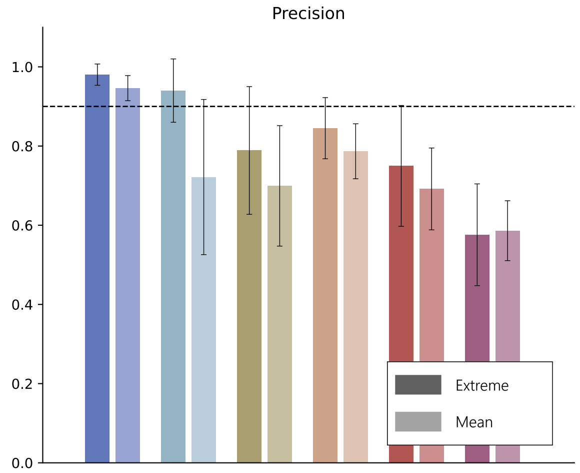


Figure D.3: Precision when setting the CCP threshold using the extreme or mean setting. The precision target is 0.9 and the metric used is the maximum softmax output. A single overall threshold is set.

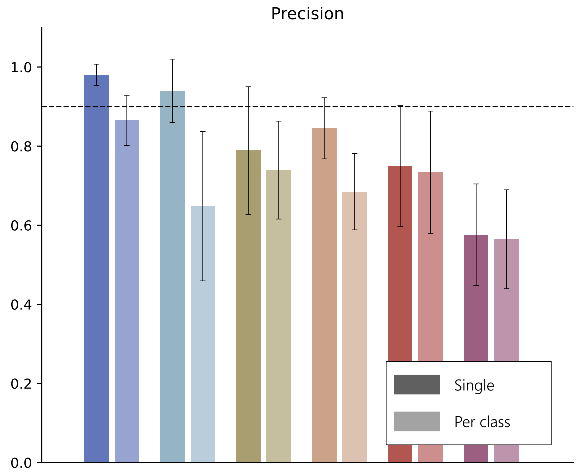


Figure D.4: Precision when setting a single overall threshold or a threshold per class. The precision target is 0.9 and the metric used is the maximum softmax output. The extreme setting is used.

D.2.2 Basic metrics

In this section, the results for the three basic metrics (maximum softmax output, maximum logit output and entropy) are presented for a precision target of 0.9. The extreme threshold setting is used for setting a single overall threshold. In Figure D.5 the ability to meet the precision target is shown. In Figure D.6 the accompanying recall is shown. For the different metrics, there does not seem to be much difference in the ability to meet the precision target. The recall is also comparable. The results for precision targets of 0.8 and 0.95 can be found in Appendix D.2.2.

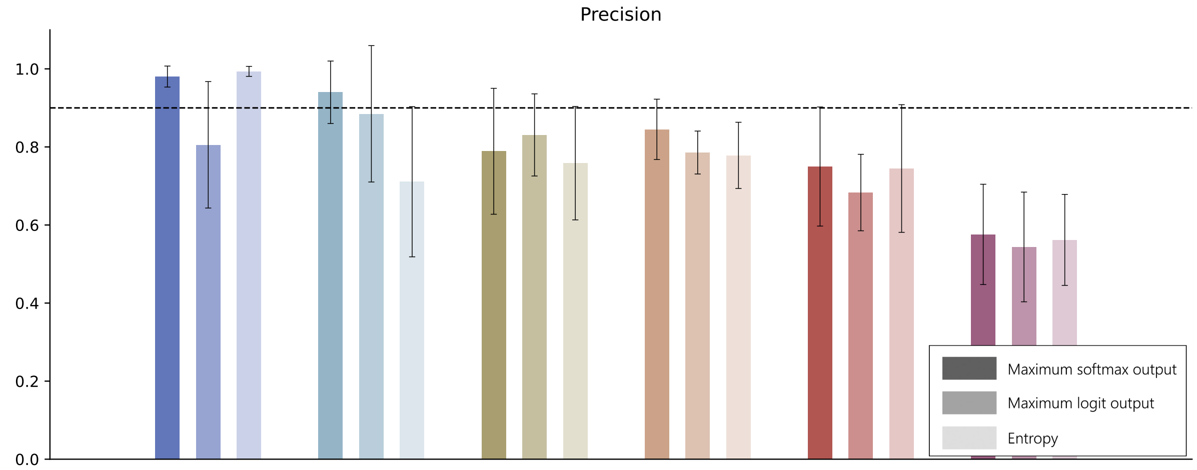


Figure D.5: Comparison of the precision obtained for the three basic metrics for a target precision of 0.9.

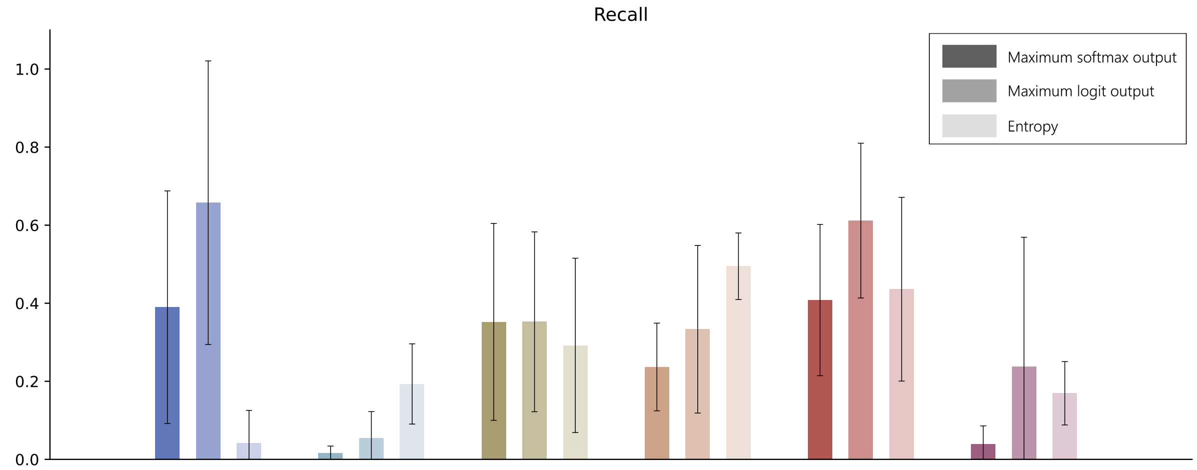


Figure D.6: Comparison of the recall obtained for the three basic metrics for a target precision of 0.9.

Target precision

The results for the basic metrics for a precision target of 0.8 can be found in Figure D.7 and D.8. These figures show how the CCP behaves when decreasing the desired quality of the system. As can be seen, for this lower precision target (lower than 0.9), more datasets manage to meet the precision target. The recall also seems to be higher.

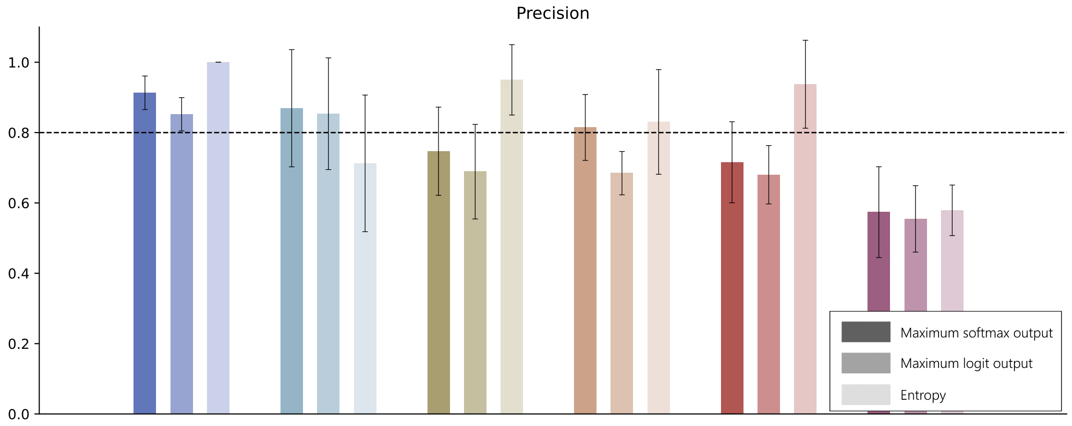


Figure D.7: Comparison of the precision obtained for the three basic metrics for a target precision of 0.8.

