IN·DEI·NOMINE·FELICITER

and

Squadra
Machine Learning Company

Human-in-the-Loop: Active Learning Approach to Image Classification using CLIP

Internship Report
July 11, 2023

Roel Duijsings (s1086375)
MSc Artificial Intelligence, Faculty of Social Sciences
roel.duijsings@ru.nl

Supervised by:
Karen Beckers, Squadra MLC
Dr. Serge Thill, Radboud University

Contents

Abstract

Image classification can be a labor-intensive job, because of manually labeling the data. Previous research within Squadra Machine Learning Company has focused on the capabilities of large multimodal models for image classification and feature extraction. It was found that the zero-shot capabilities of these models do not always suffice and fine-tuning is necessary. This paper investigates the possibilities of fine-tuning an image model for few-shot classification. For this, a human-in-the-loop Active Learning approach was used to train OpenAI's CLIP model without a highly labor-intensive labeling task. The research question that will be tried to answer is: "What is the impact of different uncertainty measures on the performance of Active Learning in few-shot classification tasks, and how does this approach compare to traditional machine learning?" The report discusses the four main uncertainty measures and compares the performance of Active Learning to traditional machine learning. Results show that entropy sampling and margin outperform traditional machine learning and are potential approaches to use at Squadra Machine Learning Company.

1 Introduction

Large image models such as CLIP and ResNet have revolutionized image classification. Recently, Squadra developed a model that achieved 90% accuracy over 4000 classes using a text-to-image model and no training data. Naturally, we are not always so lucky. These models are not so accurate in specialized tasks such as the classification of engine parts. Here, there are several competing approaches to achieving high accuracy with as little data as possible. All of these approaches make use of a model that has been trained on a very large scale, which is then fine-tuned with a small, manually labeled dataset. An implication of fine-tuning is the need for training data. Unfortunately, this data is rarely available and creating a large, high-quality dataset is very labor-intensive. To overcome this problem, a human-in-the-loop approach can be used. With this approach, a human and a machine learning model work together by creating a feedback loop where the human scores the output of the model. This human judgment will again be used to further train the model. A promising approach to look into is Active Learning, in which the classification model is allowed to choose the data from which it learns. It decides what data points are the most valuable to learn from and selects these to be labeled by the Oracle. The Oracle is also called the learner or supervisor. Active Learning's key advantage is its potential to significantly reduce the number of data points that have to be labeled. This could result in substantial financial savings and personnel efficiency for businesses like Squadra Machine Learning Company.

With this context, the objective of this internship was proposed as:

What is the impact of different uncertainty measures on the performance of Active Learning in few-shot classification tasks, and how does this approach compare to traditional machine learning?

To address this question, we will develop an Active Learning pipeline. This will enable us to investigate which uncertainty measures yield the best results and to evaluate the benefits of Active Learning in comparison to traditional machine learning. In our analysis, we will utilize the CLIP model.

2 Active Learning

Active Learning is a selective data labeling strategy that can significantly reduce the amount of data needed for training supervised models [6]. The core concept of Active Learning involves the model being able to query an "Oracle" (usually a human annotator) to label new, previously unlabeled data points. This is often used in situations where there is a large pool of unlabeled data, and the aim is to choose the most valuable samples to be labeled. The underlying assumption is that the model's performance can be improved more effectively by strategically selecting the most informative data points, instead of randomly choosing data points for labeling. In terms of performance, Active Learning often provides a marginal improvement over traditional manual labeling. To deal with costly labeling, it can also be combined with other methods, such as semi-supervised learning, data augmentation, and weak supervision, to further improve performance. Each of these strategies has its strengths and limitations, and the choice of strategy will depend on the specifics of your data, task and resources.

2.1 Workflow

The typical Active Learning procedure involves the following steps, and is presented in Appendix A:

1. Manually label a small subset L of unlabeled data pool U.
2. Train the model on this labeled subset L.
3. Using this trained model, assign a class probability to the remaining unlabeled data pool $U - L$.
4. For each unlabeled data point, compute an uncertainty score and rank these points accordingly.
5. Query the Oracle to manually label the top-k data points with the highest uncertainty B.
6. Add the newly labeled data B to the labeled data pool L.
7. Repeat steps 2-6 with the expanded labeled data pool $L + B$.

2.2 Design Choices

Implementing Active Learning involves several significant design choices such as the selection of potential samples, the measurement of the usefulness of samples, and determining the number of samples to forward to the Oracle for labeling. A central element of these design choices involves selecting the correct query strategies, such as Uncertainty Sampling, Query-by-committee and Random Sampling. These strategies are crucial in the active learning process as they quantify the confidence of the model's predictions. In this internship, we test the performance of the different algorithms of Uncertainty Sampling.

2.2.1 Uncertainty Sampling

Many of the classifiers we use, do not output just one single prediction, but a ranking of possible classes for a data point. We can use the conditional distribution output by our classifier P(Y|X) to sample points near the decision boundaries. This allows us to pick the samples that are near the decision boundary or most uncertain. There are multiple ways to measure uncertainty in Active Learning. Below we define the four most used sampling methods.

- **Smallest Margin (SMU):** This measure focuses on the top two most probable classes, calculating P(max) minus P(max-1). The lower this margin, the higher the uncertainty about the correct class.

