Capstone Project - Active Learning for Object Detection

David Pitts, Daniel Segal

October 6, 2021

1 Abstract

The project consists of 3 parts, which are in the field of Active Learning(AL) for object detection with deep learning models. The first part deals with a specific AL method which is based on weighting classes by their estimated hardness and querying images according to these weights. We call this method “Active Learning by Class Hardness”(ALCH). We found that ALCH generally performed better then random sampling, but not in a statistically significant way. The second part deals with post-hoc correlation analysis of the AL runs from the previous part and finding correlations between different properties of the queried images and the improvement in performance. The third part is about performing experiments for AL strategies which were inspired from properties found in the second part.

2 Introduction

2.1 Active Learning

Active Learning is a class of algorithms in which the labeled data points are chosen according to an Active Learning algorithm, such that the model can achieve good performance with less samples, resulting in faster training and reduced labeling costs. In particular our project deals with pool-based sampling active learning, where the model starts to train on a relatively small set of data which is labeled, and in an iterative fashion queries the pool(a set of unlabeled data points), chooses a set of data points from it to be labeled by an oracle(an entity which labels data points, e.g. a human), the model then trains on the whole data for a number of epochs until convergence, and then proceeds to the next AL cycle to query more labeled data, and repeats.

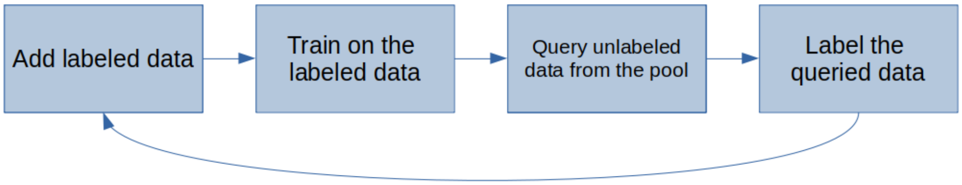

Figure 1: A cycle in an Active Learning loop

Generally, most AL methods rank the data points in the pool by how informative they are to the model, and decide to query the most informative ones. Measuring how informative a data point is can be done in several ways, for example, trying to approximate how uncertain is the model on the data point by the model's confidence scores, the entropy of the predicted labels [1], using an auxiliary model to predict the loss of the model[2].

2.1.1 Active Learning for Deep Learning

Active learning in the field of Deep Learning has some specific challenges, for example - Deep Learning algorithms mostly require large datasets, whereas generally Active Learning methods require the learning algorithm to be able to learn from small datasets. In addition Deep Learning predictions tend to be over-confident, and therefore it might be hard to gain high-quality uncertainty estimates from them.

2.1.2 Active Learning for Object Detection

Active Learning for objection detection is particularly appealing, as the labeling costs for object detection datasets are higher then other image tasks such as basic classification, as there is a need to draw multiple bounding boxes in an image, and there can be multiple classes per image.

2.2 Active Learning by Class Hardness(ALCH)

In this project we research an Active Learning method which works by sampling from a data pool, such that the amount of images we sample from each class(the class which the model predicted, which might not be the real class) is proportional to it's estimated hardness, which is estimated from a fixed validation set after every AL cycle. The process is illustrated in figure 2:

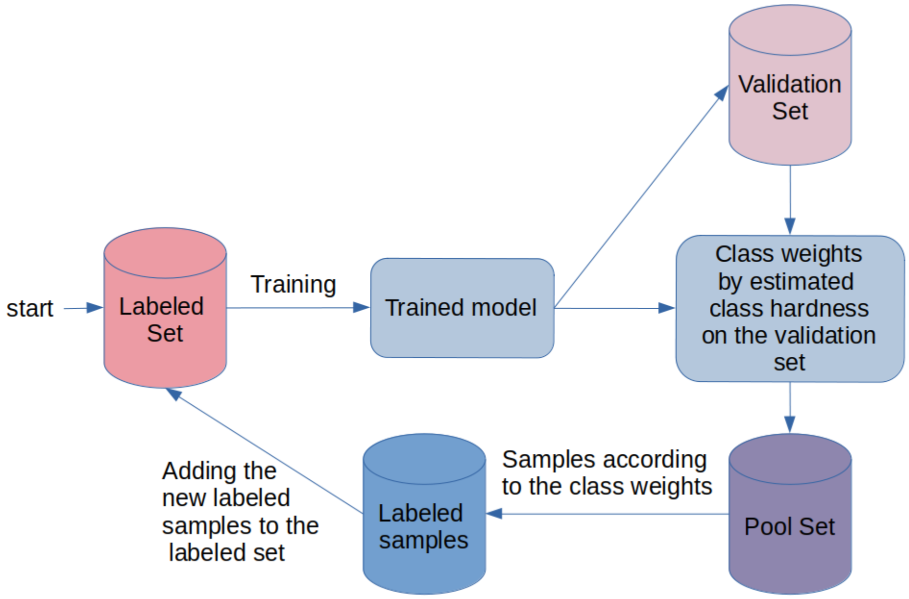
start
Labeled Set
Training
Trained model
Validation Set
Class weights by estimated class hardness on the validation set
Pool Set
Samples according to the class weights
Labeled samples
Adding the new labeled samples to the labeled set

Figure 2: Active Learning by Class Hardness diagram

For a more detailed description of the ALCH method see section 5.8.
The ALCH algorithm makes the following assumptions:

1. Labeling and training the more "informative" data points will yield better performance and faster. This is the general assumption in Active Learning algorithms.

2. Samples which come from classes that the model's performance on is low, are more informative. Hence, we should prefer to label these images. This assumption might be considered problematic when considering it from the point of view of Curriculum Learning - a training strategy which deals with learning examples in meaningful order, from easy samples to hard ones, in order to train faster and achieve better performance[6]. That said, this contradiction can be solved when considering that at the first iterations of the Active Learning loop the model is still not well trained, and therefore the harder classes might not actually be very dominant in the sampling, but they might be later in training, which is actually in accordance with Curriculum Learning.

3. At some stage, the model's predictions are accurate enough such that trusting its predictions when sampling will give more informative images than random sampling.

2.3 Post-hoc analysis for finding potential AL patterns

The second part of the project deals with studying correlations between model performance improvement and potential useful informative patterns for Active Learning, which we call hypotheses/properties. If meaningful correlations are found, they might be of interest for future research. The way we do the analysis is as follows:

1. While running the Active Learning experiments from the first part, record the information of the images the model queried in each AL cycle, the predictions on these images, and the test set performance after every AL cycle.

2. Creating a data set which has this information from the various AL experiments.
3. Forming various hypotheses which are related to model predictions or image properties(detailed below), and extracting them as features to this dataset.
4. Analyzing the relationship between the quantities corresponding to the hypotheses, to the improvement in performance on the test set.

More concretely, we create a dataset, where each row represents a cycle in Active Learning. Some of the columns are explanatory variables related to the hypotheses(for example - average area of predicted bounding boxes on images in the cycle), and some columns are used as control variables, such the number of cycle in the AL loop and the previous test loss. The target variable is the improvement in performance relative to the previous cycle, which we study its relationship with the explanatory variables related to the various hypotheses.

2.3.1 Hypotheses for Post-Hoc analysis

We have formed the following hypotheses:

2.3.1.1 Simple Hypotheses

1. Average number of predicted bounding boxes over the whole images queried in the AL cycle. The motivation is that images with a large number of objects might be harder for the model to deal with, and so should be prioritized for labeling.
2. Average area of predicted bounding boxes over the whole images queried in the AL cycle. The motivation is that images with a large bounding box area might have larger objects, or a more challenging scale, and such images might be worth labeling.
3. Occlusion - the overlap between the predicted bounding boxes in the images. As a large overlap in the bounding boxes might make it harder for the detector to predict objects properly, and such images might be worth labeling. We also refer to this property as "Mean IOU"(where IOU stands for intersection over union), as we measure this by calculating the IOU between every pair of predicted bounding boxes in the image and averaging over this.

2.3.1.2 Distance to the Closest Centroid Hypothesis In addition to the relatively simple hypotheses above, we test a more complicated hypothesis- "Distance to the Closest Centroid Hypothesis" - which involves extracting the feature representation of every sample in the training set by a pre-trained CNN as the feature extractor, possibly reducing the dimension, performing clustering based on these feature vectors, and using the distance of the sample from the nearest centroid(the cluster center) as the property. The motivation is that a centroid can be considered a highly representative sample of the dataset's distribution, and the closest an image feature representation is to that cluster center, the more representative it is of the data distribution, so sampling such images might be effective for learning faster.

This hypothesis is quite complex as there are various possibilities for the number of clusters to use and the dimension to reduce to - we used various combination of those.

For a more detailed description see section 5.10.2.

2.4 Models

We use two models:

1. DETR - an object detection model which is composed of a convolutional neural network(CNN) backbone and transformer(encoder and decoder) [3].