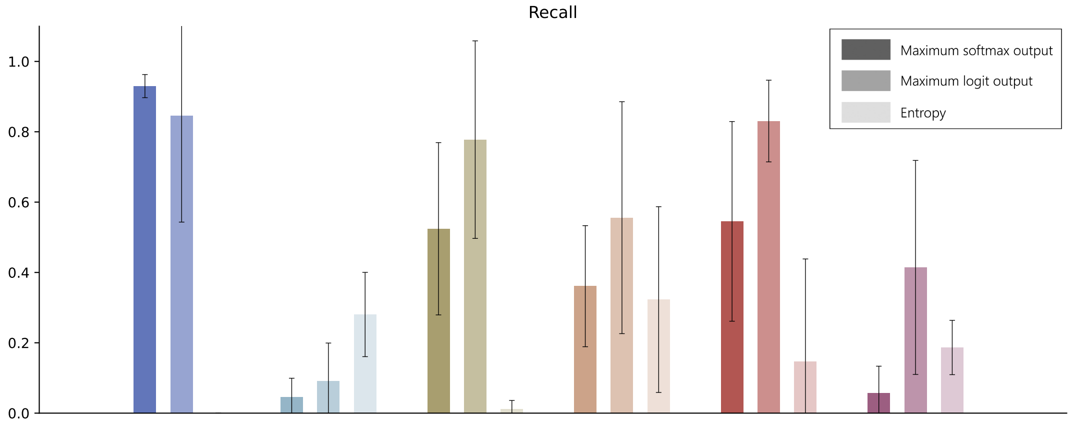


Figure D.8: Comparison of the recall obtained for the three basic metrics for a target precision of 0.8.

The results for the basic metrics for a precision target of 0.95 can be found in Figure D.9 and D.10. These figures show how the CCP behaves when increasing the desired quality of the system. The precision does seem to have increased slightly for the datasets, compared to the results for the precision target of 0.9.

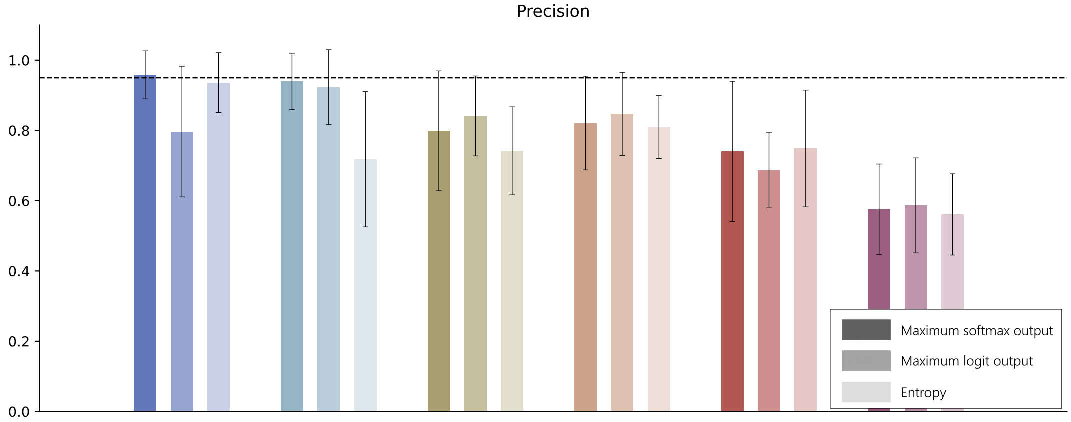


Figure D.9: Comparison of the precision obtained for the three basic metrics for a target precision of 0.95.

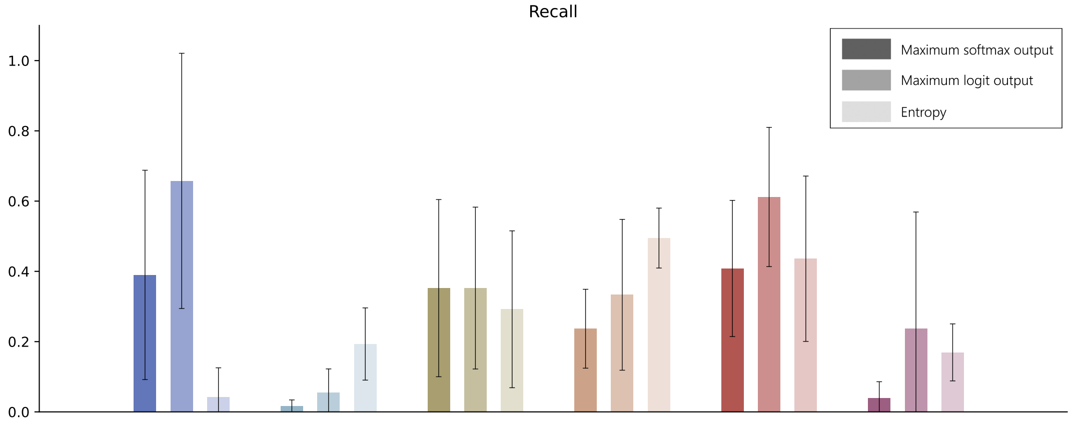


Figure D.10: Comparison of the recall obtained for the three basic metrics for a target precision of 0.95.

D.2.3 MC dropout

For different dropout-rates, different MBC were trained, the resulting classification accuracies can be found in Table D.1. For the MC dropout metrics two hyperparameters need to be set, the dropout-rate and the number of model runs R. The effect of these hyperparameters is first explored in this section. At the end of this section, the difference between using different metrics is discussed.

Table D.1: Mean test classification accuracy over 5 model runs, including the standard deviation. Results are given for models trained with 200 samples and dropout-rates: 0.1, 0.2 and 0.4. Test accuracies are based on the mean output for $R = 50$.

Dropout-rate	0.1	0.2	0.4
BAfitness	82.76 ± 6.61%	85.48 ± 9.20%	71.52 ± 8.58%
HCIgestures	55.19 ± 6.81%	50.67 ± 3.31%	48.05 ± 4.83%
OPPORTUNITY AR	77.75 ± 5.89%	76.62 ± 5.35%	74.04 ± 4.46%
Skoda	63.34 ± 2.45%	65.57 ± 3.11%	62.39 ± 4.76%
Daphnet FoG	67.75 ± 5.92%	61.47 ± 9.55%	70.85 ± 3.88%
Cow behaviour	58.52 ± 5.94%	56.88 ± 3.46%	57.52 ± 1.52%

Dropout-rate

The results for the maximum softmax output metric with different dropout-rates can be found in Figures D.11 and D.12. These results are for the maximum softmax output.

For most datasets, the dropout-rate of 0.2 generally seems to result in higher precisions. However, it is difficult to draw a conclusion due to sizeable standard deviations and differences between the datasets. Since there are big differences between the datasets, it might be a good idea to make the dropout-rate tunable in future research.

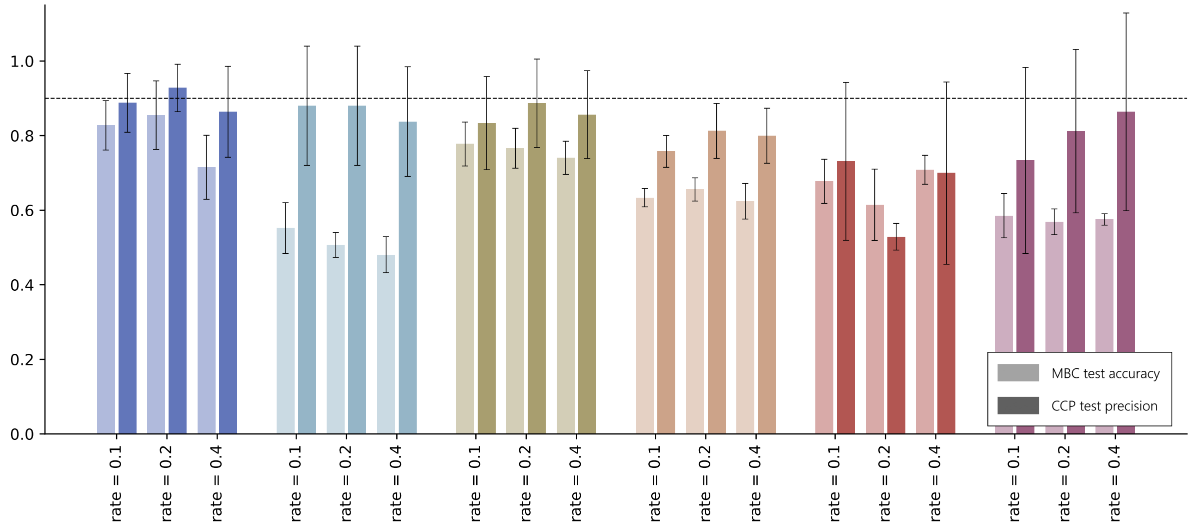


Figure D.11: Precision of the MC dropout maximum softmax output metric for different dropout rates. Results are shown for a precision target of 0.9 and $R = 50$.

