# Qualitative Data Selection with Active Learning

Linnea Allander

Torben Nordtorp

LUND UNIVERSITY

Department of Automatic Control

MSc Thesis
TFRT-6198
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden


Printed in Sweden by Tryckeriet i E-huset
Lund 2023

# Abstract

In this work, an active learning pipeline is presented that allows for comparative tests between mainly three different types of data scoring methods: uncertainty selection, diversity selection, and perfect loss selection. Tests and parameter sweeps indicate that hyperparameters like reshuffling and early stopping are required to ensure fair comparisons. We then apply our pipeline to test some naive scoring methods on two Computer Vision problems: Image Detection, and Object Detection. We conclude that comparing these scoring methods against random selection gives minor evaluation loss improvements on our datasets in the right circumstances, and discuss other aspects of data quality along the way. The repository for our pipeline can be found on GitHub at `https://github.com/voxlz/Qualitative-Data-Selection-with-Active-Learning`.

# Acknowledgements

We would like to thank Joel Sjöbom and Emma Person, our primary supervisors, for weekly feedback and guidance with our project. We want to thank our work department for providing all the computing power needed to run our tests, which without this report would not have been possible. Additional thanks to Liu Haochen and Johan Sternby for tolerating our questions about internal pipelines and Machine Learning concepts. And to the rest of the team, for their help and support. Finally, we also want to thank our LTH supervisor Pontus Giselsson and examiner Karl-Erik Årzen, for helping us write and plan this thesis work.

# Contents

# 1

# Introduction

## 1.1 Background

Most machine learning models are heavily restricted by what data they have access to. While the number of public databases has grown, most organizations struggle to find or create specialized datasets that fit their own needs. Labeling data can be time-consuming and expensive, and finding data that is compliant with laws like the GDPR can further complicate the issue. Methods to intelligently sample which data to be annotated could prove very helpful. Further, this could reduce the amount of data needed to achieve acceptable results.

Another way to increase database diversity is through synthetic data. The continual advancement of processing power and photo-realism of commercial game engines have over time resulted in easier access to synthetic image data generation. While the idea of training models with synthetic data is not new [Bochinski et al., 2016a], the cost and time required for image generation have decreased dramatically. However, when the generation of photo-realistic images becomes trivial, the priority must shift from quantity to quality. Training on an abundance of synthetic data could result in worse generalization performance, and domain transfer issues, and take unnecessary time and power to train.

Researching data selection and data prioritization would address both these issues while having the capability to further improve model performance.

## 1.2 Research Goals

Active learning is the field of research that develops methods specifically designed to prioritize and select beneficial data for training. Traditionally, data prioritization methods have been used to maximize model performance when data is sparse and labeling expensive. Human annotators are expensive and take quite long to annotate complex data. Active learning is therefore used on unlabeled data to prioritize what data to label next, to avoid costly annotation on images that do not improve the model. In our case, however, we are under the assumption that we know the

ground truth of the dataset, i.e., the data labels required for training. We assume this because we have focused on how active learning can be used on previously labeled data, like for example synthetic data. We will re-evaluate some basic active learning methods on modern deep Convolutional Neural Network (CNN) architectures given these assumptions. Through a qualitative selection of data, we hope to maximize the performance gains of our models. By further applying these different methods to two different computer vision problems, image classification, and object detection, we strive to generalize our findings and document efficient and sound method comparison strategies.

An additional research topic that will be discussed is the assumption that more data is always better. If not the case, could active learning methods be used retroactively on datasets? In addition, the ratio between real and synthetic data in a given dataset [Jeon et al., 2022] and what mixing strategy [Nowruzi et al., 2019b] is used to mix real and synthetic data will be discussed.

Much can be said about how to improve model architecture to accommodate fewer data or synthetic data, but we will limit our focus to solely look at data selection through active learning.

# 2

# Related Work

Both active learning and synthetic data generation are ways to work around the limitations of having a small labeled dataset. Active learning lessens the need for large quantities of data through smart data selection, while synthetic data generation helps extend the labeled dataset cheaply and quickly. Much research has been done on both topics, but we will focus on articles within the area of computer vision.