2. Faster-RCNN - an object detection model which uses Region Proposal Network(RPN), a way to share features with the detection and the full-image convolutional features. an RPN is a fully- CNN network that both predicts object bounds and scores for each position, RPN receives an image as input and outputs a set of proposals, each proposal includes a rectangle, marked by coordinates and a corresponding score. The main improvement in Faster RCNN is weight sharing from the region and score proposal, both nets share a common set of convolutional layers, thus they benefit from the acquired knowledge of each instead of learning it each on their own.

2.5 Datasets

1. PASCAL VOC [4] - an object detection data set which contains 17,125 images and 20 classes.

2. KITTI [5] - an object detection data set which contains 7,486 images and 8 classes.

3 Results

3.1 Active Learning by Class Hardness - DETR

For the experiments in this part, we used an object detection model - “DETR” which was partially pre-trained - the CNN backbone and encoder were pre-trained on the COCO dataset, and the decoder of the transformer was trained from scratch. To evaluate the performance we use Mean Average Precision(mAP), a popular objection detection metric. We trained DETR on 2 datasets - PascalVOC and KITTI.

3.1.1 AL experiments with DETR on PascalVOC

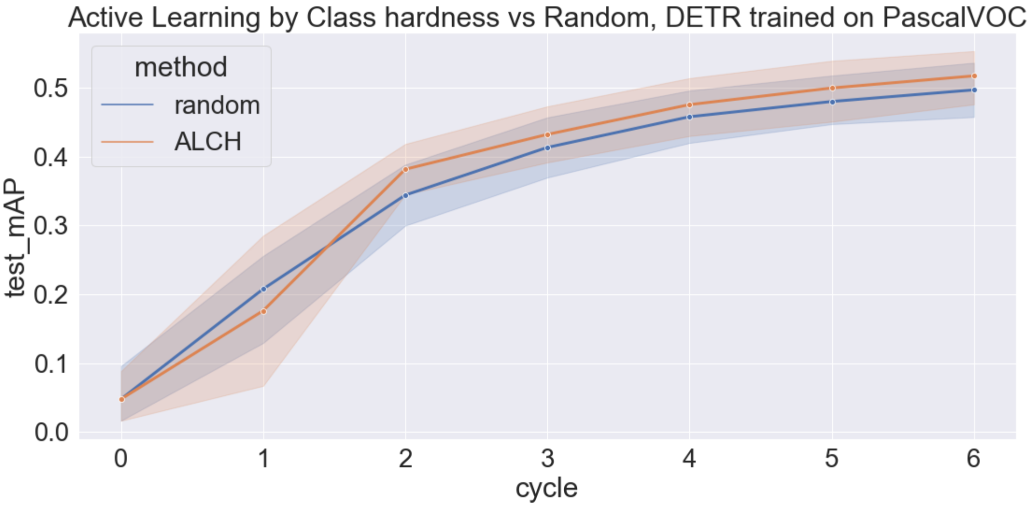

Figure 3: Active Learning by Class Hardness vs Random, on PascalVOC. The line for each method corresponds to the mean mAP on the test set. Each method had 4 independent runs. . A 95% bootstrap confidence interval is shown. In each cycle, 1000 images are added to the labeled set, and the initial training set consisted of 1000 images.

3.1.1.1 ALCH vs Random As can be seen in figure 3, just by observing the means, it seems as if ALCH is superior after the first cycle, but there is alot of variation which can be seen by the confidence intervals, so we can't say that ALCH is superior to random sampling according to these experiments.

In order to compare the performance with a statistical test, we perform a permutation test on the regression coefficients between the regression coefficients of ALCH and random, as elaborated in section 5.8.3.

We got that the p-value is 0.3429, so as was observed from the graph, we can't say that ALCH performance is significantly better then random.

3.1.1.2 Varying the Validation Set In addition to most of the experiments with a validation set of size 500, we wanted to vary the validation set size in order to see if changing it effects the performance of ALCH, and in particular, how estimating the class hardness weights directly from the test set - using the test set as the validation set, effects the results. The results are shown in figure figure 4:

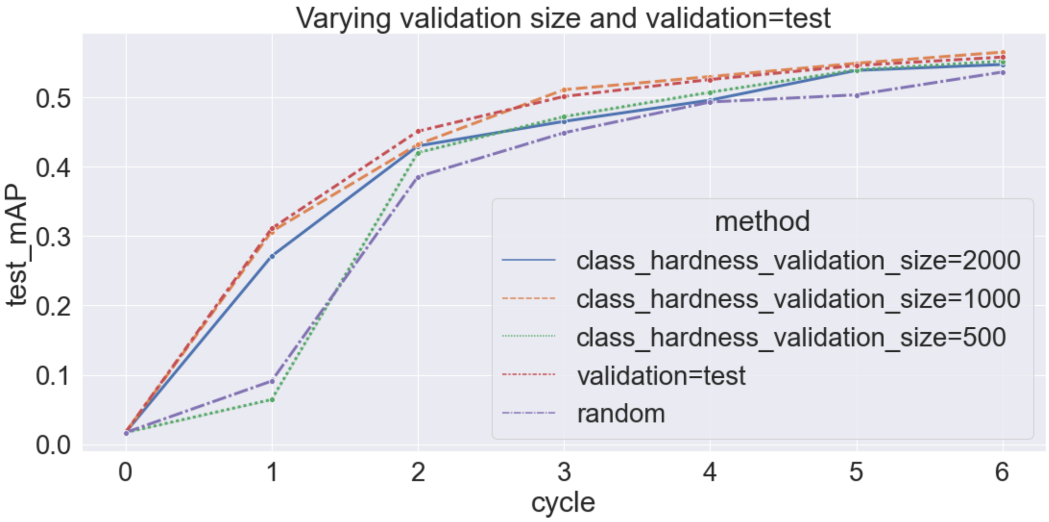

Figure 4: Active Learning by Class Hardness vs Random, on PascalVOC, in each cycle, 1000 images were labeled, and the initial training dataset size was 1000.

We can notice that the best performance is for ALCH with validation size=1000 and when validation=test, that is - when estimating class hardness weights directly from the test set. In addition, ALCH with validation size of 500 and random perform worse in this experiment. Each line in figure figure 4 corresponds to only one run, where all of the runs have the same initialization.

3.1.2 AL experiments with DETR on KITTI

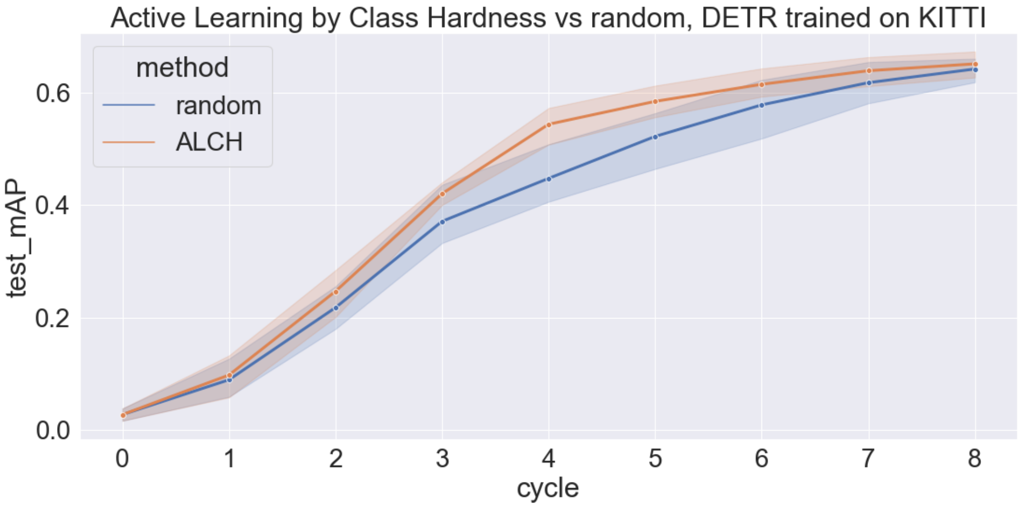

Figure 5: Active Learning by Class Hardness vs Random, on KITTI .The line for each method corresponds to the mean mAP on the test set. Each method had 4 independent runs. A 95% bootstrap confidence interval is shown. In each cycle, 500 images are added to the labeled set, and the initial training set consisted of 500 images.

As can be seen in figure 5, ALCH seems to perform better then random at cycles 3-6, but the confidence intervals are quite wide and we can't draw sharp conclusion.

In order to compare the performance with a statistical test, we perform a permutation test on the regression coefficients between the regression coefficients of ALCH and random, as elaborated in section 5.8.3.

We got that the p-value is 0.5571, so as was observed from the graph, we can't say that ALCH performance is significantly better then random.