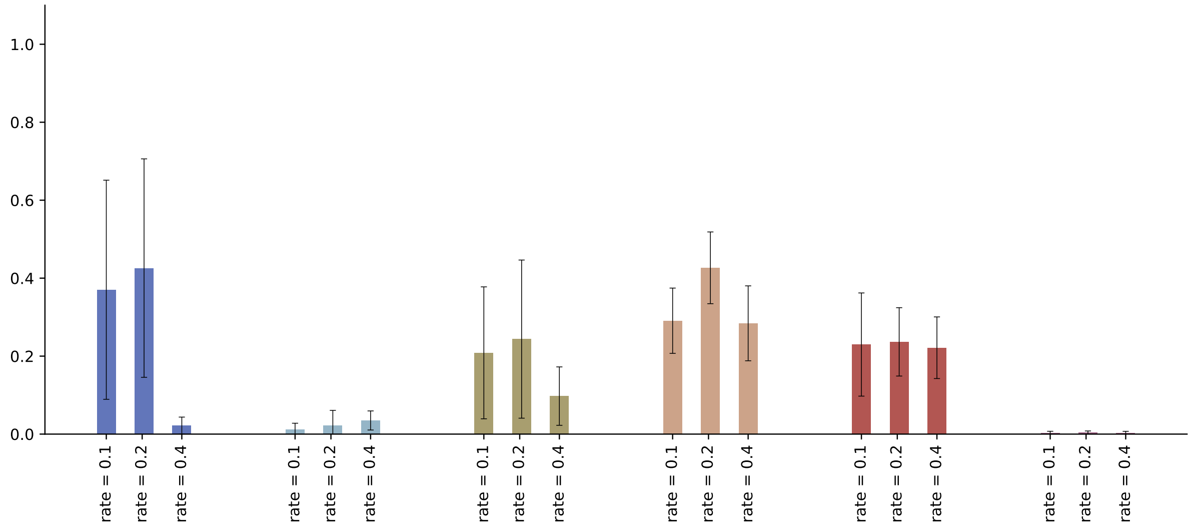


Figure D.12: Recall of the MC dropout maximum softmax output metric for different dropout rates. Results are shown for a precision target of 0.9 and $R = 50$.

Model runs

The results for different amounts of model runs R can be found in Figures D.13 and D.14, these results are also for the maximum softmax output metric. As can be seen, the difference in both the test accuracy and precision is slight between $R = 1$, $R = 10$ and $R = 50$. $R = 50$ and $R = 10$ do seem to be slightly beneficial for the precision compared to $R = 1$. Between $R = 50$ and $R = 10$ no consistent differences are observed.

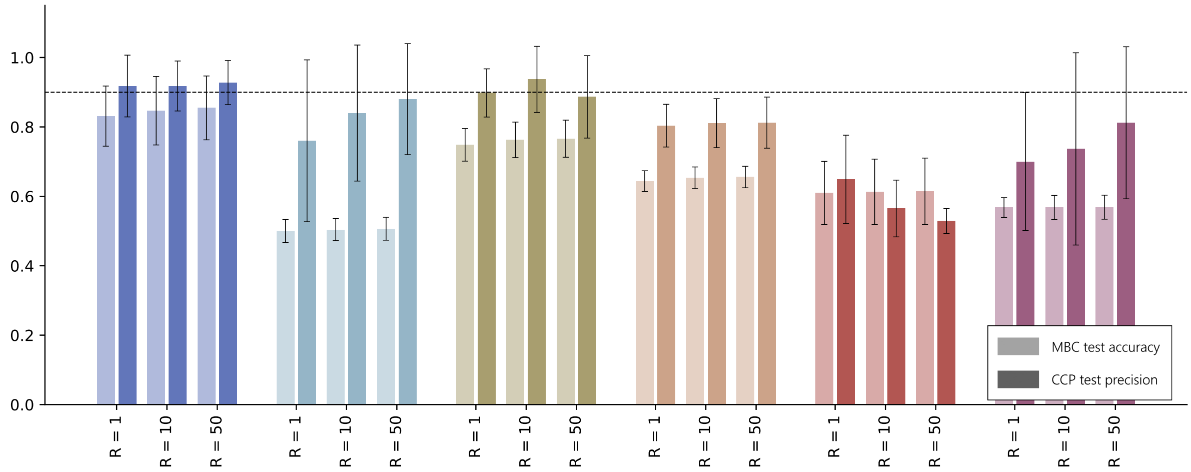


Figure D.13: Precision of the MC dropout maximum softmax output metric for different values for the number of model passes R. Results are shown for a precision target of 0.9 and a dropout-rate of 0.2.

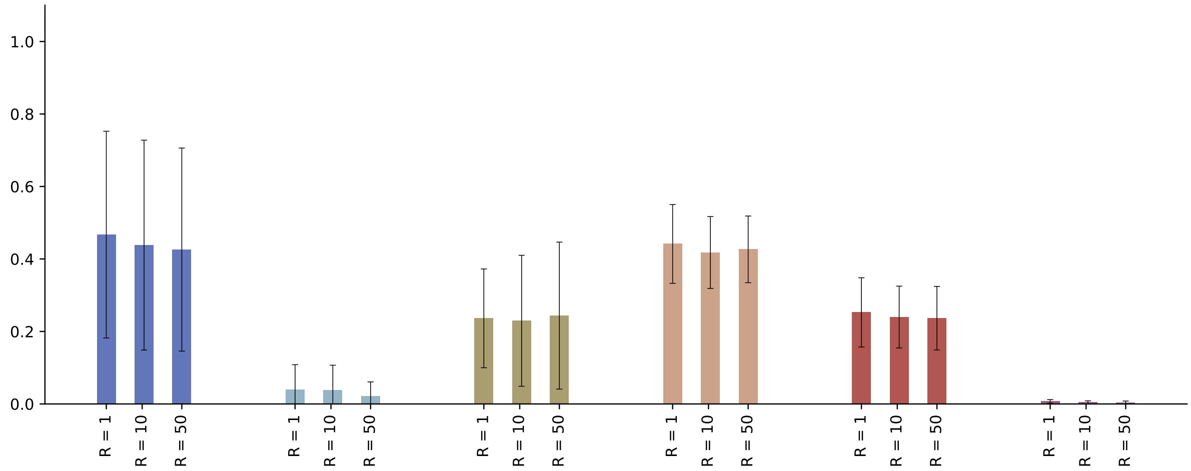
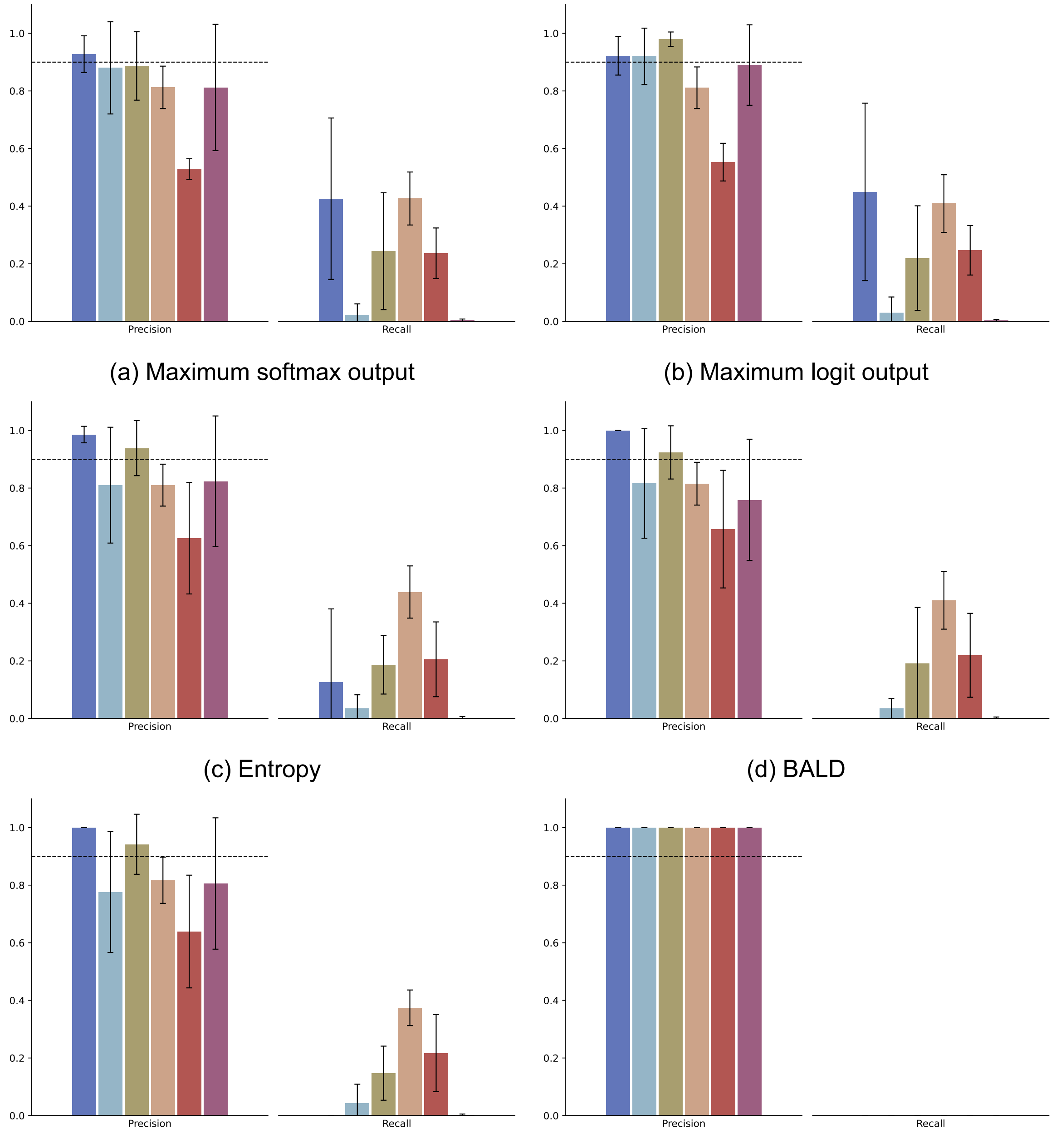