- **Least Confidence (LCU):** This measure identifies the example for which the classifier is least confident about the most probable class. LCU selection only considers the most probable class and selects the example that has the lowest probability assigned to that class for relabeling.
- **Entropy Reduction:** This measure computes the entropy over the predicted class probabilities for every unlabeled example in each active learning step. The example with the highest entropy, which represents the highest uncertainty in class membership, is selected for relabeling. Given an image's prediction probability distribution $P = [p_1, p_2, ..., p_N]$ for N classes, the uncertainty score is calculated as: $score = -\sum_{i=1}^{N} p_i \log_2(p_i)$
- **Random Sampling**: This measure, unlike the others, does not aim to pick the most uncertain samples. Instead, it randomly selects examples from the pool of unlabeled data. This method serves as a control or baseline measure to evaluate the effectiveness of the other uncertainty measures.

2.2.2 Other Critical Factors

In addition to deciding on suitable uncertainty measures, it is equally essential to consider a few other critical factors. These include:

- Determining the initial number of samples to label (at iteration zero);
- Specifying the size of your ranking list (that is, how many samples you score in each iteration), and from this list, determining the top samples to select for labeling;
- Identifying when to terminate the labeling process;
- Deciding whether to train the model on all labeled samples or only those labeled in the most recent iteration;
- Considering if it's best to start with a fresh model each iteration or if you should fine-tune the same model in every iteration;
- Assessing the accuracy of the Oracle.

2.3 Active Learning and CLIP

Contrastive Language–Image Pretraining (CLIP) is a multimodal neural network model developed by OpenAI [5]. It is trained on a wide variety of images and their associated text captions, allowing the model to learn the relationships between images and natural language. CLIP is trained using contrastive learning, which aims to maximize the similarity between matching pairs of images and text and minimize the similarity between non-matching pairs.

In the context of Active Learning, CLIP can be used as the underlying model that provides predictions on unlabeled data and uncertainty scores for each data point. The most uncertain data points, according to CLIP's output, can then be sent to the Oracle for labeling. As such, Active Learning could be used to fine-tune CLIP on a specific task, iteratively improving its performance by prioritizing the labeling of the most informative examples.

3 Problems with Active Learning

3.1 Class Imbalance

In a multiclass setting like this, imbalanced classes can be a serious issue. If some classes have much more representation than others, the model can become biased towards those classes, reducing the overall performance. Possible solutions to class imbalance are oversampling the minority class, (proportional) stratified sampling or early stopping [2]. In this project, to mitigate class imbalance, we strategically selected only the top 40 classes, each containing a volume of images ranging from 1,700 to 25,000. This approach effectively minimizes the class imbalance issue.

3.2 Sampling the whole unseen dataset

In classical Active Learning, the search for the most informative samples is done through the entire unseen dataset. Each iteration of Active Learning involves the recalculation of the uncertainty scores of each sample for the new model. Thus, for large datasets, searching the entire training set is very time-consuming and computationally expensive. Several strategies can be employed to prevent this overprocessing:

- Batch Active Learning[1]: Instead of selecting a single instance in each iteration, Batch Active Learning selects a set of instances for annotation. While it's still necessary to score all samples, it reduces the total number of iterations, hence the computational cost.

- Approximate Ranking[3]: Instead of searching for the most uncertain samples, one could use approximate ranking algorithms. These algorithms aim to identify a subset of top-ranking instances rather than ranking all instances.

- Data Subsampling: A straightforward solution is to randomly select a manageable number of instances from the unseen dataset for scoring. However, this method carries a risk of missing important samples.

In this project, due to limited computational resources, we adopted a Data Subsampling approach, capping the number of unseen images to be scored at 300. We randomly selected these images and ranked them based on their uncertainty score. This count of 300, while being computationally economical, also allows for a robust sampling of the most informative samples, eliminating the need to score the entire dataset.

3.3 Computational Efficiency

In classical Active Learning, for each iteration, the model has to recalculate the uncertainty score for each sample and also has to train the model. This process can lead to an increase in resource consumption, which should be carefully considered. A good balance between the size of the ranking set, and the desired performance of the model is vital.

4 Dataset

The initial dataset we used consisted of over 360,000 images of automotive parts from 205 classes. Upon analyzing this dataset, we found that it posed significant challenges for classification models. Classes are closely related to each other, but within classes, there are large differences. Previous attempts by MLC at zero-shot classification using CLIP and ResNet50 on this dataset have yielded an accuracy of only 32%. Given these reasons, we decided to use a more suitable dataset: a fashion dataset. This dataset contains 198 classes of different fashion items, such as T-shirts, jackets and dresses. The dataset is imbalanced, so we decided to only focus on the top 40 classes with the most items. In total, this dataset consists of 246,841 images.

5 Methods

In this section, we will describe the methods used for this research. First, we describe the methods used to compare the performance of the four most used uncertainty measures. Next, we describe the methods used to compare the performance of Active Learning to traditional machine learning and to evaluate Active Learning's added value.