3.1.3 AL experiments with class hardness , Faster-RCNN , Pascal-VOC

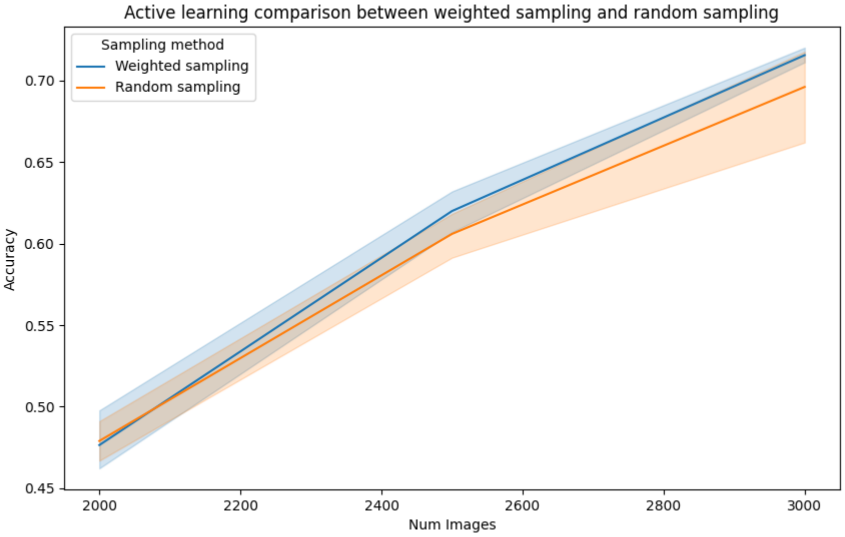

Figure 6: Each graph is composed of 4 different runs, i.e., 4 cycles were performed for each variation of the active learning, the mean accuracy is shown surrounded by a 95% confidence interval , in each cycle 1500 images were labeled and 500 were chosen.

3.1.3.1 Class Hardness vs Random In the weighted variation, the variance is significantly lower, which further supports the effectiveness of active learning with regard to object detection, the mean of the weighted sampling is also higher, in other words, the accuracy graph of weighted sampling dominates the random sampling with regard to the mean, and the variance is significantly lower.

Thus the results seem to be intuitive to the success of active learning, using a sample method that caters to the model needs yields not only a higher mean but also a lower variance while using the same amount of data and proves itself as a valid method to exchange resources from data collection to computation.

3.1.3.2 The affect of convergence on the active learning process (Class hardness) How does convergence affect the active learning process? We have been told that it is always preferred to reach convergence before continuing to the next cycle, we preferred to check that assumption by our self to understand whether she's also valid for object detection, we hypothesized that the convergence on the new samples will aggregate after enough cycles, thus we changed the number of epochs to 3 from 5 while increasing the number of images added per cycle to 1000 from 500, thus giving the model more data to train on but reducing the number of epochs, i.e. giving the model less time to refine his weights but increasing the amount of information he encounters.

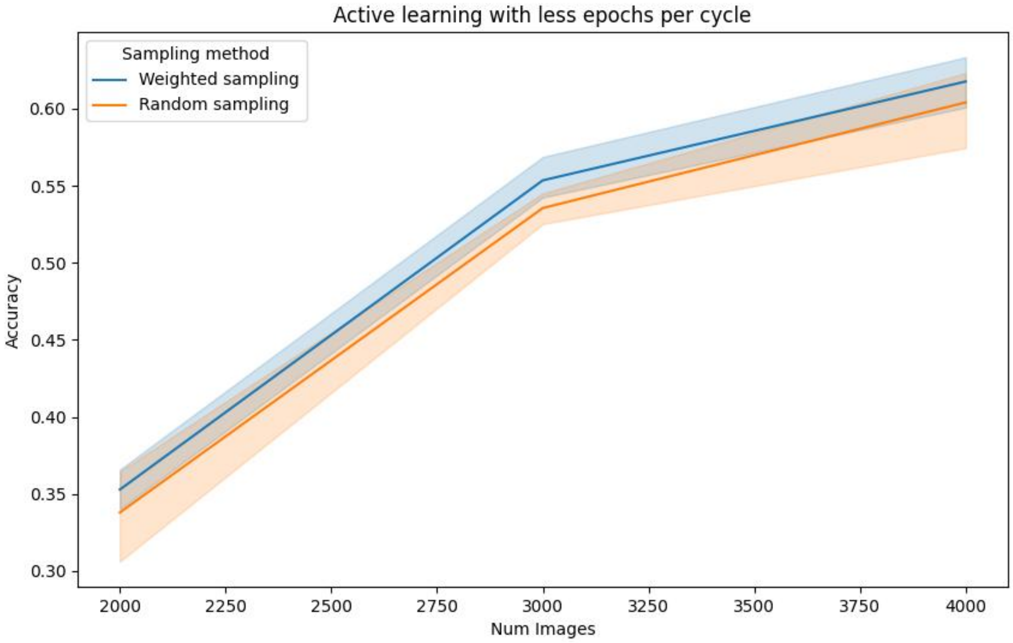

Figure 7: Each graph is composed of 4 different runs, i.e., 4 cycles were performed for each variation of the active learning, the mean accuracy is shown surrounded by a 95% confidence interval. In each cycle 2000 images were labeled and 1000 were chosen.

3.1.4 Active learning by properties of bounding boxes

In the section, we focus on the bounding box properties, mainly the area and their amount.

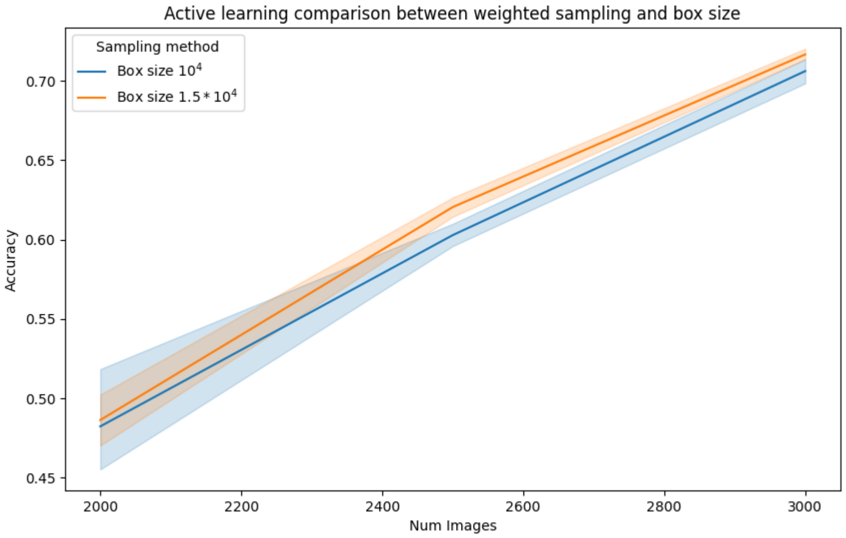

Figure 8: Each graph is composed of 4 different runs, i.e., 4 cycles were performed for each variation of the active learning, the mean accuracy is shown surrounded by a 95% confidence interval. in each cycle 1500 images were labeled and 500 were chosen.

3.1.4.1 Comparing hyper parameters for box area active learning Clear domination of the bigger box size over the smaller is seen as we hypothesized, we can observe that not only is the accuracy mean greater, the confidence intervals for 2500 and 3000 are distinct, the lower bound of the bigger size images doesn't intersect with the upper bound of the smaller, which indicate not only a dominate of the graph but also a domination of the confidence intervals. Thus the result support hypothesizing and the model has an easier time learning when the predicted objects bounding boxes are larger, in other words, the more area that is covered the more weights are affected, and the performance of the model increases as a result and we thus chose the higher threshold as the final one.

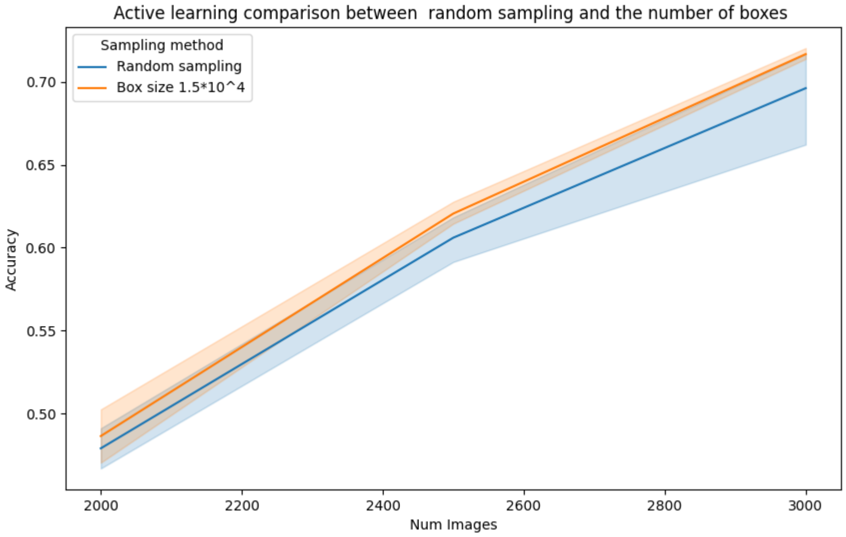

Figure 9: Each graph is composed of 4 different runs, i.e., 4 cycles were performed for each variation of the active learning, the mean accuracy is shown surrounded by a 95% confidence interval. in each cycle 1500 images were labeled and 500 were chosen.