Figure D.14: Recall of the MC dropout maximum softmax output metric for different values for the number of model passes R. Results are shown for a precision target of 0.9 and a dropout-rate of 0.2.

Metrics

The mean classification accuracy of the MBCs used in this section can be found in Table 5.2. The results for each of the MC dropout metrics, as discussed in Section 5.1.2, can be found in Figure D.15 for a precision target of 0.9. The results are given for a dropout-rate of 0.2 and $R = 50$. There is little difference between the different metrics. Only the variance-ratio gives very different results.

For this metric, the threshold is always set to the maximum. As a result, the target precision is reached, but the recall is zero. This has likely to do with the discrete nature of the variation-ratio. The threshold can only separate discrete cases, if one case is too strict and the other not strict enough, an intermediate threshold cannot be set.



(a) Maximum softmax output

(b) Maximum logit output

(c) Entropy

(d) BALD

(e) Variance

(f) Variation-ratio

Figure D.15: MC dropout results on the test data for a precision target of 0.9. Results are shown for dropout-rate = 0.2 and $R = 50$.

D.2.4 Convergence objective

For the objective for the convergence estimation of the number of labelled samples the MBC's accuracy or the mean between the MBC's accuracy and the CCP's recall can be used as objective. In Figure D.16, the results for both options are shown. The results are based on the basic maximum softmax output metric for a target precision of 0.9.

As can be seen, the precision target is more closely met when using the classification accuracy as the objective. The number of automatically labelled samples is slightly higher when using the mean as objective. In Figure D.17 the average number of samples manually labelled per cascade level is shown. As can be seen, the mean objective stops the number of labelled samples at a very early stage, causing the system to underperform. It could be the case that the system does not accurately estimate the recall or that the recall does not continually increase.

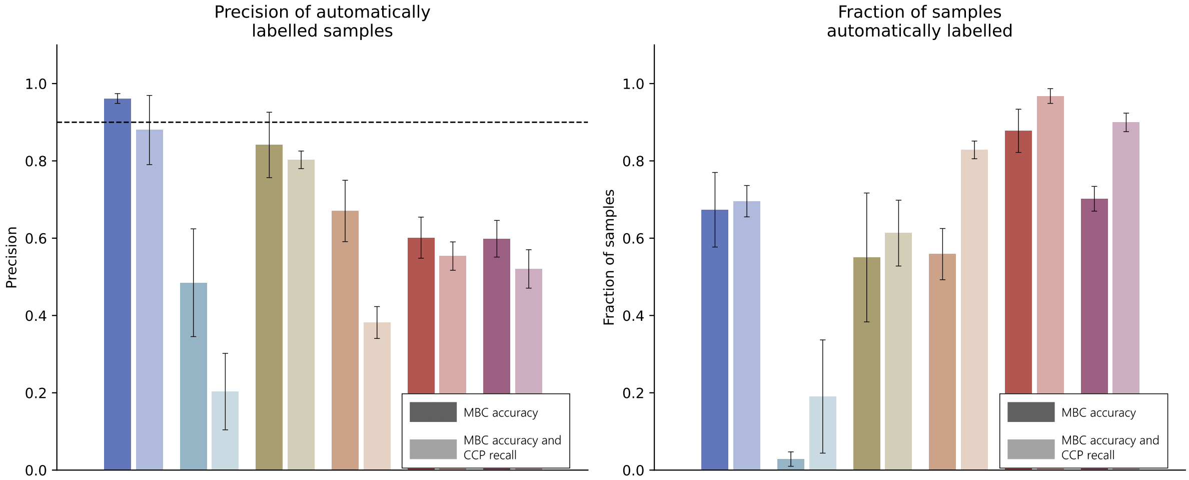


Figure D.16: Precision of the automatically labelled samples and the fraction of samples automatically labelled for the different objectives for the convergence with respect to the number of labelled samples. Note, the precision is weighted per class but the fraction of automatically labelled samples is not.

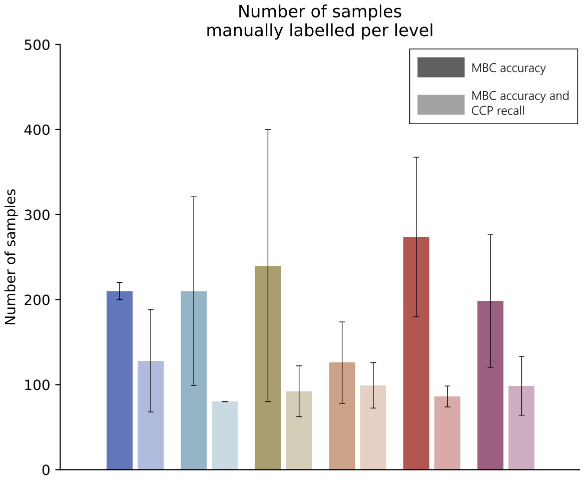


Figure D.17: Number of automatically labelled samples per cascade level for both experiments with the different objectives (for the convergence with respect to the number of labelled samples).

E USER STUDY

E.1 Information brochure

Information brochure – The Smart Annotation Tool

User study on the OPPORTUNITY++ dataset

Brochure version: 05-11-2022

You have been asked to participate in a user study where you would be observed while you test the prototype of a software tool. You would be interviewed afterwards. If you decide to participate, the user study will be conducted by Annemarie Jutte as part of a master thesis project at the University of Twente. The master thesis is conducted as an external project at Noldus Information Technology. You have been selected to participate due to your affinity with behavioural annotation. You are free of choice on whether you want to take part in this user study or not. If you do not take part, you will not face any negative consequences. If anything in this information brochure or the consent form is unclear, please ask any questions you may have.

The Smart Annotation Tool

For my master thesis I am developing the so-called Smart Annotation Tool. This tool is developed to aid behavioural researchers with the annotation of behavior of people and animals in videos and accelerometer data. This is currently a very time-costly task, the Smart Annotation Tool should reduce this effort by automating parts of the annotation. This is done through a collaboration of user and machine learning. In this user study I am interested in the workings of an early prototype of this tool and specifically how possible users feel about using the tool. This information can be used to further advance the development of the tool.