5.1 Methodology for Comparing Uncertainty Measures in Active Learning

One of our objectives is to compare the performance of the four different uncertainty measures to determine which one is most effective. To achieve this, we designed an Active Learning experiment where we could test the performance of the four different uncertainty measures

For each run, at iteration zero we randomly select three images from each of the 40 classes via stratified sampling. This gives us an initial set of 120 annotated images. These samples are used to train our model. Next, the model makes predictions on 300 unseen samples. This returns a class probability for each image. Based on these probabilities, we calculate an uncertainty score using the chosen uncertainty measure. The images are ranked by their respective uncertainty scores. The top 10 images, which represent the most informative samples based on uncertainty, are selected, annotated, and added to our pool of labeled data.

In the next iteration, we have an enlarged labeled dataset: the original 120 samples plus the 10 newly annotated ones, resulting in a total of 130 samples. This process of selection, annotation, and retraining is repeated over 10 iterations. Each uncertainty measure is tested over 10 such runs. In total, this results in 4 uncertainty measures * 10 runs each = 40 runs. For validation, we select 200 images from the validation dataset and let the model make a prediction on the image and label combination. We calculate the top 1 and top 5 accuracy per iteration.

5.2 Methodology for Comparing Active Learning and Traditional Machine Learning Performance

To assess whether Active Learning outperforms traditional machine learning, we set up a test for traditional machine learning as well. For the traditional machine learning test, to ensure similar conditions, we also start with 120 samples chosen via stratified sampling at iteration zero. This model, starting with a baseline CLIP model, is fine-tuned and evaluated on 200 validation samples. For 10 iterations, we increment the sample size by randomly selecting an additional 10 samples per iteration. Notably, for each iteration, we start anew with a basic CLIP model and fine-tune it—rather than continuing to retrain an existing model—thereby ensuring each model is independently developed.

In contrast to random sampling, where we retrain the same model, we start with a baseline model every iteration. This strategy allows us to compare the performance of Active Learning and traditional learning, both having access to the same number of samples. It enables us to understand the value of retraining the model on the most informative samples, as opposed to training the model just once on that number of samples. In essence, this approach evaluates the performance of few-shot classification with that of traditional classification.

6 Results and Discussion

This section evaluates and discusses the results of the project. First, we discuss the results of the four uncertainty measures. Second, we discuss the results of the performance of Active Learning vs. traditional machine learning, to find out the added value of Active Learning.

6.1 Uncertainty Measures

To evaluate the performance of the different uncertainty measures, we executed 10 runs for each measure, where each run consists of 10 iterations. So we started with a stratified sampling of 120 samples and added 10 samples per iteration. Figure 1 shows the plots of the accuracies and trend line of the individual uncertainty measures. Each line represents a single run, and the dashed line represents the trend line. This trend line represents the average trend of the accuracy across the different runs and iterations. It indicates how the accuracy tends to evolve, on average, as the number of iterations increases.

The plots display that for each type of uncertainty measure, the individual runs are quite unstable. This might suggest a degree of inconsistency but can be explained by the nature of Active Learning. Since the model selects the most informative samples, it chooses the ones that are the farthest away from its decision boundary. This results in a greater likelihood of encountering diverse and more challenging samples in each run, which can introduce fluctuations in the outcome.

However, the trend line, which represents an average across all runs, depicts a positive trend with an increase in the number of iterations. This trend implies that as the model continues to learn from increasingly challenging samples, its accuracy improves on average. So, even though there can be some fluctuation in individual runs, the overall trend aligns with what we expect from Active Learning: as the model is exposed to more challenging samples, it is learning and hence becoming more accurate on average.

One of the objectives of this study is to identify the best uncertainty measure. To achieve this, we overlay the four trend lines in one plot. As shown in Figure 2a, the *margin* and *entropy* measures show a steeper slope than the other two. This suggests that these measures aid the model in enhancing its accuracy at a quicker pace. Interestingly, it appears that from the seventh iteration, *margin* begins to outperform *entropy*. However, it's noteworthy to mention that the four trend lines do not start at the same accuracy value at iteration zero. This is because the selection of samples at iteration zero is done using stratified sampling. We select exactly three samples per class at iteration zero, so there might be some randomness in the images, where some runs might start with more informative samples than other runs. Because of this, the model might have a head start in terms of accuracy. This problem could be addressed by increasing the set of samples or the number of runs. This will require more computational resources.