3.1.4.2 Comparing box area active learning with random sampling Comparing the effeteness of active learning by the size of the box to random sampling clearly indicates domination with regard to the mean accuracy and the confidence intervals while they intersect to a high degree in the first cycle, in the second cycle their intersection is limited, thus when comparing the effectiveness of active learning by box size to random sampling we can see a clear advantage to the current active learning method.

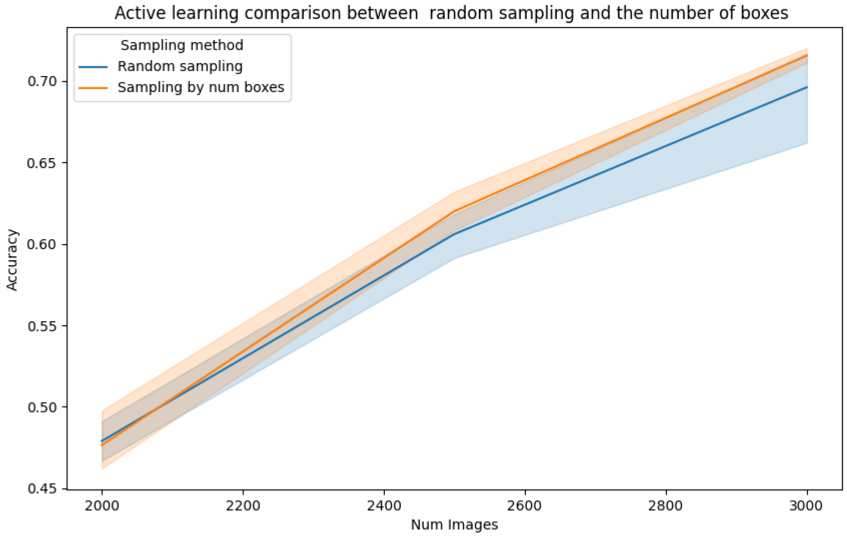

Figure 10: Each graph is composed of 4 different runs, i.e., 4 cycles were performed for each variation of the active learning, the mean accuracy is shown surrounded by a 95% confidence interval. in each cycle 1500 images were labeled and 500 were chosen.

3.1.4.3 Comparing amount of predicted boxes with random sampling the results indicate a clear advantage for sampling by a number of boxes while the random confidence interval isn't dominated by the current active learning method, the distinction between the average is clear.

In conclusion, active learning by the number of boxes proved itself as a valid active learning method.

3.1.5 Comparing between different active learning methods

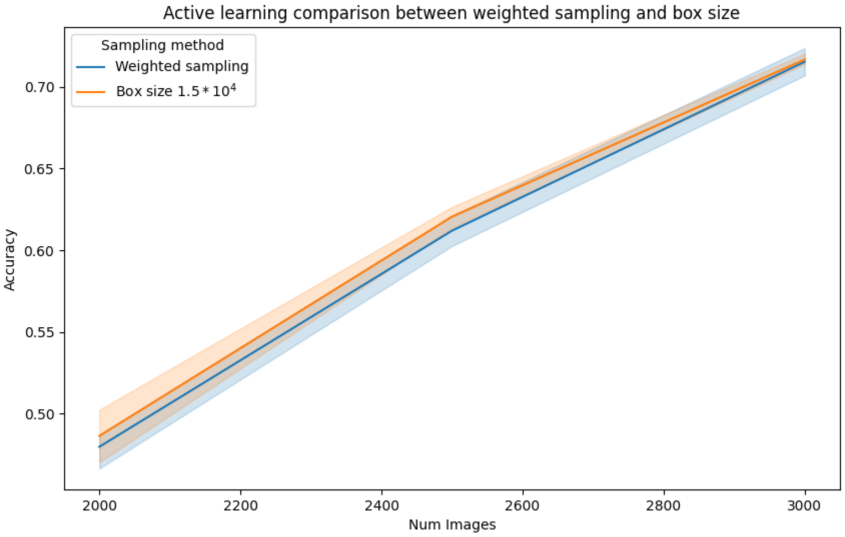

Figure 11: Each graph is composed of 4 different runs, i.e., 4 cycles were performed for each variation of the active learning, the mean accuracy is shown surrounded by a 95% confidence interval. in each cycle 1500 images were labeled and 500 were chosen.

3.1.5.1 Weighted sampling vs Box size An active learning comparison between two different methods, weighted sampling, and sampling by box size, at first we can see a small difference in the last cycle since sampling by weights require the model to find both the area in the image and classify it correctly, while sampling by the area size requires the model only to find a larger enough area without classifying the object thus the latter hypothesis require less training for the model in order to produce a successful query, this hypothesis would also explain the big difference in the second cycle, predictions that rely on bounding boxes doesn't require classifying the object itself, thus needs less accuracy in order to produce a correct query for the oracle, based on the same logic we could attempt to explain why box size sampling confidence intervals are smaller.

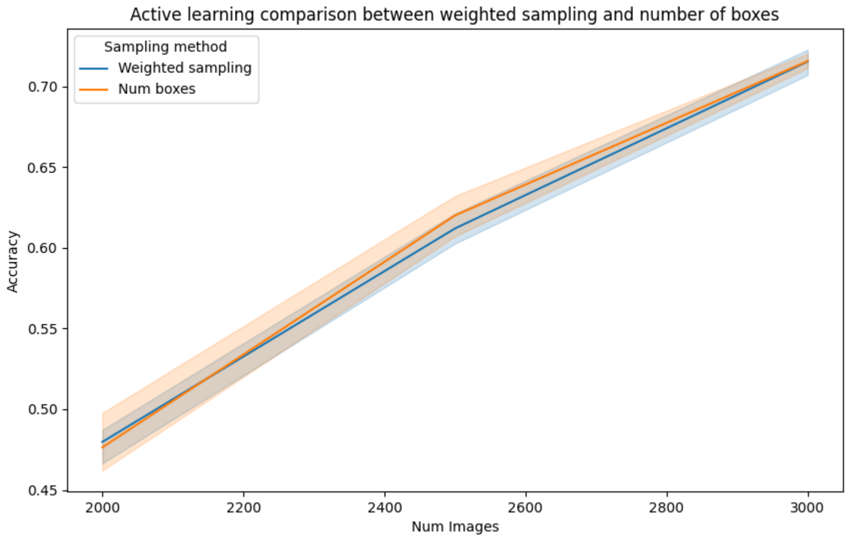

Figure 12: Each graph is composed of 4 different runs, i.e., 4 cycles were performed for each variation of the active learning, the mean accuracy is shown surrounded by a 95% confidence interval. in each cycle 1500 images were labeled and 500 were chosen.

3.1.5.2 Weighted sampling vs amount of boxes A comparison of sampling by weights and number of boxes, the results are similar to before and we believe they appear as such for the same reasons, the number of boxes predicted with high enough confidence relays only on the question of whether an object exists in a certain area on in the image or not without the need to classify him as well, therefore the difference in the second cycle is explained by the capability of the more simple hypothesis to produce more accurate results early on, but at the last cycle, the model has been trained enough to reach the same amount of accuracy.

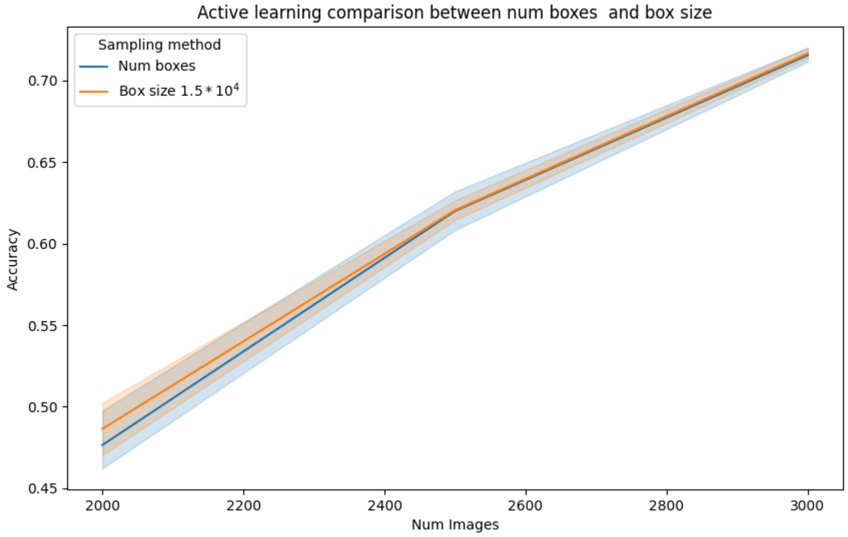

Figure 13: Each graph is composed of 4 different runs, i.e., 4 cycles were performed for each variation of the active learning, the mean accuracy is shown surrounded by a 95% confidence interval. in each cycle 1500 images were labeled and 500 were chosen.