Testing of the tool

This study will involve you using the prototype of the Smart Annotation Tool under the researcher's instructions. You will be observed by the researcher while using this tool. You will use the tool to annotate (part of) the OPPORTUNITY++ dataset. Which is a dataset of people performing various locomotion tasks, like: walking, sitting, standing and lying down. While using the tool, you will be shown short video clips. You will be asked what kind of behaviour the subject in the video is displaying. Using the labels you specify, a machine learning model will be trained. This model will be used to annotate the parts of the video clips you do not have to label. You will be shown videos to which a label has been automatically assigned.

While you are using the tool, you will be asked to formulate your thoughts out loud, this is referred to as a think aloud protocol. This will be helpful for the researcher obtain insight into how users approach the tool.

Interview

After using the tool you will be asked to take part in a small interview. This interview will concern questions on your experience with the tool, like your thoughts and feedback. The full study is expected to take less than an hour: 30 minutes for explaining and using the tool and less than 30 minutes for the interview. There will be no scheduled break, but you may take a break at any time you desire.

Further data collection

During the user study data will be collected. While you use the tool, the researcher will write down observations. Statements you make while using the tool might also be written down and stored. The same goes for answers you give during the interview. The labels you give to video clips while using the tool will be stored, as will the model trained using these labels and the resulting annotation. While you are using the tool, parts of the process will be automatically timed. This helps the researcher obtain insight into time management within the tool.

Furthermore, you will be asked for your area of work and experience with behavioural annotation, this data will be stored with the other data. You will be asked whether audio recording may be used during the session. This would be useful for processing the results. After processing the results, the audio recordings will be deleted. You are free to deny this. Your personal data will not be shared with any third parties.

The use of your data
From the data collected, the following may appear in the researcher's master thesis and related reports on the Smart Annotation Tool: quotes from the interview, statements made by you, observations made by the researcher, labels you gave video clips, statistics on the annotation created and the durations of parts of the annotation. This data will only appear fully anonymized.

All data collected during the user study will be removed before the end of my master thesis, besides the data included in reports (see the previous paragraph) and used to further the design of the Smart Annotation Tool. The removal will take place at least before 31-12-2022, but earlier if the thesis is already finished.

You may withdraw from this study at any time, with no further questions asked. After the study your data can be used for the further development of the Smart Annotation Tool. The data will be used in the master thesis report relating to this research. Furthermore, any data published in the master thesis report may be used in any reports relating to the development of the Smart Annotation Tool. Only anonymized data will be published. After the data has been published it may not be removed from reports or products upon withdrawal. However, any other data stored will be removed when requested.

If you have any questions at any point do not hesitate to contact the researcher of this study right now or later, using the contact information below.

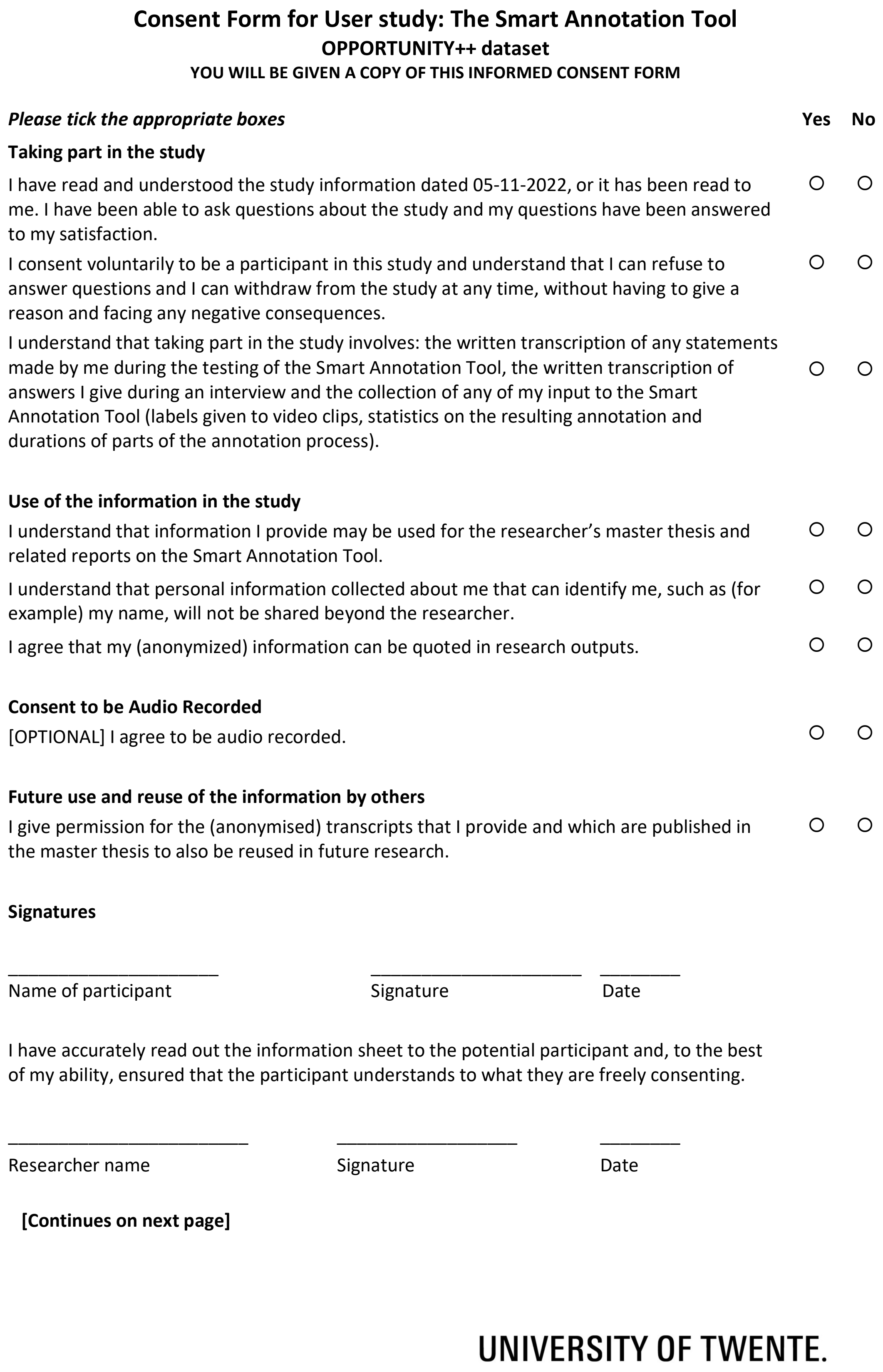
Thank you for your interest in this study.

Annemarie Jutte,
[redacted]

If you have questions about your rights as a research participant, or wish to obtain information, ask questions, or discuss any concerns about this study with someone other than the researcher(s), please contact the Secretary of the Ethics Committee Information & Computer Science: ethicscommittee-CIS@utwente.nl.

This research is supervised by: drs. E. van Dam ([redacted]), dr. ing. G. Englebienne ([redacted]) and dr. Nicola Strisciuglio ([redacted]).

E.2 Informed consent form



Consent Form for User study: The Smart Annotation Tool

OPPORTUNITY++ dataset

YOU WILL BE GIVEN A COPY OF THIS INFORMED CONSENT FORM

Please tick the appropriate boxes	**Yes**	**No**
Taking part in the study		
I have read and understood the study information dated 05-11-2022, or it has been read to me. I have been able to ask questions about the study and my questions have been answered to my satisfaction.	○	○
I consent voluntarily to be a participant in this study and understand that I can refuse to answer questions and I can withdraw from the study at any time, without having to give a reason and facing any negative consequences.	○	○
I understand that taking part in the study involves: the written transcription of any statements made by me during the testing of the Smart Annotation Tool, the written transcription of answers I give during an interview and the collection of any of my input to the Smart Annotation Tool (labels given to video clips, statistics on the resulting annotation and durations of parts of the annotation process).	○	○
Use of the information in the study		
I understand that information I provide may be used for the researcher's master thesis and related reports on the Smart Annotation Tool.	○	○
I understand that personal information collected about me that can identify me, such as (for example) my name, will not be shared beyond the researcher.	○	○
I agree that my (anonymized) information can be quoted in research outputs.	○	○
Consent to be Audio Recorded		
[OPTIONAL] I agree to be audio recorded.	○	○
Future use and reuse of the information by others		
I give permission for the (anonymised) transcripts that I provide and which are published in the master thesis to also be reused in future research.	○	○

Signatures

______________________ ______________________ ________

Name of participant Signature Date

I have accurately read out the information sheet to the potential participant and, to the best of my ability, ensured that the participant understands to what they are freely consenting.