## 2.1 Scoring Methods

One popular active learning method is sampling new data based on a confidence score, also called uncertainty sampling [Lewis and Gale, 1994]. In the paper that originally introduced the concept, they looked at the model output vector, i.e., a vector containing the model confidence score for each class, to gauge uncertainty. This method of uncertainty calculation is called uncertainty sampling. In this thesis, we mostly look at the methods referenced in Human in the Loop book [Monarch, 2021]. Another method, called Learn-loss prediction, is to prioritize new data based on the predicted loss, inspired by [Yoo and Kweon, 2019]. These and similar methods will be the focus of this thesis.

Other methods of determining confidence scores also exist. Bayesian Machine learning models directly produce confidence scores through probability prediction [Kwon et al., 2020]. Further separation into epistemic and aleatoric confidences, i.e. systematic and statistic confidences, has also been attempted in object detection contexts [Choi et al., 2021].

Confidence score based on consistency in detection has also been researched. These work by looking at how consistently the model predicts classes or objects by giving it two quite similar images, or by looking at how much it corrects itself throughout the detection process. Research has been made on consistency scoring through input augmentation [Yu et al., 2022] and by looking at for example bounding box differences in two-step detection models [Kao et al., 2018].

## 2.2 Synthetic Data

Synthetic data is also fairly researched. In 2016 a fully generated synthetic dataset was evaluated to have competitive results compared to real data [Bochinski et al., 2016b]. They used an old game engine to generate synthetic data. There have also been attempts at extending datasets with synthetic images, like this paper from the surveillance sector [Simpson et al., 2022]. Evaluating if photo-realism, backgrounds, color, or texture are needed for good detection has also been researched, with surprising results. In another paper from 2019, they attempted to generate training images by layering hundreds of distracter objects, i.e. objects similar to the ones we want to detect, on top of each other. This resulted in better performance compared to using real data [Hinterstoisser et al., 2019]. In [Nikolenko, 2021] it is discussed how different models look at different cues, and how some articles have managed to train on synthetic images without color or textures.

Most research agrees upon the benefit of synthetic data. But mixing strategies, like the ratio of real to synthetic and which dataset should be tuned on the other, is also a topic of interest [Jeon et al., 2022]. Some research suggests that just mixing synthetic and real data might perform worse than training solely on synthetic and fine-tuning on real data [Nowruzi et al., 2019a]. Others have found that while mixing datasets, too much synthetic data might make it perform worse. In [Rajpura et al., 2017], going beyond a 10/90 real to synthetic split was unfavorable.

## 2.3 Semi-supervised Models

In addition to the above-mentioned research into active learning and synthetic data, other methods of dealing with costly or inaccessible label data have been proposed. Semi-supervised models try to train on their own prediction, avoiding the need for manual labeling altogether. Recent research includes using adversarial models to improve accuracy [Ma et al., 2023]. Attempts to combine active learning and semi-supervised learning have also been made [Sijin et al., 2023].

# 3

# Method

## 3.1 Data Scoring Methods

Throughout our research, we have evaluated a multitude of data selection methods against each other and random selection, which has been included to act as a baseline. This section will aim to explain these selection methods.

### 3.1.1 Uncertainty Scoring

The uncertainty selection methods try to gauge the model's uncertainty in the predictions it produces. These methods are commonly based on classification scores or in other words, how confident the model is that an object belongs to a certain class. In our work we have implemented the following methods as described in [Monarch, 2021]: least confidence seen in Equation 3.1, the margin of confidence in Equation 3.2, and entropy in Equation 3.3. These functions $\phi(x)$ use the probability function $P_\theta$ of the predicted class $y$, given the input $x$ (i.e. an image). The main difference between these functions is that least confidence looks at the most confident class prediction ($y^*$), and margin of confidence at the two most confident class predictions ($y_1^*$ and $y_2^*$) whilst entropy looks to all class predictions ($y$) in order to determine the uncertainty of input $x$. Also, note that $n$ denotes the number of classes. Selection is then done through descending sort and selecting the ones with the highest uncertainty score.