3.1.5.3 Amount of boxes vs average boxes area Comparing the results of the number of boxes predicted and the average surrounded area, both active learning methods don't depend on classification, rather only on properties of the bounding box, thus both improve rapidly from the first cycle and achieve almost the same accuracy mean. The main difference lays in the confidence interval, we believe that the prediction based on the area size remains more consistent because a large enough area means almost serves as a guarantee for pixels with information to extract, but the number of objects predicted could result in a lot of object with a small area, i.e. in theory an image with 10 objects where each object is of size 50*50 would include far less information (pixel-wise) then an image with 3 objects with a mean area 15,000, thus the model ability to learn is less efficient since the same area would be scanned less time through the CNN forward run, more area means more weights would be engaged. Therefore while the number of boxes is a sufficient way to perform active learning his confidence intervals are wider than the area method because a larger area of bounding boxes almost guarantees more information in the image.

3.2 Post-Hoc analysis for Active Learning Results

As a measure of improvement on the test set between the cycles, we use the difference in the loss on the test, which is a weighted loss of classification and bounding box related losses.

We used control variables as there are various possible confounders which effect the model performance. We elaborate more on this process in section 5.10.1.

In order to evaluate the hypotheses, we used multiple linear regression where the dependent variable is the difference in the test loss between the cycles. For testing the significance of coefficients we use $\alpha = 0.05/9 = 0.005$, as we use Bonferroni correction for a total of 9 hypotheses(3 simple hypotheses, and 6 sub-hypotheses of the "Distance to the Closest Centroid Hypothesis").

3.2.1 Simple Hypotheses Results

We performed statistical analysis for the simple hypotheses described in 2.3.1.1. In order to asses the impact of each quantity(related to a specfic hypothesis) on the improvement in performance. Here are the results:

Table 1: Results of Multiple Linear Regression on Simple Hypotheses

	coef	**std err**	**t**	***P*-value**	**[0.025**	**0.975]**
const	-1.2341	0.042	-29.579	0.000	-1.317	-1.152
cycle(control)	-0.3553	0.119	-2.979	0.003	-0.591	-0.119
previous test loss(control)	-1.3283	0.116	-11.467	0.000	-1.557	-1.099
mean # of bounding boxes	-0.1259	0.125	-1.004	0.317	-0.374	0.122
mean area of bounding boxes	-0.1704	0.064	-2.647	0.009	-0.298	-0.043
mean IOU	0.1404	0.114	1.235	0.219	-0.084	0.365

As can be seen in table 1, the explanatory variables "mean number of bounding boxes" and "mean IOU" have a high p -value $>> \alpha$, which means that the hypothesis corresponding to these explanatory variables have **no evidence**. On the other hand, Mean area of bounding boxes has a smaller p-value$= 0.009$, but its greater then $\alpha = 0.005$. So there is only **weak evidence** for the "Mean area of bounding boxes" hypothesis.

3.2.2 Distance to the Closest Centroid Hypothesis

Similarly to section 3.2.1, we test the hypothesis with a multiple linear regression model.

When creating the data-set for each possible combination of the dimension and number of clusters, due to high multicollinearity we removed some of the configurations as was stated in 5.10.2.1.

So here test 6 sub-hypotheses, for various combinations of the dimension to reduce to and the number of clusters in KMeans. We got the following results:

Table 2: Results of Multiple Linear Regression on Simple Hypotheses

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
const	-1.2341	0.040	-30.482	0.000	-1.314	-1.154
cycle	-0.3356	0.123	-2.724	0.007	-0.579	-0.092
prev_test_loss	-1.2740	0.115	-11.100	0.000	-1.501	-1.047
dimension=2_# clusters=5	0.0529	0.060	0.877	0.382	-0.066	0.172
dimension=2_# clusters=25	-0.0184	0.052	-0.351	0.726	-0.122	0.085
dimension=2_# clusters=100	-0.0662	0.067	-0.995	0.322	-0.198	0.065
dimension=2_# clusters=500	-0.1783	0.064	-2.785	0.006	-0.305	-0.052
dimension=2_# clusters=1000	0.1876	0.055	3.403	0.001	0.079	0.297
dimension=512_# clusters=1000	0.0191	0.056	0.339	0.735	-0.092	0.130

As can be seen in table 2, the lowest p-value(for non-control variables) is for dimension=2, number of clusters=1000, a p-value of $0.001 < \alpha$, so this coefficient is significant, and this hypothesis has strong evidence,

It is important to notice that this most significant configuration(dimension=2, number of clusters=1000) has a positive coefficient, as opposed to the the subsequent lowest p-value configuration which has a negative coefficient. The meaning of this is elaborated upon in section 4.2.

4 Discussion

4.1 Active Learning by Class Hardness

In this part, we wanted to check if the active learning strategy of sampling by estimated class hardness(ALCH) is a good AL strategy. We tested this by comparing the performance between ALCH and random sampling. From the results that we got, we can't say that ALCH is better then random in a statistically significant way, according to the statistical tests we performed. That said - graphically, it does seem that ALCH is better, as the curves for ALCH are mostly above the random sampling curves in the various models and datasets experiments. This suggests that performing more experiments might clarify this, as explained in the potential pitfalls which follows.

There are potential pitfalls in the research of this part:

1. The models were pre-trained(on a different dataset), therefore the learning of the most basic representations of the neural network was probably not done when training by active learning in the experiments, and that most of the training while performing AL was fine-tuning. Therefore, it is likely that training with AL from scratch would have showed very different results. Initially we attempted to train from scratch, but the training time was too long.

2. We didn't have the resources to perform enough active learning experiments with different initializations, which is a possible reason for the non-significant p-values that

we got when comparing ALCH to random. When considering the results graphically, it does seem that ALCH is better than random, and so perhaps by performing more experiments statistical significance could have been achieved.

3. Hyper-parameters were chosen either by us (Confidence threshold in classification for ALCH) or used as default (learning rate, momentum, optimizer), hyper parameters proved in the past to be tricky and could either make or break a project, thus we're uncertain whether our choices were optimal. This can be mitigated by performing automatic hyper-parameter optimization, but it usually requires more computational resources. It should be noted that the non-active learning hyper-parameters(such as learning rate, optimizer, etc) we chose worked for the original architecture creators, so it is likely that the hyper-parameters are at least decent, while they may not be optimal.

There were 2 questions which were raised during research of ALCH:

1. How does the size of the validation set, from which the weights of class hardness are derived from, effects the performance of ALCH. We would expect that a larger validation set will be better for estimating the true class hardness.

2. How will the performance of ALCH be effected if the class weights are estimated directly from the test set(not from a validation set)?

As can be seen in 3.1.1.2 it seems that when estimating the class weights directly from the test set, and when the validation set size is bigger then 500, that is when it is 1000 or 2000, the performance of ALCH improves. This suggests that reducing uncertainty about the hard classes improves the performance of the method, and a future research direction might be to estimate the hard classes in a different way than we did, with a lower uncertainty about the hard classes.

Some additional future research directions might be performing more ALCH experiments with more compute, comparing ALCH to other AL methods besides random sampling, and attempting AL by class hardness for other tasks besides object detection, such as image segmentation

4.2 Post-Hoc Analysis Discussion

The goal of the Post-Hoc Analysis was to formulate various hypotheses about properties which might be beneficial for AL for object detection, and analyze if they were correlated with improvement over cycles in the Active Learning experiments. This was done by analyzing the coefficients of the various properties in a multiple regression model, and their associated p-values. The improvement can be measured by the difference in the test loss between 2 subsequent cycles. Therefore - the lower the test loss in a specific cycle, the better the improvement of the model in this cycle. Therefore - a positive coefficient of some explanatory variable implies that performance might be better when the values of that variable is low, and vice versa.

As was noted in section 3.2, The following hypotheses had relatively higher evidence:

1. Mean area of bounding boxes per cycle - the results suggest that labeling images in AL where the mean area of the bounding boxes is large(as predicted by the model), might be a good AL strategy, because as seen in table 1 the coefficient of “mean area of bounding boxes” is negative. This might also suggest that the model has more difficulty with larger objects.

2. In “Distance to the Closest Centroid Hypothesis”, the following configuration for UMAP and KMeans++ had stronger evidence: dimensionality reduction to 2 dimensions, number of clusters=1000.