_________________________ ___________________ ________

Researcher name Signature Date

[Continues on next page]

UNIVERSITY OF TWENTE.

Study contact details for further information:

The researcher can be contacted at: [redacted]

The research supervisors can be contacted at: drs. E. van Dam ([redacted]), dr. ing. G. Englebienne ([redacted]) and dr. Nicola Strisciuglio ([redacted]).

Contact Information for Questions about Your Rights as a Research Participant

If you have questions about your rights as a research participant, or wish to obtain information, ask questions, or discuss any concerns about this study with someone other than the researcher(s), please contact the Secretary of the Ethics Committee Information & Computer Science: ethicscommittee-CIS@utwente.nl.

E.3 Instructions

Note, the Dutch translation of the instructions is used during the user study.

We are here today to test the prototype of the Smart Annotation Tool. The idea of the Smart Annotation Tool is to speed up the traditional method of behavioural annotation, where you need to go from front to back through all the data and precisely annotate the behaviour. Instead of labelling all this data by hand, only a small part of the data needs to be labelled. All the other data will be annotated using artificial intelligence.

I will first give a quick explanation of how the tool works. You will be repeatedly asked to label sets of videos. The AI will use these labels to create a model, it uses this model to recognize videos that are similar to the ones you have labelled. It will need quite some videos before it starts to get good at recognizing videos. Therefore, it will take a while before the model will help you. When the tool has determined that it has learned enough, it will label a set of video clips. It will only label the video clips the tool is most confident about.

More specifically, it will aim to get at least 95% of the automatically labelled samples correctly labelled. In final versions of the Smart Annotation Tool this will be a percentage you can control. You can see the videos that have been automatically labelled the side panel. And if you go to the automatic labels tab, you can see exactly how videos have been labelled. If you go to the manual labels tab, you can also see how you labelled videos yourself and make any changes if you made a mistake.

Today we are going to label a dataset for which various locomotion tasks need to be annotated. The subject of which we want to create the annotation is the person with the blue jacket. You will be presented with short video clips of this person. You can assign one of 4 behaviours for each video clip: walking, standing, sitting and lying. Try to choose the best fitting behaviour whenever possible. If someone takes a step this should be considered walking. If multiple behaviours occur in a single clip, try to choose the behaviour that occurs closest to the middle of the clip. If this is not an option or if a different behaviour occurs that fits in no way with the four that we are annotating, please select the 'Other' behaviour. If you cannot see what behaviour occurs, due to, for example, objects blocking the view, please select the 'Unknown' behaviour. I will show you some examples of more difficult cases, how they should be labelled… [Show some examples]

Note the exact choices you make is not that important. The most important thing is that your annotations are consistent.

While using the tool you will be asked to follow a so-called think aloud protocol. The idea is that you say any thoughts that come up during the annotation process aloud. This way I can get into your thought process while you interact with the tool and get a better idea of what is going on.

After trying out the tool I will ask you some questions on how you have experienced using the tool. It is important to keep in mind that the point of this prototype is really the workings of the AI and the creating of annotation. The user interface needs to work, but it might not work optimally.

Do you have any questions before we begin? Then now you can start labelling. You will need to label 108 samples, after this a set of samples will be automatically labelled. Do not forget to click on apply after you have labelled samples.

E.4 Numerical results

In Table E.1, the performance of the Smart Annotation Tool in terms of the precision and labelling effort can be found for each of the participants in the user study. In Table E.2, statistics regarding the samples labelled by the users can be found. The users are referred to as ‘user 1’, ‘user 2’ and ‘user 3’.

In both tables, an example user is shown, which is the result of an annotation by the researcher. This results was shown to user 3, since they did not get any automatically labelled samples. An additional example user is also mentioned in the tables. This shows that the precision does not need to be 1.000, although it was for the three users.

Table E.1: Performance of the Smart Annotation Tool for the users in the user study. Note, users all labelled the same 108 samples, no convergence estimation was used.

	Automatic precision	Recall	Number of manually labelled samples	Number of automatically labelled samples
Example user	1.00	0.05	108	75
User 1	1.00	0.02	108	28
User 2	1.00	0.00	108	0
User 3	1.00	0.11	108	166
Additional example user	0.75	0.24	108	355

Table E.2: Overview of statistics regarding the manual labels given by the users in the user study. Annotation duration is an approximation of the time spend annotating. *User 1 also inspected the ‘Manual Labels’ tab during this time.

	Rater-reliability (no ‘Other’ or ‘Unknown’)	Number of samples labelled ‘Other’	Number of samples labelled ‘Unknown’	Annotation duration
Example user	0.85	0	0	-
User 1	0.82	0	1	23 min*
User 2	0.71	0	1	14 min
User 3	0.83	0	0	15 min
Additional example user	0.88	0	0	-

E.5 Interview questions

Note, the Dutch translation of the questions is used during the user study.

1. What area of work do you operate in?
2. How much experience with the annotating of behaviour do you have?
3. Overall, how did you feel about using the tool?
4. What parts of using the tool did you like?
5. What parts of using the tool did you dislike?
6. How did you feel about determining which behaviours were occurring in the various video clips?
7. How did you feel about assigning labels to the video clips once you had decided on behaviours?
8. How did you feel about the length of the video clips?
9. How do you feel about the quality of the annotation? Are the behaviours labelled as you want them to? Why/why not?
10. If not, when would you be satisfied with the quality?
11. Do you feel like the tool gives you sufficient information to determine the quality of the annotation you are creating?
12. How do you feel about the time it took you to create the annotation?
13. How much time would you be willing to wait in between batches of samples?
14. How much of the dataset would you be willing to label to create an annotation?
15. If/when you have to annotate a dataset in the future, would you consider using this tool? Why/why not? Can you foresee any issues?
16. Is there an important aspect of annotating in general that the tool currently overlooks?
17. Is there anything you would like to add?

E.6 Transcripts

Questions 1 and 2 from the interview questions are omitted, to preserve the privacy of the participants. Transcripts have been translated from Dutch to English. Note: comments by researcher are in italics.

E.6.1 User 1

Think-aloud protocol

I am immediately interested in these three headers. Automatic labels will the model do. Manual labels are my labels. I do not immediately see a flow of what will happen, what should I do first.

[*Enlarges video*]. This is standing.

[*Clicks again on the video and gets a short explanation as to how to label from researcher*] Ah now I can classify multiple labels as stand.

I find this difficult... It is very difficult.

In this video he really wants to go somewhere.

Now I have labelled 0.37% of the dataset!

I can now seem them at manual labels.

[*Opens the empty 'Lie' tab*] Oh I did not have lie yet.

Sits close to the screen to see everything clearly.

What is exactly the definition of standing?

Clicks next to the select box multiple times.

I thought maybe after I have labelled these samples I can immediately load new samples. Then I can label more sitting samples at the same time.

Process disrupted by bug, where a selection is not made if you click to quickly.

He is not moving yet, I think that is just standing.

Waits for images to reload, then notices that they have not yet clicked on apply

This is really difficult.

It is moving, he really wants to start walking.

[*Goes to manual tab*] I am just going to have a look whether I change my mind on some, whether I think it is consistent.

I actually do think so! But these are really doubtful cases.

I can change these labels.

I notice that I am already interpreting such an image. With knowledge of humans I think, he will make a step. But I have labelled the others as stand. I already take into account that I think the model will not understand this if I classify it as walk. I am really thinking about what the model expects.

I think the labelling is easier if you can also click on the images to select them and you can click two times to enlarge them. It is difficult that it needs to be precise at the moment.

Researcher forgot to inform user of the 'Other' and 'Unknown' categories, is now explained.

There is one sample I thought was 'Other'. [*Does find the sample, but does not end up labelling it such.*]

Maybe it can be improved with arrow buttons or short cuts.

Tries to click quickly, selection disappears.

Bug where white gif is shown, fixed by letting user reload the image.

I still feel like I am not completely consistent the whole time.

I do of course want to reach a high accuracy.

Clicks too quickly.