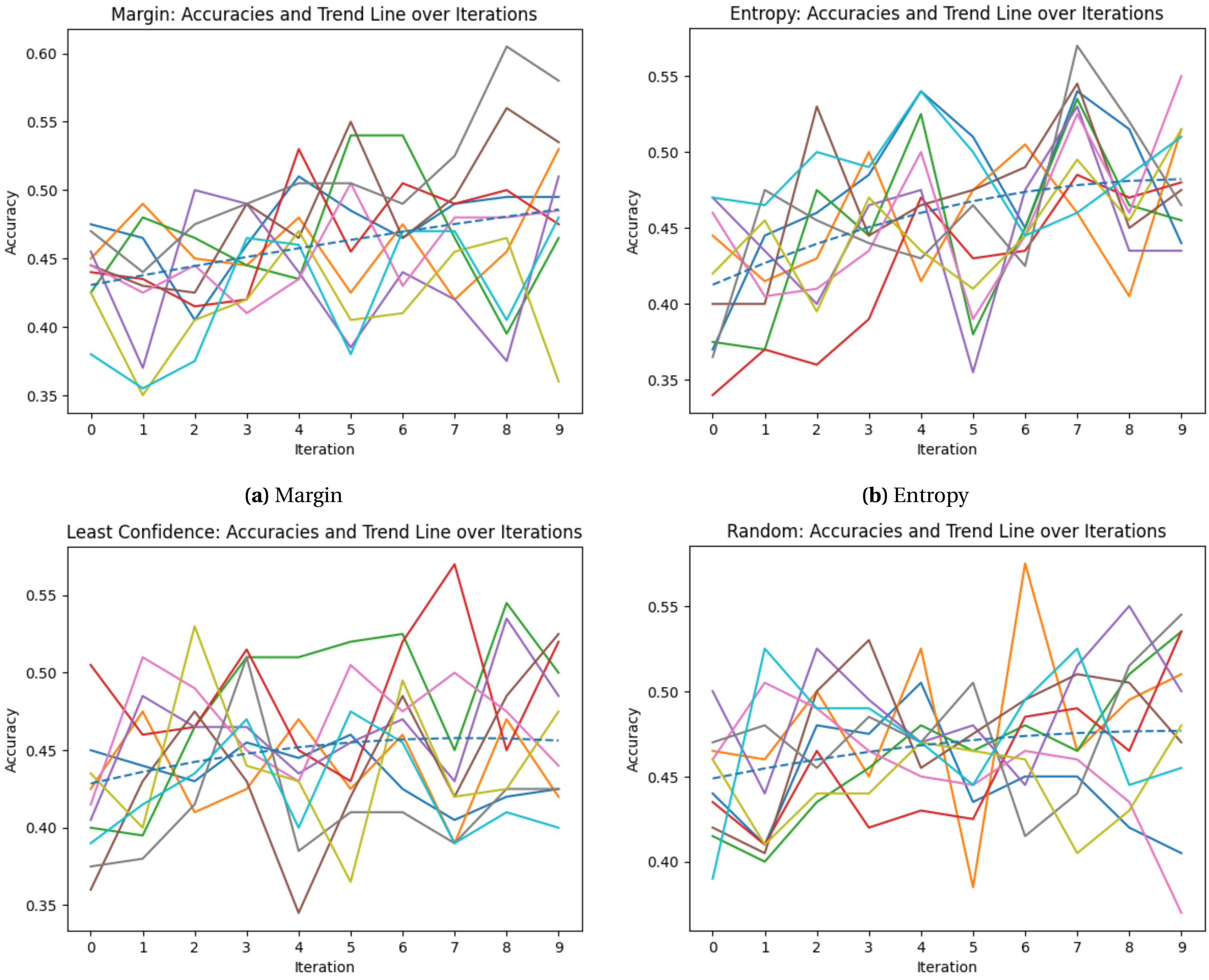
One more approach to compare the performance of the uncertainty measures is by tracking their progress after iteration zero. In Figure 2b we demonstrate the increase of accuracy in percentage points. For instance, we see that after 10 iterations, the model using entropy sampling has increased by 7 percentage points. *Margin* seems to show a constant increase, although it appears to be converging slightly towards the end. When comparing the four uncertainty measures, we see that *entropy* and *margin* outperform *least* and *random*.

Entropy achieves the most rapid accuracy increase in the initial iterations, before leveling off in the later iterations, reaching its peak accuracy around the ninth iteration. In contrast, *margin* shows a consistent accuracy increase, suggesting that it has not yet reached its peak. This implies that if we were to increase the number of iterations, the accuracy of the model using the margin measure could improve further.

As depicted in Figures 2a and 2b, it can be inferred that for a smaller number of iterations, entropy sampling would be the optimal choice. However, for a larger number of iterations, margin sampling seems to be more effective. Future studies could test this hypothesis by conducting tests with more than ten iterations.

6.2 Active Learning's added value

Now, since we have evaluated what are the best uncertainty measures, we can compare the performance of Active Learning to traditional machine learning. This way we can evaluate whether Active Learning is