Another way of understanding these selection methods is shown in Figure 3.1. The black dot in the figure represents 3 different classes. The blue area around one of these classes symbolizes where the model is most certain that another example within this area would belong to the nearest class. As we move further away from one class towards another class, the color shift from blue to green to yellow and possibly to red. The model becomes less certain of which class the input should belong to. With these scoring methods, we aim for examples in the red area where the model is most uncertain about which class they belong to. Depending on the method, the red areas vary in size.

$$\phi_{LC}(x) = (1 - P_\theta(y^*|x)) \times \frac{n}{n-1} \tag{3.1}$$

$$\phi_{MC}(x) = 1 - (P_\theta(y_1^*|x) - P_\theta(y_2^*|x)) \tag{3.2}$$

$$\phi_{ENT}(x) = \frac{-\sum_y P_\theta(y|x) log_2 P_\theta(y|x)}{log_2(n)} \tag{3.3}$$

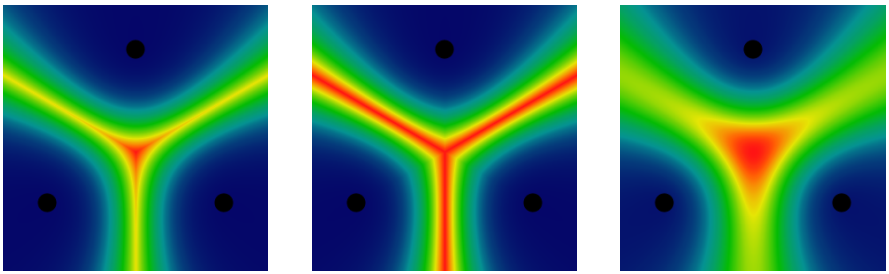

**Figure 3.1** The uncertainty selection methods. Least confidence (left), margin of confidence (middle), and entropy (right). Color visualizes the confidence around the classes (black), with blue being high confidence and red being low confidence.

### 3.1.2 Diversity Scoring

Diversity scoring is the process of increasing the label diversity, ensuring that all labels are sufficiently represented, in a given dataset. Ensuring that scoring is representative of the trained class labels could be beneficial. In our research, we have evaluated the effects of naive class diversity, where we ensure the scoring is even among classes while keeping the prioritization order from our uncertainty scoring. For example, given we have 2 classes, human and car, our diversity scoring would ensure there would never be more than 3 samples of one class given it selects 5 images. Diversity scoring can be used together with uncertainty scoring, but can also be used by itself. When used together in this thesis, it is applied after uncertainty scoring. There are other types of diversity scoring methods that we do not evaluate in this thesis, like dataset diversity (even distribution of data looking at dataset origin) and ensuring that the distribution has the same class appearance ratio as the training data.

### 3.1.3 Loss Scoring

As described in [Yoo and Kweon, 2019], learning loss is a task-agnostic selection method based on loss. In machine learning, the loss is defined as the difference taken of the prediction loss and the true label loss. The prediction loss and the true

label loss are calculated using a loss function, i.e., the categorical cross-entropy loss function. However, we might not know the true label. To avoid this we can through training a separate head (or model) predict what the produced loss would be on any new input, and rank them based on that. This is based on the assumption that high loss is correlated with the importance of given training data. Since we are also interested in working with synthetic data, we can then assume to know the true labels and thus sort based on perfect loss estimations. This method will be referred to as loss descending. In our case, we are using the same loss function(s) as the model to produce our loss predictions or perfect loss score.

## 3.2 Active Learning Pipeline

For this project, we wanted to simulate the real process of performing active learning on a model and training data. Conventionally you perform a full retrain of your model after each active learning iteration. Furthermore, it is common for modern CNNs to make use of a backbone, with pre-trained weights. To mimic this procedure, we also make use of "baseline models", that is to say, pre-trained weights applied to a model, and reset to this "baseline" after each loop. That way we do a full retrain on the newly selected data. With this in mind, a training pipeline was created. Please see the repository on GitHub for our active learning pipeline `https://github.com/voxlz/Qualitative-Data-Selection-with-Active-Learning`.