I can do this one as 'Unknown' [About an image where subject is half. outside of the screen.]

Continues labelling after the 108 samples. [*Has to wait for the loading screen.*]

Now it is training.

Does not notice that the annotated samples have been updated.

Number of clips labelled automatically…

[*Goes to manual labels tab*] It is not sorted by the order you have annotated them.

Let's look at the automatic labels.

Lie, yes that should not be difficult, I think!

It says unlabelled, but it is sit. [*Is informed that this still needs to be changed.*]

Stand has zero, and walk also has zero.

Interview

1. Overall, how did you feel about using the tool?

   Quite straightforward, quite easy. Generally, I found it quite nice to work with. User friendliness and speed can be improved upon. I would like short cuts and that I do not need to exactly select the box. Maybe load more images.

2. What parts of using the tool did you like?

   It is very nice that there is something that labels sequences for you and you can see in real time how much it has labelled. That you can see for the automatic labels whether they are correct or not. Also that you can see, in this category there is nothing. The interactive part, that you can label constantly and see what and whether something is added. You get the feeling you're not doing it all by yourself.

3. What parts of using the tool did you dislike?

   The first thing I think of, is that you really need to select those small boxes. That was the most tiring for me. That gets you out of the flow.

4. How did you feel about determining which behaviours were occurring in the various video clips?

   That was quite difficult, not for sitting and lying, but for standing and walking it was. Especially if he is behind the counter. Everyone says, you stand behind the counter, but every now and then you're shifting a bit. You shift your balance a bit.

   I think that is inherent to the data, not the tool itself I think. It is also of course related to the clip duration. If you make them longer, it only gets more difficult.

5. How did you feel about assigning labels to the video clips once you had decided on behaviours?

   See previous.

6. How did you feel about the length of the video clips?

   I would say the clip length is appropriate. For these categories that is. You only still have those transitions. It would be a lot easier if you can put them all on one pile, but I do not know if the network can do anything with that.

7. How do you feel about the quality of the annotation? Are the behaviours labelled as you want them to? Why/why not?

   Currently it is 100% of what it has labelled. That is really nice, I immediately got the idea that it is correct. But you also want it to do stand and walking. That is what I have annotated the most. [Goes to the precision target] if I lower this, I want it annotate more. If you can change the quality and see what happens, how many labels will be annotated. That you can also have a look at them. Maybe that it already indicates this with the labels. I assume it works based on a certainty and can adjust it.

   When I use DeepLap cut, I see a video it is struggling with. Then I sought for frames it was struggling with myself. For that it worked really well. I feel like you cannot trust on the confidence of the network for the delivering of samples to label.

8. If not, when would you be satisfied with the quality?

   That is difficult to say. Once you think there are mistakes every now and then. If you want 100% accuracy, you need to look at everything. But if 10% is annotated incorrectly, it should still be faster this way. If it goes towards 50% it becomes worthless.

9. Do you feel like the tool gives you sufficient information to determine the quality of the annotation you are creating?

   You can also know that by really looking at it at the moment. Maybe if you would sort it based on confidence, you will have more tools to really get an impression. At this confidence, a mistake is made. The check whether automatic labels are correct or not, is currently the same process as the labelling itself. It would be even more easy, if the confidence would be more clear. It is very difficult to give such a value to a sample. If the process would be more efficient, maybe more than 10% incorrect would be enough.

10. How do you feel about the time it took you to create the annotation?

    How long did I take? [About 25 minutes] That is still quite slow, but I think it could be twice as fast if you know for yourself how you should annotate difficult situations. With some user friendliness and smart keys, a lot of time can be won. Then not a lot needs to be changed fundamentally.

11. How much time would you be willing to wait in between batches of samples?

    That kind of interruptions, especially when you're in a flow, I find a bit annoying. If the machine learning would keep running in the back and you can just continue labelling. Because in principle, the model does not need it. I notice that if I stop for a bit I get distracted.

12. How much of the dataset would you be willing to label to create an annotation? I think half of it. But it does depend on the size of the datasets, the more you have labelled, the better it should be. How more you start to expect. 25% is already really nice.

13. If/when you have to annotate a dataset in the future, would you consider using this tool? Why/why not? Can you foresee any issues?

    Yes. It has already labelled 25% for me. You just get that for free, that works very nicely. Maybe more mistakes do creep in with the automatic labels, because you don't really want to check that anymore. In the beginning you get a lot of trust, lying and sitting it does correctly. Later you might end up at difficult parts. And maybe then it starts going wrong.

14. Is there an important aspect of annotating in general that the tool currently overlooks?

    No, I don't think so.

15. Is there anything you would like to add?

    I think it is a cool project. I would like to be able to browse myself. That I can select myself.

E.6.2 User 2

For this user no samples ended up being labelled automatically. Therefore, they were shown the automatic annotations for a different annotation process Responses and observations recorded after these annotations were shown are shown in a different colour.

Think-aloud protocol

I am trying to label my first. I click on it. [*Becomes full screen, tries again and selects video*.] and I think I have selected it.

That should be walk and then I click apply.

This is also a walk in my opinion and then apply.

Continues going through the videos one by one.

Okay, then I get new ones, that goes automatically.

So I can also just select a couple of walks if I see them.

Yes, Apply.

Two sitting samples are labelled incorrectly, by forgetting to switch label.

Accidentally labelled a Lie sample as Unknown.

Bug found: Other and Unknown cannot be found in the manual labels tab and thus cannot be corrected.

Clicks next to bullet to select label, this area is not selectable.

It is a lot of clicking, but I know it will gain me something. So it is not that bad.

Bug where white gif is shown, fixed by letting user reload the image.

Almost forgets to change label.

Doubtful, but I go for walking.

[*Doubting about standing or walking.*] These remain issues for me… what is exactly happening.

When you are doing this it is funny that when you need to annotate animal body points. How much work that is. This is less calm, because there you just go from image to image. At some point you get square eyes.

Does not notice that the requested 108 labelled samples have been obtained and continues labelling.

Sit is one that is very easy. But this is great, very convincing. These are all correct. There is no walking and no lying, that is unfortunate. That is disappointing. Because that was my issue. These stands are actually stand [no in between cases]. This is convincing.

Interview

1. Overall, how did you feel about using the tool?

   The clicking with the mouse, is very exhausting. Besides that, I think the tool is fine in use. Annotation progress I did not use. Upload and home also not, there are more choices, I did not have to use. Very straightforward. At the moment you have actually only 4 real labels, is uncluttered. When you go to 10 classes it becomes more difficult. Unless a big part is very unique. I liked that you can look, the last time I was annotating stand, so I can continue with that.

2. What parts of using the tool did you like?

   See previous.

3. What parts of using the tool did you dislike?

   Nothing, I like how it works. Only the continuous clicking is tiring. Using touch screen would make it easier for people. I was thinking, I selected these, these these and then I still need to press Apply. But otherwise, the chance of making mistakes is bigger. I would rather press Apply than go back in the dataset and fix them.

4. How did you feel about determining which behaviours were occurring in the various video clips?

   Generally, very easy, only if one leg is moving and the subjects stays in place. That is a gray area. If you annotate to find things automatically… I would annotate the movement of one leg as walking.

5. How did you feel about assigning labels to the video clips once you had decided on behaviours?

   Straightfoward. First you click on the video and think, this is not how I should do it. Once you figure it out, it is very easy. I would change the working of Unlabelled, for example to To label. Then it is more clear, I need to select this and then this will be labelled.

6. How did you feel about the length of the video clips?

   Totally fine. For this cause I think it is fine, but the moments that one leg is moving you want to know what happens next. If after that a second movement comes then it is walking. The duration is fine, if the videos are longer you also need to look longer to check whether a change will occur.

   [And for other datasets] I think so, but I cannot know it like this. [I annotated with] mice and rats videos that were also 2 seconds I think. That was difficult to check for what would happen. For human behaviour and these behaviours it is fine. Interaction between people would make it more difficult.

7. How do you feel about the quality of the annotation? Are the behaviours labelled as you want them to? Why/why not?

   See next.

8. If not, when would you be satisfied with the quality?

   That is difficult, 100%. I get presented with what it knows with 80% certainty. Difficult starting point. I would expect it to present the uncertain samples. When will you start trusting the model to be good enough. Do I need to verify everything?

   You need to be convinced as end-user that the system does what you expect it to.

   I would like a way to be shown the doubtful moments, the others should all be labelled correctly.