(a) Margin

(b) Entropy

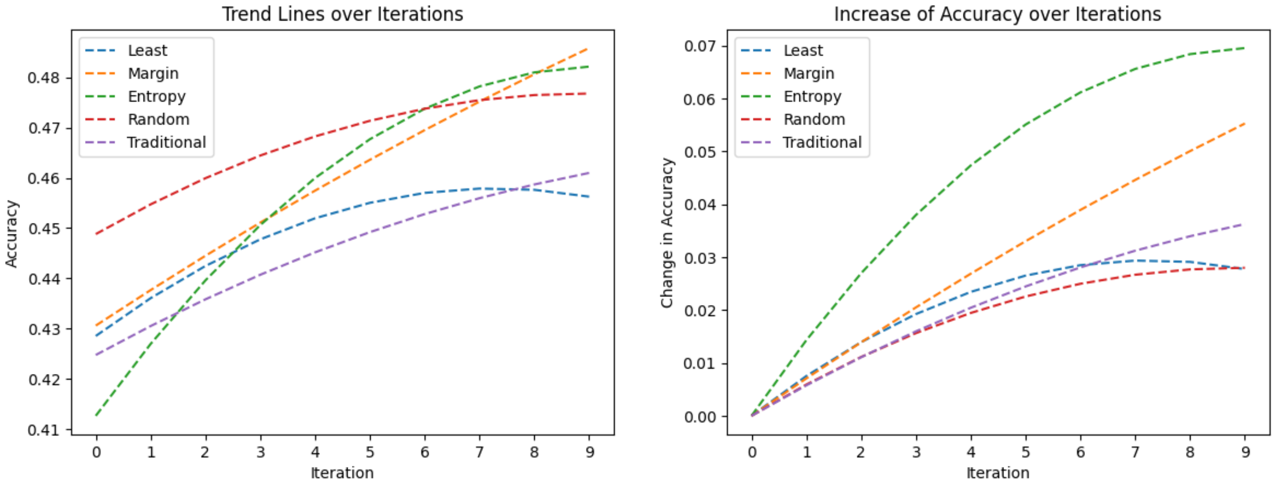
(c) Least Confidence

(d) Random

Figure 1: Results of the different uncertainty measures. Each measure was performed for 10 runs, each consisting of 10 iterations. The dashed line is the two-dimensional trend line. These trend lines serve to illustrate the average performance trajectory across all runs.

worth it for Squadra MLC. In Figure 2 we can see the results of the four uncertainty measures and traditional machine learning. For traditional machine learning, we did not perform multiple iterations. The iteration axis stands for the number of samples that the model was trained on. So for example, in iteration zero the model was trained on 120 samples and in iteration nine the model was trained on 120+9*10=210 samples. What we can see in Figure 2a is that the trend line of traditional machine learning seems to grow steadily, but not as strong as *margin* and *entropy*. *Traditional* does outperform *least*, especially in the higher range of iterations. From Figure 2b we can see that *traditional* drastically does not increase as much per iteration as *margin* and *entropy* do. *Traditional* seems to have the same increase as *least* and *random* up to about iteration six. After iteration nine it outgrows them.

From these two plots, we can conclude some interesting findings and answer our research question. After ten iterations, traditional machine learning has an increase of three percent points, while *margin* increased by about five percent points and *entropy* even by seven percent points. Indeed, Active Learning looks to outperform traditional machine learning if you use entropy sampling or margin sampling. In terms of increase over ten iterations, Active Learning can double up on the increase in comparison to traditional machine learning. To answer our research question: the impact of the different uncertainty measures depends on the specific measure. While entropy sampling and margin sampling outperform traditional



(a) Trend lines.

(b) Increase of accuracy.

Figure 2: The trend lines and their increase for the four uncertainty measures and traditional machine learning.

machine learning, least confidence sampling and random sampling do not. In comparison to traditional machine learning, Active Learning seems to be a beneficial approach in terms of accuracy. Using an Active Learning approach with entropy sampling or margin sampling can significantly boost the increase of accuracy by about two to four percentage points.

7 Limitations

The findings of this internship project should be viewed in light of certain limitations which could potentially impact the results.

- Due to performance constraints, we had to limit the number of samples and iterations. Initiating with more samples in iteration zero, or implementing more sampling for ranking, might have resulted in more accurate insights into the performance of the uncertainty measures.
- We opted for stratified sampling for this study. Future research could investigate the impact of random selection at iteration zero to further optimize the process.
- The dataset utilized may not be representative of all possible scenarios. Different datasets may yield different results and further research should encompass a variety of datasets to provide a more comprehensive overview of the model's performance.
- Due to computational limitations, we could only score and rank 300 images, rather than the entire training dataset. This disproportionately represented high-volume classes in the ranking, leading to an underrepresentation of potentially informative low-volume classes. This limitation undermines the effectiveness of uncertainty sampling and may lead to an underestimation of the potential of Active Learning.

8 Conclusions

Active Learning has always been seen as a potentially beneficial approach to machine learning problems. The primary benefit of Active Learning is that it can potentially provide high accuracy with less labeled data compared to standard supervised learning. By prioritizing the most informative or uncertain data points for labeling, Active Learning can make efficient use of an Oracle's time and effort.

However, Active Learning also comes with several challenges and drawbacks. For instance, it assumes that you have access to an Oracle that can provide accurate labels when queried. In reality, labeling can be expensive, time-consuming, or even impossible in certain instances. There can also be cases where the model's uncertainty does not align with the true informativeness of a sample. Furthermore, Active Learning can be susceptible to bias, as it inherently prioritizes data that the model finds difficult over potentially easy but still informative examples. Lastly, while Active Learning can be efficient in terms of required labeled data, it can be computationally expensive, as it typically involves retraining the model several times.

This report evaluated an Active Learning approach to an image classification problem with the use of OpenAI's CLIP model. The four main uncertainty measures were discussed and their performance was compared. Finally, the added value of Active Learning compared to traditional Machine Learning was assessed.

The results show that entropy sampling and margin sampling significantly outperform traditional machine learning. These results suggest that an Active Learning approach with the use of entropy sampling or margin sampling will boost the accuracy of the model on image classification problems.

This project might serve as a foundation for potential enhancements in future research. First of all, increasing the number of samples used in the Active Learning process will increase model accuracy and thus give a clearer representation of the benefits of uncertainty sampling. Second, increasing the number of iterations will show a better picture of the limits of margin and entropy sampling. Lastly, our use of stratified sampling at iteration zero might not reflect real-world scenarios. Therefore, studying the impact of random sampling, despite potentially having fewer labeled classes initially, could be a beneficial area for future research. This will test the robustness of Active Learning in less balanced conditions.