This pipeline is used to perform our experiments and evaluate which of the selection methods, least confidence, margin of confidence, entropy, loss descending, and random, provide the best model improvement. During each experiment, we run the pipeline multiple times for several selection methods while keeping all other parameters fixed. Only the training data then is altered as a result of the given method. For each selection method, we iteratively accumulate training data from the selection pool, reset the weights of the model, then tune and evaluate the model. A general overview can be seen in Figure 3.2 and a simplified pseudo-code over our pipeline can be found in Algorithm 1.

### 3.2.1 Metrics

We use certain metrics to evaluate training and evaluation performance. For the image classification use case, we primarily look at the categorical cross-entropy loss which is simply called loss in the figures referring to image detection. The top 1 and top 5 accuracies are used to measure how well the model predicts the correct class given an image and will be a value between zero and one. This accuracy measure will check if the correct class is present in the top x most likely predictions from the model, and do so for every image in the current test or evaluation dataset, then calculate the average success value.

Regarding the use case of object detection, two loss functions are used. One for classification and one for localization. The model's performance is then measured in average precision (AP) and mean average precision (mAP). AP is calculated in several steps, first, the intersection over union (IOU) is calculated for the predicted bounding boxes and the ground truth boxes. Then, given these IOU values, the predicted objects are sorted on their confidence score. From this, we can produce a precision-recall curve with precision and recall calculated at different confidence score thresholds. Precision represents the ratio of true positive detections to the total number of predicted detections, while recall represents the ratio of true positives to the total number of ground truth objects. AP is then the area under the curve. With this metric, we get a single value that represents both classification and localization. The mAP is then the mean AP taken over all classes in a multi-class problem.

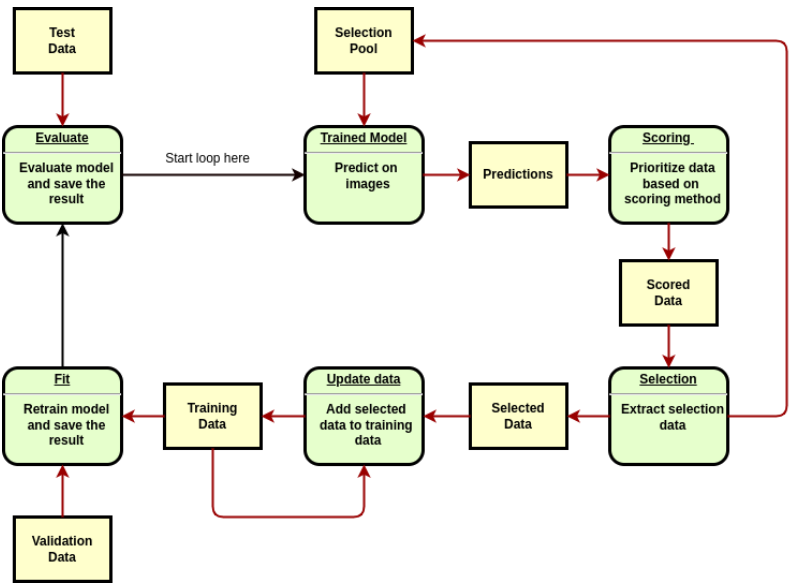


**Figure 3.2** Overview of the developed active learning pipeline. Predictions refer to the model output, like what class it thinks the image belongs to.