   As reported in table 2, besides being the lowest p-value in this multiple regression model, this configuration has a positive coefficient, as opposed to the subsequent lowest p-values configuration which has a negative coefficient. This is very important, as this general hypothesis of “Distance to the Closest Centroid Hypothesis” predicts that sampling images which are closer to a centroid should yield higher performance then sampling images which are more distant. Therefore, the positive sign of this lowest p-value coefficient is persistent with this prediction, so this observation supports the “Distance to the Closest Centroid Hypothesis” hypothesis. This suggests that direct AL experiments for researching this hypothesis might be worthwhile.

Notice that there are some weaknesses to our post-hoc analysis methodology:

1. Relatively small sample size - only 144 data points, where each data point corresponds to a single AL cycle, where the cycles were collected from a total of 22 different runs. A larger sample size might have yielded clearer conclusions, but a meaningfully larger sample size required a much larger time or computing power. In addition with a larger sample size we could have controlled confounders in a more variety of ways - such as stratification, and possibly analyze interactions between hypotheses.

2. To analyze the effect of each property on the performance we used Multiple Linear Regression, which is a common way to statistically analyze relationships between variables, and is capable of capturing linear relationships, but not non-linear relationships, so its modeling capability very limited. We do think that this technique is a good choice for its relative simplicity and the small sample size, and we handled multi-collinearity in the per-processing, but future research might benefit from other statistical techniques.

4.3 Discussion active learning without convergence

The result indicate that reducing the amount of epochs from 5 to 3 but adding data does not impact the active learning process and he maintains his domination with regard to the accuracy graphs , which raises the question about whether needless repetition is performed in active learning? If an image is added in the first cycle and we perform 10 cycles until we finish current instance and each cycle includes 5 epochs , then the same image will be forwarded through the network for around 50 times , in comparison to an image that was added toward the end (e.g. cycle 7) which would only be forwarded 10 times. Should images that were added in the same cycle received such impact on the model only based on the time they were added? Perhaps learning them in aggregation over the whole active learning instance while reducing the need to fully converge in each cycle would provide the same results.

4.4 Discussion active learning by bounding box properties

4.4.1 Box area

The following section discuses hyper parameters with regard to the area of the box prediction

4.4.1.1 To chose the threshold γ We had a major problem which consist of choosing the threshold, which bounding box fits? there is no intuitive explanation for it, why for example should a prediction with a big bounding box be "better" than one with a smaller? why should the active learning model prefer one size over the other? In a perfectly trained model there is no need really, Faster-RCNN, in the end, is CNN model, which means he hovers over the model using 3x3 filters and examines each pixel, thus any bounding box that is bigger than 3x3 should be examined and classified corrected. But since the model is not in fact trained to perfection, we expected that bounding boxes with more area would pass through more filters layers in the CNN, thus more weighted from more filters would be dedicated to deciphering the patterns in the current box. In other words, as the area of the objects enlarges the back propagation process would update more weights in the model, more weights being updated by an image that includes a high amount of information should result in a faster training period, or at least better than random to some extent.

To check our hypothesis we chose two thresholds$\gamma \in \{10^4, 1.5 * 10^4\}$, a size of 10^4 means an average of boxes of size $(100, 100)$ should be expected, and the latter results in an average area of $(125, 125)$. In other words, how does a 50% increase from a humble base size affect the training? We expected the bigger size to prevail over the smaller since the larger area of bounding boxes should also be easier to learn, and the results showed clear domination as we expected.

4.4.2 Number of predicted boxes

Previously we have seen a strong correlation between the number of predicted boxes and the accuracy, thus we used it as a possible active learning process for object detection, we hypothesis that as the model gets more confident in his prediction he also increases the amount of bounding box.

4.4.2.1 Discussion about the correlation As the model gets more confident in his prediction, more predictions would surpass λ , images with small patterns that indicate an object might have been misses due to the model performance in previous cycles but would have been found, we chose a dynamic threshold of$\gamma = 4 + c$, i.e. the number of prediction boxes must surpass 4 in the first cycle, 5 in the second and so on.

We chose γ baseline as 4 because earlier attempts with a higher threshold could not find enough images from the unlabeled set, we tried to set it as 7 and the model predicted less than 100 images with more than 7 bounding boxes, then we tried with 6 and the model predicted less than 300, thus we chose 4 as the start baseline to ensure throughout training across all instances 500 added images per cycle.

The main drawback of the current method, there is some implication to consider with the dynamic threshold, there are images with a specific type of information that the model may yearn for but they won't be accessible to him since they only include 5 ground truth object, our main hypothesis is that images with a lot of objects in them would prepare an object detection model better for an evaluation on the data set, the hardest parts of object detection are images with plenty of objects, thus we consider images with a lower amount of bounding boxes to be less difficult for the model, therefor they include less uncertainty in them and would update the weights to a lower degree then images with an abundance of objects. Also, more objects would mean a higher probability of occlusion, we estimate that an occluded object is harder to predict, and training for example only on images with a small amount of predicted boxes would hinder the model capacity to handle occlusions.

5 Materials and Methods

5.1 Preprocessing - DETR training

DETR's implementation in Pytorch by facebook was written with COCO in mind for training and evaluation. As was described in 2.4, the model was partially pre-trained on COCO, and therefore to evaluate the Class Hardness method, we wanted to train it on other datasets, which are in a format different from COCO. Object detection datasets contain annotations which contatin bounding box and category information in the images, which requires careful parsing when converting between different formats. For the PascalVOC dataset, whose annotations are in VOC format, this was done with the help of a script from github [7] for converting from VOC xml to COCO json format.

As for the KITTI dataset, we used fiftyone [8] for converting between the KITTI format to COCO json format.

5.2 Preprocessing - Faster-RCNN training

MXNet package includes the option to download a pretrained Faster-RCNN model on COCO , in the same fashion as DETR training we wanted to train it on another datasets to avoid bias , MXNet has a comfortable option that allows training over VOC while being pretrained on COCO , thus the preprocessing is encapsulated within MXNet.

5.3 Training DETR

DETR is a heavy model which requires a large computing power in order to train it in a reasonable time from scratch - Facebook trained it for 3 days on 8 V100 GPU cards, while we had access to less powerful GPUs on university computers and a single limited instance on the Google Cloud platform.

In order to be able to train it in a reasonable time we have performed the following:

1. We used a pre-trained CNN backbone and encoder which were trained on COCO. The decoder part of DETR, which is more directly responsible for the object detection(as opposed to feature extraction from the image), was trained from scratch.

2. We performed early stopping to train until convergence in each cycle with a patience parameter of 2(number of epochs without, or very mild improvement), in order to balance the trade off between training time and performance.

5.4 Training Faster-RCNN

5.4.1 Training Faster-RCNN by Class Hardness

The algorithm remained almost the same besides the following changes :

We trained the model for 20 epochs on the full pascal-VOC data set, then transformed the prediction accuracy of each class into a discrete distribution by using the following formula :

$$W_j = \frac{e^{-w_j}}{\sum_i^n e^{-w_i}}$$

We chose a negative power instead of positive to encourage low values to enlarge while the large value will shrink, an intuitive way to increase the likelihood of classes with low accuracy in our distribution.

The model takes into consideration the confidence when choosing which images to forward, images were chosen only if they surpassed a threshold that changed with every cycle (c denotes the current cycle):

$$\lambda_c = 0.3 + \frac{1}{11}c$$

As the model has been trained on more cycles, his confidence is more accurate, therefore as the training continues we focus on forwarding images that fit the discrete distribution and increase the chances to avoid sending to our oracle a miss classification of our model.

In each cycle a constant number of digits were sampled from the discrete distribution defined by the weights, then from the pool of the predicted images, we take exactly K_j images for class j, where K_j symbolizes the current amount of digits j in the current sampled from the discrete distribution.