9 Internship Activities

9.1 Onboarding and Learning

The internship took place from March to July of 2023. Most of the work was done either at the office of Squadra MLC in Oss (16 hours a week), or at home (4 hours a week). For the first two weeks, I had the pleasure of meeting my colleagues at Machine Learning Company and familiarising myself with the organization. We discussed the internship objectives and laid out an initial plan. We also reviewed previous research undertaken at MLC. My first steps included a preliminary exploration of Active Learning, leading to the development of the internship proposal. I also delved deeper into the concepts of Active Learning and Semi-Supervised Learning, and the CLIP and ResNet models. Upon comparing these two human-in-the-loop methodologies, it became evident that Active Learning was the most promising choice for our purposes.

9.2 Building the Active Learning Pipeline and Training Models

We decided that this internship would focus on investigating and building an Active Learning pipeline for MLC. A detailed flowchart was designed, to visualize all components of the Active Learning pipeline, providing a complete schematic representation of the process (Figure 3). During the third week, I began developing a skeleton code, drawing inspiration from the constructed flowchart. This led to the creation of a basic framework for the pipeline. Given this skeleton, I performed more research into how these components could be filled in.

As part of the process to establish a user interface for sample annotation, I explored Label-Studio, a well-known tool in this domain. Simultaneously, I considered building a custom label interface to cater to specific needs that could arise during the process. Label-Studio offers a valuable feature in the form of a useful machine learning integration. This allows a seamless incorporation of Label-Studio into your custom machine learning pipeline, enhancing functionality and creating a smooth and professional workflow.

I also experimented with the essential functionalities of ResNET, a potential classifier. A significant accomplishment was setting up CUDA for GPU processing, as this significantly improves the performance by leveraging the power of the GPU for processing tasks. The successful integration with Label Studio enabled the automatic export of labeled tasks, thereby streamlining the training process. However, I encountered some issues with inputting training data correctly. I opted to revisit PyTorch basics to ensure a stronger foundation. I identified a series of tutorial videos on YouTube that provided valuable insights.

9.3 Optimization and Evaluation

The YouTube tutorials significantly bolstered my understanding of PyTorch, enabling me to build custom ResNet18 and CNN models. I now have a clear grasp of the components required in a training loop and the format of input data. I created a test subset of 10 classes of 27,616 images. This initial training, conducted on a CPU, required over five minutes for the first epoch, prompting me to halt the loop. Upon realizing I had installed the CPU version of PyTorch, I corrected this and installed the appropriate version for GPU usage. The training time was considerably reduced, and I achieved a 66% accuracy rate with the default model. I anticipate that further adjustments will be necessary, but I am optimistic given the progress made so far.

The training loop has now been optimized to incorporate various hyperparameters, allowing users the flexibility to determine the specific classes for training, and set the number of epochs, among other customizable parameters. In addition to using manually selected classes for training, I've also integrated an option to randomly select a predefined number 'N' of classes along with a specified number of samples per class. This addition enhances versatility and enables a more robust exploration of the dataset. We had a close look at the dataset again. It was observed that some classes exhibit close correlations, potentially affecting the validation accuracy. We were not content with the results of ResNet50. A colleague advised me to look into CLIP. This combines both the image data and the text of the label. We might gain extra information about the image by using this text. I build the training loop with the CLIP model and compared its performance with ResNet. We found out that CLIP outperforms ResNet and therefore decided to use this model in our Active Learning pipeline. The next crucial component of our process is the ranking of unseen images. For this purpose, we employ Uncertainty Sampling, with a particular emphasis on Margin Sampling. This method aids in identifying and prioritizing those images that the model is most uncertain about, thus facilitating a more efficient learning process. By giving each image an uncertainty score, we can make a ranking and pick the top X images to be labeled by the Oracle. This was the last part of the training phase. Despite encountering some errors, I significantly enhanced my understanding of Label-Studio integration, backend servers, and data pipelines.

Now we continue with the test phase, to determine whether Active Learning is actually beneficial for MLC. For this, we used the automotive parts dataset and we conducted numerous runs of 10 iterations with multiple hyperparameters. I was not satisfied with the results as there was no clear improvement in accuracy over the iterations. I thought this was because of mistakes in the code, so Karen and I debugged the code for possible mistakes but found none. We made a plan for subsequent steps. The disappointing results might be due to the challenging dataset, or perhaps the ranking size is insufficient? The disappointing results caused significant frustration. Although there was no apparent improvement with each iteration, a change in the dataset to a fashion dataset resulted in an unexpected outcome. When we averaged the accuracy per iteration across several runs, we noticed a gradual increase. Does this imply an issue with the initial dataset? Should we shift from margin sampling to entropy-based uncertainty sampling for improved results? We tried doing this. However, entropy sampling did not appear to enhance accuracy over time compared to margin sampling.

Analyzing the new fashion dataset revealed a substantial class imbalance with one class, 'N/D', accounting for about a quarter of all images. This class stands for 'not determined' as some images do not have a subcategory. We decided to exclude this class, as this is not actually a real class.