**Algorithm 1** The Active Learning Loop

**Require:** selection_pool, method, model, test_data, validation_data

```
procedure ACTIVE_LEARNING(...)
    for number_of_selections do
        sample, selection_pool ← SELECT_SAMPLES(method, selection_pool, model)
        train_data ← train_data + sample
        model.load_weights(weights)
        train_result ← model.fit(train_data, validation_data)
        eval_result ← model.evaluate(test_data)
        save(train_result, eval_result)
    end for
end procedure

procedure SELECT_SAMPLES(method, selection_pool, model)
    predictions ← model.predict(selection_pool)
    selection ← UNCERTAINTY_SAMPLING(method, selection_pool, predictions)
    sample, selection_pool_remainder ← GET_SAMPLE_SUBSET(seleciton_pool, selection)
    return sample, selection_pool_remainder
end procedure

procedure UNCERTAINTY_SAMPLING(method, selection_pool, predictions)
    if method is random then
        selection ← random.sample()
    else if method is least_confidence or margin_of_confidence or entropy then
        uncertainty_score ← PRED_TO_UNCERT(method, predictions, number_of_classes)
        selection ← tensorflow.argsort(uncertainty_score, 'DESCENDING')
    else if method is loss_descending then
        loss_fn ← model.compiled_loss
        losses ← CONVERT_TO_LOSS(loss_fn, selection_pool)
        selection ← tensorflow.argsort(losses, 'DESCENDING')
    end if
    selection ← selection[: budget]
    return selection
end procedure
```

### 3.2.2 Figure Explanation

In this report, we will present many active learning graphs. Thus it is important to note that these graphs will not be regular loss or accuracy graphs over epochs of a single training, but instead, every data point will be the result of a completed iteration in the active learning loop. Please refer to Figure 3.3 while reading the next paragraphs.

Unless stated otherwise, every data point represents a complete training, with the same model, the same starting weights, and the same hyper-parameters. The only difference is what training pool is available to the model. Each selection method will cumulatively select more data, meaning that later models will see the same and more data as the ones before. Then the difference between the selection methods will be the order the training pool is accumulated.

We run our experiments multiple times, which we represent as a colored shadow behind the average values, which is the plotted line. This shadow represents the standard deviation from the average. The number after the selection method in the legend shows how many times we ran that particular method.

There will also be graphs presenting the average number of epochs it took for a data point to complete its training. When early stopping is used we may see a difference between how long the different selection methods train for.

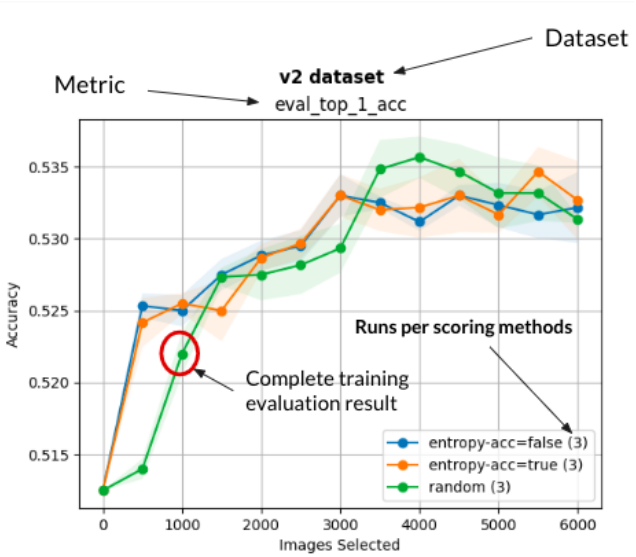


**Figure 3.3** A typical active learning figure. It describes what dataset, metric, scoring methods, amount of runs, etc. are used in an experiment. The shaded area is the standard deviation of multiple runs.

# 4

# Experiments

Before evaluating the selection methods we ran some tests, using the VGG-16 network [Simonyan and Zisserman, 2014], to make sure we would have a fair comparison between our scoring methods. The active learning pipeline, therefore, has some notable configuration parameters in the next section. Following the parameter optimization, the pipeline is used for two major experiments. First, we looked at the easier problem of image classification, and then the problem of object detection is studied by performing similar experiments with an object detection model.

## 4.1 Pipeline Configuration

We made some parameter tests and sweeps to evaluate the best way to evaluate the active learning selection methods. We also use the following tests to validate our assumptions and to ensure a fair comparison between our scoring methods.