An image was chosen to query if at least 1 of the class prediction of the trained model thus far included the digit and the current box prediction digit was above the λ

5.5 Active learning by Class Hardness (Faster RCNN)

Algorithm 1 Faster RCNN Active learning by Class Hardness

```
1. Sample V from pool and fix it as validation set
2. Sample $D_L$ from pool - the initial labeled data set , such that $D_L \cap V = \emptyset$
3. 3. While $|D_L| < |PULL|$(active learning loop)
    (a) Train the model on $D_L$
    (b) Obtain model prediction on the unlabeled set and chose only the bounding boxes
        with a confidence score $\lambda > c * \frac{1}{11} + 0.3$
    (c) Generate a sample from the weight distribution
    (d) For each digit class in the sample , find a box in an image that is classified as
        corresponding class , if the prediction is above $\lambda$ , append our initial data set $D_L$
        with the current image and remove the image from the unlabeled pool , in other
        words $D_L \leftarrow D_L \cup I$ , where $I$ is the current image.
```

5.6 Active learning by box area (Faster - RCNN)

Algorithm 2 Active Learning by box area

```
1. 1. Sample V from pool and fix it as validation set
2. 2. Sample $D_L$ from pool - the initial labeled data set , such that $D_L \cap V = \emptyset$
3. 3. While $|D_L| < |PULL|$ (active learning loop)
    (a) Train the model on $D_L$
    (b) Obtain model prediction on the unlabeled set and chose only the bounding boxes
        with $\lambda > c * \frac{1}{11} + 0.3$
    (c) Calculate the area of each box in the current prediction
    (d) Calculate the average bounding box on the current image , if the average bounding
        rectangle size is greater then $\gamma$ , append our initial data set $D_L$ with the current
        image and remove the image from the unlabeled pool , in other words $D_L \leftarrow D_L \cup I$
        where $I$ is the current image
```

5.7 Active learning by amount of boxes (Faster - RCNN)

Algorithm 3 Active learning by amount of boxes

```
1. 1. Sample V from pool and fix it as validation set
2. 2. Sample $D_L$ from pool - the initial labeled data set , such that $D_L \cap V = \emptyset$
3. 3. While $|D_L| < |PULL|$ (active learning loop)
    (a) Train the model on $D_L$
    (b) Obtain model prediction on the unlabeled set and chose only the bounding boxes with $\lambda > c * \frac{1}{11} + 0.3$
    (c) Extract the amount of boxes in the current image , if the amount of boxes predicted is greater then $\delta$ , append our initial data set $D_L$ with the current image and remove the image from the unlabeled pool , in other words $D_L \leftarrow D_L \cup I$ where$I$ is the current image.
```

5.8 Active Learning by Class Hardness(DETR)

In detail, the AL algorithm works as follows:

Algorithm 4 Active Learning by Class Hardness

```
    1. Sample $V$ from the pool - and fix it as a validation set.
    2. Sample $D_L$ from the pool - the initial labeled dataset,
    such that $D_L \cap V = \phi$
    3. While $|D_L|$<|PULL|(active learning loop):
        4. Train the model on $D_L$
        5. $\forall i \in [K]$ calculate the loss on the validation set $V$:

                $L\left(C_i\right) = \frac{1}{|C_i|} \sum_{x_{ij} \in C_i} \ell\left(x_{ij}\right)$

           notice that $x_{ij}$ is a bounding box in the i'th class, $\ell$ is a loss
           function per bounding box, and
           $|C_i|$ is the total instances of the i'th class in the
           validation set.
           6. $\forall i \in [K]$ calculate the weight for each class:

                $W(C_i) = \frac{L(C_i)}{\sum_j L(C_j)}$

           to obtain a weight vector:

                $\vec{w} = \left[W\left(C_1\right), ..., W\left(C_k\right)\right]$

           7. If using noisy weights: inject noise to $\vec{w}$.
           8. Run the model on all the images in the pool to obtain
           pseudo labels which will yield the *predicted* classes in the images.
           9. Construct a query $Q$ of $m$ images from the pool, such that for
           a particular class $j$, the query contains $W(C_j) \cdot$
           $m$ images which are predicted as class $C_j$.
           If not enough images were found, then sample the rest
           randomly to complete the query.
           10. Update the labeled dataset:$D_L \leftarrow D_L \cup Q$
           and remove the images in $Q$ from the pool.
```

Notice that the validation set is not used in this context for evaluating the performance of the Active Learning method itself, but is used as a way to estimate the hardness of each class. To estimate the performance of the Active Learning method itself we use a separate test set for all of the experiments. The method is evaluated on this test set after every AL cycle via the “Mean Average Precision”(mAP) metric, which is a very popular metric for evaluation of object detection models.

5.8.1 Noise Injection

In some of the runs we injected noise to the estimated class weights. This was done by multiplying the estimaed class weights by element wise by some number to get a new weight vector ,and sampling from the Dirichlet distribution with this multiplied vector as the parameter to get the new noisy weights class. More concretely - if $\vec{w}$ is a weight vector for the class weights, then the new noisy weight vector is $w_{noisy} \sim Dirichlet\left(\vec{w} \odot C\right)$, where $C \geq 1$, the smaller C, the more noisy are the weights.

In practice we didn't use the ALCH experiments with noisy weights in the analysis of ALCH vs Random due to reproducibility problems, but we did use them for the Post-Hoc analysis.

5.8.2 Active Learning parameters

The query size(number of images to label after each cycle) when training with DETR was:

1. 1000 images on PascalVOC

2. 500 images on KITTI.

5.8.3 Statistical test for comparing ALCH to Random performance

For each method(ALCH and Random) we want to estimate its performance trend(as expressed by the mean average precision on the test set), we can do this by fitting a linear regression model for each run performance data, where the explanatory variable is the cycle number, and the target is the test mean average precision. From this linear regression model, we can extract the slope, such that the better the method, the larger its slope coefficients, as a larger mAP corresponds to better performance. So for each method, we fit a linear model to its performance data and save the slopes, and conduct a permutation test(wilcox.test in R) between the list of ALCH runs slopes and Random runs slopes, where the null hypothesis is that ALCH slopes are not bigger then the slopes of the Random runs. We then report the p-values in the results(section §3).

5.9 Preprocessing - Post-Hoc Analysis

The data for the post-hoc analysis was saved during the experiments for Active Learning on PascalVOC. For each cycle in the AL, the following information was saved:

1. The performance on the test set after training in the AL cycle.

2. The following query information:

   (a) The image id's of the queried images in the AL cycle.

   (b) The model's prediction on the images which were queried in the AL cycle.

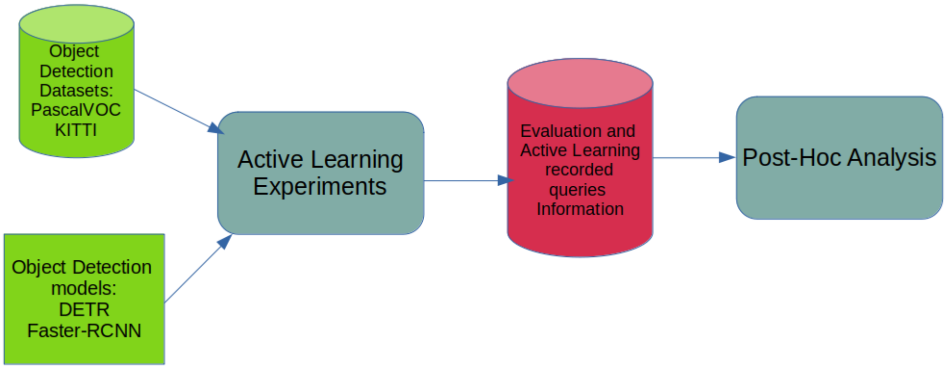
Object Detection Datasets: PascalVOC KITTI
Object Detection models: DETR Faster-RCNN
Active Learning Experiments
Evaluation and Active Learning recorded queries Information
Post-Hoc Analysis

Figure 14: Pipleline for data acquisition for the Post-Hoc Analysis

5.10 Post-Hoc Analysis Methods

5.10.1 Statistical Methodology

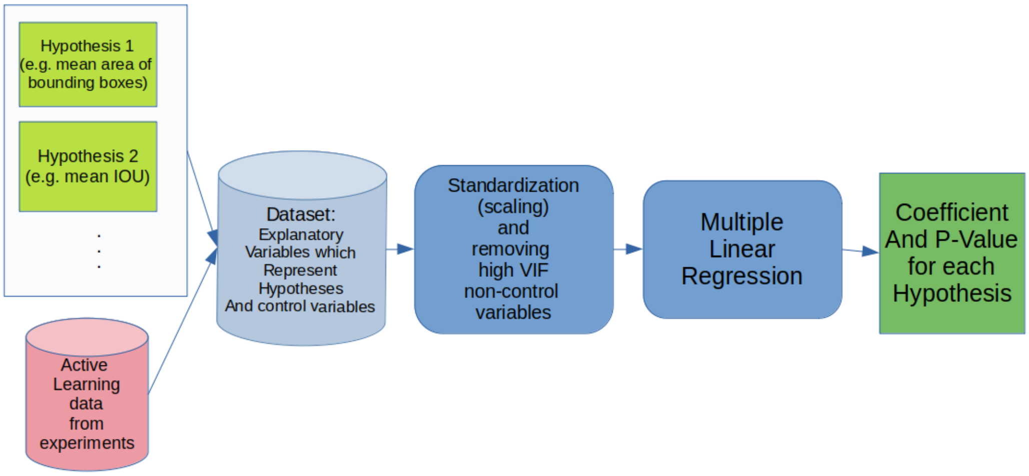
Hypothesis 1 (e.g. mean area of bounding boxes)
Hypothesis 2 (e.g. mean IOU)
.
.
.
Active Learning data from experiments
Dataset: Explanatory Variables which Represent Hypotheses And control variables
Standardization (scaling) and removing high VIF non-control variables
Multiple Linear Regression
Coefficient And P-Value for each Hypothesis

Figure 15: Statistical Methodology Pipeline

In order to test the hypotheses, and that their associated properties improved performance, it was not enough to observe the direct correlation between the explanatory variables(of the various hypotheses) and the improvement in performance, as there are possible confounders. For example, the mean number of bounding boxes, a property which we use to test the hypothesis that labeling images with a large number of bounding boxes(as the detector predicted) may improve performance, is correlated with the cycle number - as the cycles progress and the object detection model is trained more, it will tend to predict more bounding boxes, as can be seen in figure 16.