We implemented a variety of configurations, each designed to explore the model's performance with different uncertainty measures under certain conditions. We evaluated and discussed the results and wrote down everything in this internship report.

9.4 Employing Label-Studio as a Labeling Interface

Given the iterative nature of Active Learning, the requirement for a capable labeling interface becomes crucial. One such interface that we utilized is Label-studio, renowned for its ability to seamlessly integrate with machine learning tools [4]. I took the initiative of setting up this additional integration, an accomplishment of which I am very proud. It combines both my knowledge of backend and frontend development. This facilitates users to label data, automatically train the model on it, and obtain a ranked score for the next set of images to be labeled. Successfully establishing this labeling interface delivers potential benefits to MLC, especially when the aim is to enable customers to take an active role in labeling their own data. Essentially, this blend of fields not only showcases a practical use of my abilities, but also sets the stage for a more engaging and tailored way of managing data for MLC's customers.

10 Reflection on the Learning Goals

This internship has been an enriching journey of exploration, discovery, and professional growth. It has been a big stepping stone for me, where I've learned a lot - from how things work in a professional setting to getting deep into the world of Machine Learning. It's helped me grow both professionally and personally.

A significant portion of my internship was dedicated to learning and applying machine learning concepts, particularly in the realm of Active Learning. While the initial stages presented some roadblocks, these challenges were not unmanageable. They forced me to rethink my strategies, question my understanding, and experiment with alternatives - a process that I found to be immensely educational. One of the primary goals was to gain a clear vision of the inner workings of a machine learning/data science company. This objective was accomplished by observing various project meetings, participating in team discussions, and one-on-one mentorship sessions. The experiences gave me a nuanced understanding of the complexities and components involved in machine learning at an industry level. I discovered my fascination for the iterative nature of building and refining machine learning models and their potential applications.

The technical skills I developed during this internship are really valuable to me. One major challenge I faced was working with LabelStudio's Machine Learning Backend integration, which was as enlightening as it was demanding. The learning curve involved familiarizing myself with backend servers and APIs, along with getting a grasp of LabelStudio's integration. The troubleshooting process during the prediction phase was an invaluable learning experience. It didn't just help me get better at fixing mistakes, but it also really broadened my knowledge.

Another big part of the internship was working with different kinds of datasets. At first, I wasn't happy with the results I was getting, which made me wonder if there was something wrong with the data I was using. It turned out that the type and quality of data really matters when you're building a machine learning model. Looking back, switching to a different dataset made a big difference in how well the model worked.

Working on different aspects of machine learning models, like tweaking hyperparameters and experimenting with various sampling methods, was a notable part of my learning journey. The outcomes didn't always meet my initial expectations, but every attempt - whether successful or not - deepened my understanding of the intricate elements that can influence a model's performance.

On a personal level, this internship really tested my problem-solving skills. It showed me the importance of not giving up when I faced tough situations. Working through problems, figuring out solutions, and making sense of unclear results challenged my patience and strength. But all these experiences helped me grow. They made me better at handling tough situations and I consider these lessons some of the most important things I learned during my internship.

Looking back, the internship gave me a platform to apply my theoretical knowledge in real-world scenarios, thereby increasing my understanding and giving me a broader perspective of the data science field. I feel better equipped and more motivated than ever to continue my exploration in the ever-evolving world of machine learning. This internship experience has not just met, but exceeded my initial learning objectives, leaving me with a deep respect for the field and a strong wish to keep learning and advancing in it.

11 Acknowledgements

I would like to thank my internship supervisor at Squadra Machine Learning Company, Karen Beckers, for closely working with me and supervising me. She helped me by sharing her experience and provided me with useful insights into the Data Science department. I also wish to express my thanks to Squadra Machine Learning Company for allowing me to work at their company and for providing me with invaluable experience and exposure to practical machine learning applications. Their culture of continuous learning fostered a stimulating work environment, which has greatly enriched my professional development. I would also like to thank Dr. Serge Thill, from Radboud University, for acting as my internal assessor for this internship. His assistance with the administrative aspects and documentation related to the internship was incredibly helpful. Finally, I would like to thank Radboud University, for giving me an opportunity to work in an external company environment as a part of my Master's internship.

References

[1] Gui Citovsky et al. "Batch active learning at scale". In: vol. 34. 2021, pp. 11933–11944.

[2] Seyda Ertekin, Jian Huang, and C. Lee Giles. "Active Learning for Class Imbalance Problem". In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '07. Amsterdam, The Netherlands: Association for Computing Machinery, 2007, pp. 823–824. ISBN: 9781595935977. DOI: 10.1145/1277741.1277927. URL: https://doi.org/10.1145/1277741.1277927.