   I cannot imagine the results are the same if you also take sequences in mind.

   When I set the target precision at 90%, I expect a new piece of white to appear. I can then continue annotating. If that is a manageable amount. Based on the 2 second clips see what happens before and after. Is that the same? If it is the same every time, it would be a waste to need to look at them again and again.

9. Do you feel like the tool gives you sufficient information to determine the quality of the annotation you are creating?

   I think this is where I should see this. This is automatically labelled and this still needs to be. All those are clips of which the model is not sure what to do. That is not a lot. Then I would get discouraged. But it needs to get better. If I changed the target precision to 90%, would the automatic labels get even smaller? So if you have labelled 109 by hand, then it can label 75 with a certainty of 80%. If I have labelled 200, I would expect 200 or maybe more to be labelled automatically. Or is this a wrong assumption?

10. How do you feel about the time it took you to create the annotation?

    Good question. If I bring the model further along. But if I need 10 times as many and only get a little further, that does not make me happy. How quickly does a model learn to deal with this?

11. How much time would you be willing to wait in between annotations?

    -

12. How much of the dataset would you be willing to label to create an annotation?

    10% max. But maybe that is too heavy at the moment. If you have a dataset and you know a lot of future datasets will be comparable so you do not need to do it again [then more would be acceptable].

    In Deeplab Cut you only need to label 100-120 frames and you do that 100 or 200 times. That is a lot of work. But if I can use it, then it is very comparative.

    The difficult thing of video clips is the sequentiality. Right now they are cut based on size, here he is walking and there he is walking further. That is a sequence of walking. I can get that video 15 times now. That is something I would not expect, it all belongs together. I would actually expect that the system would be smart enough to take this in consideration. Then the white part will decrease.

13. If/when you have to annotate a dataset in the future, would you consider using this tool? Why/why not? Can you foresee any issues?

    Yes, I think it's rather clear. However, with the results as they are now, I am not convinced. I do not have any practical applications now. If I translate it to animal behaviour, I am not sure whether it will miss difficult to distinguish behaviour. I have not experienced how the last part will be annotated.

    [You say that you actually want 100% accuracy.] I find that very difficult. I would expect that the parts where the precision is lower, that you can utilize that. I am not sure yet.

E.6.3 User 3

Think-aloud protocol

Will there be an option to add labels in the real version? In the beginning of annotating, you still figure out which labels there are.

[That is not yet implemented.]

It would maybe good to be able to put everything you don't know in the 'Other' category, where you can later label all other behaviours.

Should I select the check box and then the label and click Apply? [Yes]

Then it is greyed out and you have had it.

Is the subject still on the same location? Or is it already Walk?

I have to label 108 of these, that is quite a lot.

In the middle of the video you said?

Nice that you can click and it enlarges, that is quite handy.

Is that a chair? [Yes, that should be Sit]

Users labels everything one by one.

Maybe what you have labelled can be removed when you press Apply, such that you do not get the suggestion to label them again.

Crouching was Standing?

Can you change that if you enlarge it, it is immediately selected?

He [*the subject*] does such random things.

Even if three samples are in a row, subject still labels one by one.

Bug: white gif, but noticed that if the sample is enlarged, you can still see the video.

Is there also a kind of timer? Oh yes, I have labelled 70.

User clicked too quickly multiple times, had to select multiple times.

Looks at how many samples still need to be labelled.

When will it update?

How will you see that it did something?

User labelled 108 samples, waiting for update.

Looks at automatically labelled samples.

It does still say that it is unlabelled [*This still needs to be changed in the code*].

He is lying everywhere. It is nice, he is sitting the whole time.

I hope that if I label more, more are automatically labelled. It is not that if I move up the precision that it will need more?

If I label one more walk sample, will it immediately use that to label more?

I expect that people will turn the [target precision] to 100. I would keep it under wraps or forbid to put it to 100. The first thing you do is turn it to 100, you just want it work. Then they will call support, ‘it’s not working.’

Interview

1. Overall, how did you feel about using the tool?

   Quite good. There are of course still start-up problems, but the idea is perfect, I think. The integration of the tool I would see within a programme, rather than separate. There is already a manual annotation function in Ethovision, I would combine it with that. [How does this function work?] Like The Observer, but a little less fancy.

2. What parts of using the tool did you like?

   The preview [of the videos]. If you hover a video, I would automatically play it and present the middle of the video as thumbnail. I would show the middle of the videos as still images, since that is what you are annotating. I think annotating the middle of the video as it is now, is really influenced by what you have labelled before. If it is already walking and it is a bit of a doubtful case, you will again select Walk.

3. What parts of using the tool did you dislike?

   Not being able to add my own labels.

4. How did you feel about determining which behaviours were occurring in the various video clips?

   Fine. Maybe you can add a description to the labels, like an information button. Such that you can see the ethogram.

5. How did you feel about assigning labels to the video clips once you had decided on behaviours?

   Fine, works well. Logical interface.

6. [*You did not leverage the possibility of batch labelling. Why is that, would you like to use it?*] I did not know it was possible. I think that if you annotate correctly, you need to annotate one by one. But especially with sitting, where it is clear that nothing else will happen, it could be a good option. If it is possible, it is a useful option.

7. How did you feel about the length of the video clips?

   Fine, for this. It is always about the middle of the video anyways. Because that means that everything before the middle of the video is not relevant anymore. I would think it is stronger if you take the first frame and use everything after that as context. [Then it would not be a problem that you cannot see what happens before the first frame?] No, not really, since it is not really relevant.

8. How do you feel about the quality of the annotation? Are the behaviours labelled as you want them to? Why/why not?

   That was fine. The ones where something happened, those are really good.

   [When would you not be satisfied?]

   Honestly, once there is one thing that is clearly not good, then I would start doubting how good it is.

   If it starts making mistakes, I would start doubting when it will be efficient.

9. Do you feel like the tool gives you sufficient information to determine the quality of the annotation you are creating?

   I don't think that is possible in the long run. I have no idea what is good. It now says 80% is good enough. It says it is 80% certain that is Lie. That does not tell me much. You will go for visual inspection anyway. If there is too much data, you will not do this and you need to start trusting the system. I would remove the target and set a target as a company.

10. How do you feel about the time it took you to create the annotation?

    Fine

11. How much time would you be willing to wait in between annotations?

    I don't know, I would keep it within seconds. 10 seconds is too long. You are in a flow and then it takes too long. 5 seconds would about be the max. [Points out the progress text underneath the progress bar] I would remove that, that is already here. Just give this to a UX designer.

12. If/when you have to annotate a dataset in the future, would you consider using this tool? Why/why not? Can you foresee any issues?

    Yes. As long as it is accurate. [Would it need to be 100% accurate?] No you can't do that anyway. Keeping 95% would be nice. There may be some false positives, but it needs to be accurate.

13. How much of the dataset would you be willing to label to create an annotation?

    Depends on the size of the dataset. I would aim for 50%, as maximum. If [the tool] does 70-80% it is still better, than nothing, but it also needs to feel like it is better. If you see that you need to label 50% and 50% is automatically done, you will be happy. But it should likely be less, right? I am assuming if you have 1600 samples and you have labelled 500, it should be able to label the others.

    For human subject, so much happening, for animal datasets they are generally in standard cages. Maybe it will go quicker.

14. Is there an important aspect of annotating in general that the tool currently overlooks?

    Yes, defining an ethogram. For more difficult behaviours, you need a definition.

15. Is there anything you would like to add?

    No. Very good that there is an Other category. I would replace Unknown with obscured. I would put obscured with Other.

E.7 Feedback user interface

In Table E.3 feedback regarding the user interface is summarized, as gathered during the user study.

Table E.3: User interface.

Positive points
Logical interface (3)
Issues
User friendliness and speed can be improved (1) The clicking with the mouse is very exhausting (2) The selection area of the video clips is too small (1) With more [behaviour] classes it might become cluttered (2) Manual samples are not sorted by the order in which they are labelled (1)
Bugs
If user selects clips before previous labels are applied, the selection disappears Sometimes a white image appears due to slow gif creation (can be reloaded and then it works) Samples are still shown as ‘Unlabelled’ in ‘Automatic labels’ tab
Suggestions
Make video selection area bigger (1) Add short keys (1) Replace already labelled samples immediately or remove them (1 ,3) Change ‘Unlabelled’ label to ‘To label’ would make it more clear (2) Add an ethogram (3)