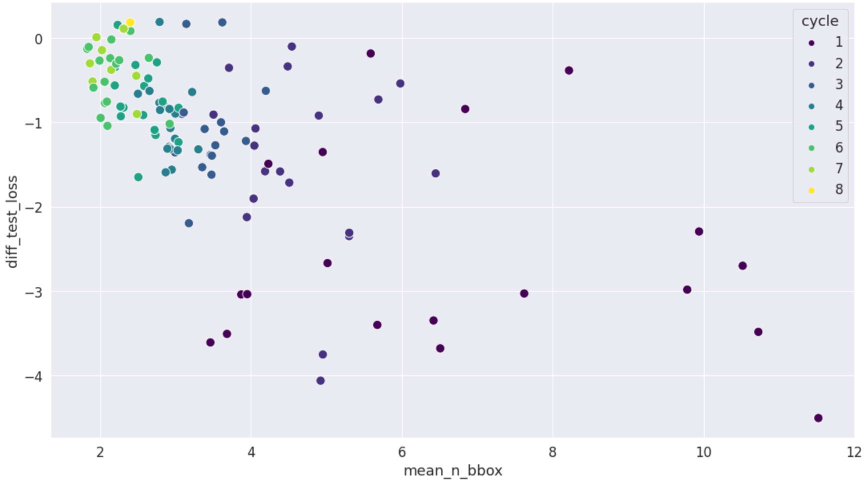

Figure 16: The difference in test loss vs the mean number of bounding boxes(as predicted by the detection model). Each point represents an AL cycle, which is colored by the cycle number. The mean number of bounding boxes is correlated with the test loss difference, but so is the cycle number, which is also correlated with the mean number of bounding boxes. Therefore the cycle should be included as a control variable.

Another variable which we controlled for is the previous test loss, as it might also effect the improvement - if the test loss is very low in some AL cycle, it is less likely to decrease in the next AL cycle.

In order to control for these confounding variables we first attempted stratification - grouping the data points by their cycle number, and testing the hypotheses individually in each group. We then discovered that this is problematic because of the small sample size, as the number data points in each group was too small.

Therefore, we used a multiple linear regression model and included in it the control variables, along with the explanatory variables of the various hypotheses. This setup makes sense, because in a such a model, we can interpret the coefficients individual effect on the response as the other variables are fixed, which allows us to control confounding variables.

We also removed explanatory non-control variables with a VIF>10 to reduce multicollinearity as elaborated in 5.10.2.1. Such high VIF variables were only configurations(which are also explanatory variables) of “Distance to the Closest Centroid Hypothesis”. For “Simple Hypotheses”(described in 2.3.1.1) , there weren’t any explanatory variables with VIF>10.

5.10.2 Distance to the Closest Centroid Hypothesis detailed description

The processing for analyzing the hypothesis was created as follows:

1. Extract the feature representation every image in the training set(of PascalVOC), by running pre-trained CNN - resnet18 on every image, without the fully connected layer.

2. Keeping the features as is, or performing non-linear dimensionality reduction with UMAP [9] to obtain feature vectors with a smaller dimension, in order to improve the clustering in the next step.

3. Running KMeans++ on the on the data from the previous step.
4. Obtaining the distance of every feature vector of the training set from its closest centroid.
5. For every AL cycle, and every image which was queried in the AL cycle, obtain the distance of it's feature vector from it's closest centroid(already calculated in step 2). Calculate the average of these distances per cycle.
6. Use the above calculated average distances per cycle as a predictor in model to test the hypothesis that querying images in AL by their distance to the closest center improves performance.

If the hypothesis is true, we would expect that in the cycles were the mean distance is smaller, the improvement would be larger.

This process is illustrated in the following diagram:

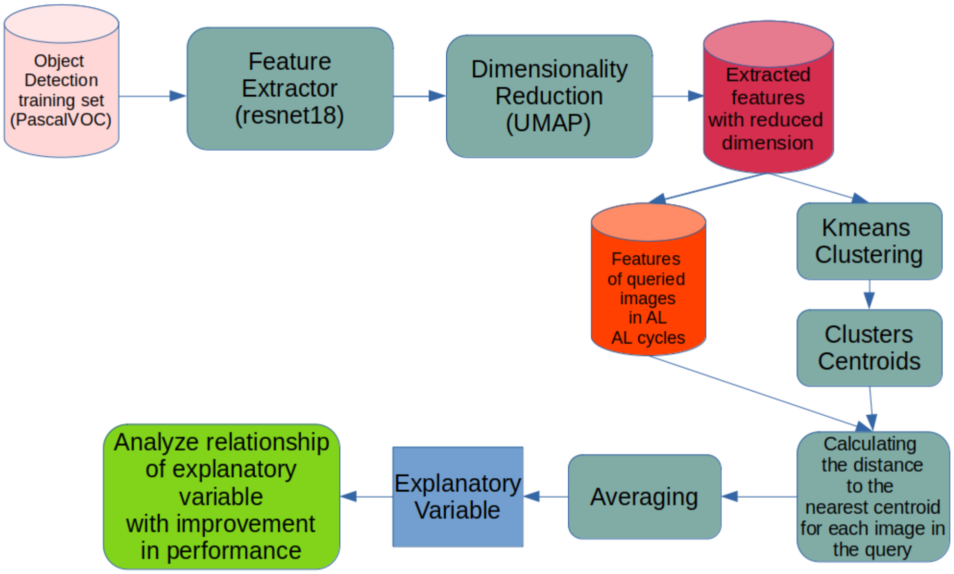
Object Detection training set (PascalVOC)
Feature Extractor (resnet18)
Dimensionality Reduction (UMAP)
Extracted features with reduced dimension
Features of queried images in AL AL cycles
Kmeans Clustering
Clusters Centroids
Calculating the distance to the nearest centroid for each image in the query
Averaging
Explanatory Variable
Analyze relationship of explanatory variable with improvement in performance

Figure 17: Creating the data for testing the "Distance to the closest Centroid" Hypothesis

5.10.2.1 Removing configurations There are various possibilities as to the dimension we should reduce to and the number of clusters to use, so at first we wanted to use all of the possible combinations of the following:

- number of clusters: $\{5, 25, 100, 500, 1000\}$
- Dimensions to reduce to: $\{2, 25, 150, 512\}$. Where 512 is the original dimension of the features vectors which resnet18 outputs, so no actual reduction to 512.

Which got us a total of 20 "sub-hypothesis" in relation to the "Distance to the Closest Centroid Hypothesis".

That said, before testing the 20 sub-hypotheses we checked for the VIF(Variance Inflation Factor) of the explanatory variables corresponding to these sub-hypotheses, and removed all explanatory variables whose VIF>10 in order to reduce multicollinearity, which left us with only 6 out the 20 sub-hypotheses, corresponding to these configurations:

1. dimension=2, # clusters=5
2. dimension=2, # clusters=25
3. dimension=2, # clusters=100
4. dimension=2, # clusters=500
5. dimension=2, # clusters=1000
6. dimension=512, # clusters=1000

Notice that most of the dimensions are 2, which might make sense considering that KMeans is an euclidean distance based clustering algorithm, which might work better for lower dimensions.

We tested these 6 sub-hypotheses(configurations) with multiple linear regression as explained in 5.10.1.

Division of work

- David Pitts: Active Learning with Faster-RCNN - AL by Class Hardness, AL by box area, AL by amount of boxes.
- Daniel Segal: Active Learning by Class Hardness with DETR, Post-Hoc analysis for possible AL patterns.

References

[1] Settles, B. (2009). Active Learning Literature Survey (Computer Sciences Technical Report No. 1648). University of Wisconsin–Madison.

[2] Donggeun Yoo and In So Kweon. Learning loss for active learning. In CVPR, pages 93–102, 2019.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In ECCV, 2020

[4] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2):303–338, 2010

[5] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

[6] Xin Wang, Yudong Chen, and Wenwu Zhu. 2021. A survey on curriculum learning. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)

[7] yukkyo (2021). voc2coco. GitHub. Note: https://github.com/yukkyo/voc2coco.

[8] Moore, B., & Corso, J. (2020). FiftyOne. GitHub. Note: https://github.com/voxel51/fiftyone.

[9] McInnes, L., Healy, J., & Melville, J. (2018). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. ArXiv e-prints.