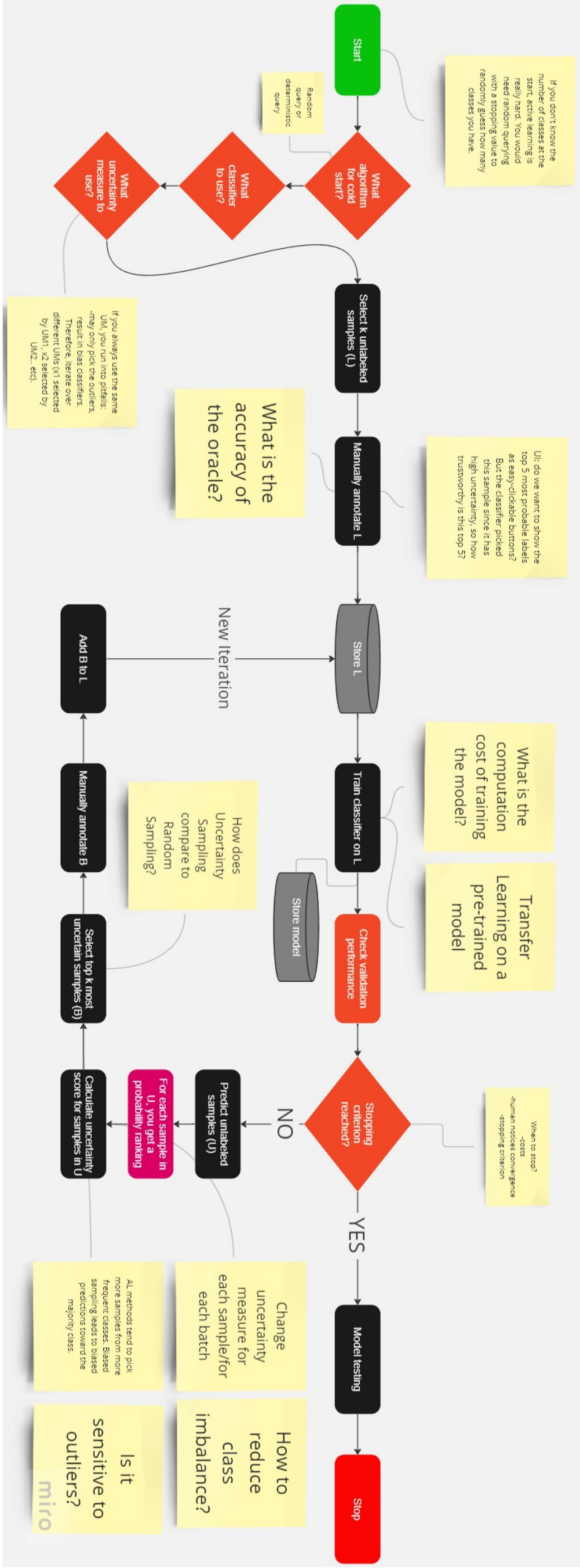
[3] Yunchao Gong et al. "Deep convolutional ranking for multilabel image annotation". In: 2013.

[4] Heartex. *Label Studio Documentation — Integrate Label Studio into your machine learning pipeline*. URL: https://labelstud.io/guide/ml.html (visited on 07/04/2023).

[5] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV].

[6] Burr Settles. "Active learning literature survey". In: University of Wisconsin-Madison Department of Computer Sciences, 2009.

A Active Learning Pipeline Flowchart


Start
What algorithm for cold start?
What classifier to use?
What uncertainty measure to use?
Select k unlabeled samples (L)
Manually annotate L
Store L
Train classifier on L
Store model
Check validation performance
Stopping criterion reached?
NO
YES
Predict unlabeled samples (U)
For each sample in U, you get a probability ranking
Calculate uncertainty score for samples in U
Select top k most uncertain samples (B)
Manually annotate B
Add B to L
New Iteration
Model testing
Stop
miro

If you don't know the number of classes at the start, active learning is really hard. You would need random querying with a stopping value to randomly guess how many classes you have.

Random query or deterministic query

If you always use the same UM, you run into pitfalls: -may only pick the outliers, result in bias classifiers. Therefore, iterate over different UMs (x1 selected by UM1, x2 selected by UM2.. etc).

What is the accuracy of the oracle?

UI: do we want to show the top 5 most probable labels as easy-clickable buttons? But the classifier picked this sample since it has high uncertainty, so how trustworthy is this top 5?

What is the computation cost of training the model?

Transfer Learning on a pre-trained model

How does Uncertainty Sampling compare to Random Sampling?

When to stop? -costs -human notices convergence -stopping criterion

Change uncertainty measure for each sample/for each batch

AL methods tend to pick more samples from more frequent classes. Biased sampling leads to biased predictions toward the majority class.

How to reduce class imbalance?

Is it sensitive to outliers?

Figure 3: The flowchart. See: the Flowchart on Miro

B Source code

The repository containing the code and most important results can be accessed here. (Only accessible by Squadra MLC personnel)

C Recommendation: User Interface for Image Labeling Procedure

When dealing with the task of labeling a dataset containing 200+ classes, careful consideration must be given to the interface provided to the Oracle. Directly displaying the image alongside all 200+ class labels can result in an overwhelming and cluttered visual interface. Therefore, such an approach is generally discouraged. Given that the model generates predictions for all labels for each unseen image, we obtain a highly valuable shortlist of the top 5 labels. Our conducted tests reveal a top 5 accuracy rate exceeding 90%. Therefore, we suggest presenting these top 5 labels as likely options to the Oracle. Because there still might be a chance that the actual label is not in this top 5 list, we recommend the use of a sixth button named 'other' or the inclusion of a text box. This user interface setup offers the Oracle a quick and flexible way of labeling a large number of images, enhancing the robustness and flexibility of the labeling process.