### 4.1.1 Reshuffling

While testing our active learning pipeline, we noticed that the training loss and accuracy between selection methods were not the same even when the same data was selected. Essentially, two identical models produced different results on the same data. This seemed irrational and made us investigate this issue further. It turned out that the only differentiating factor was the data order. Since our selection methods are designed to select challenging images, a difficulty gradient will naturally occur in our training pool. This can seemingly lead to catastrophic forgetting, even during a single epoch. We assume that training on harder batches in the beginning, and easier batches at the end of an epoch could lead to worse performance on the harder images.

To mitigate this, the training pool needs to be shuffled after each accumulating selection step. Notice that our data was already shuffled before the active learning loop began, but that our implementation of active learning reintroduces a scoring-based order, which can affect the training stability. By shuffling our selected data pool after each active learning loop, we ensure that the training correctly ends with

similar training performance regardless of the data-scoring method. The difference between not shuffling your data after selection compared to shuffling can be observed in Figure 4.1 and 4.2. Especially note the convergence or divergence once all images have been selected.

From this point on, the reader can assume all datasets are shuffled after selection.

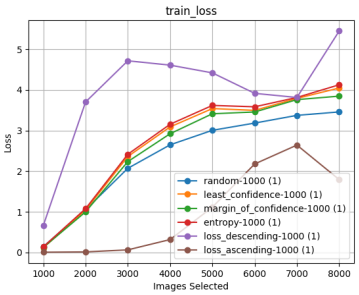


**Figure 4.1** Same data results in different loss when all images are selected

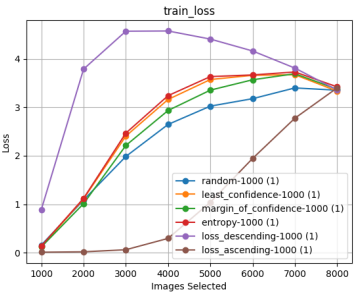


**Figure 4.2** Shuffle ensures same performance with same data with all images

### 4.1.2 Mitigating Effect of Data Quantity

In our attempts to achieve fair and comparable results between our scoring methods, the variance in data quantity was deemed a problem. Normally, one would train a certain amount of epochs, where each epoch would train on each image once. If the training pool is larger, the model trains longer. Should we ensure each model gets to train with a fixed time, or should we perhaps train until the model's data potential was reached, i.e., train until the model no longer improves on the data?

Early loops would have less data, and therefore get less backpropagations during training. The worry was that later loops would improve, not because of better data selection but purely on longer training times. One solution to this was setting a fixed steps per epoch value, and ensuring that the same amount of time and backpropagations occur regardless of data size. This solution, however, did not take into account overtraining the model, which could punish selection methods that are "too good" since they would get more time to overtrain. Under- and over-training could artificially skew the results, primarily the results at the beginning and end of the graph.

Another solution could be to instead train until the model stops improving. The comparisons would now be fair in the sense that you compare the data's training potential, instead of training time. With this mindset, larger training pools should be allowed to train for longer since they might have more potential. It could also avoid the issue of overtraining altogether. All this could be achieved with early stopping. During early testing, we found that accuracy seemed to improve even

after the loss had gone past optimum. It may be useful to test to evaluate based on accuracy instead of validation loss as well.

A test was designed with the above in mind, examining a regular epoch training (dynamic time) against fixed steps per epoch (fixed time) and two with early stopping (dynamic time, fixed potential). These two runs with early stopping look at the loss and the accuracy of the validation dataset respectively, and were configured with a patience value of 3 epochs, meaning that the model trained for 3 more epochs to check for further improvements before reverting back to the best-performing weights.

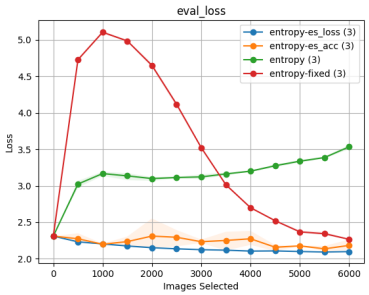


**Figure 4.3** Evaluation Loss. Both early stopping methods (blue and orange) generalize well as they maintain a low loss value. Note that early stopping on loss (blue) performs the best.


**Figure 4.4** Top 1 accuracy. Early stopping (blue and orange) performs better and has a high accuracy value throughout.

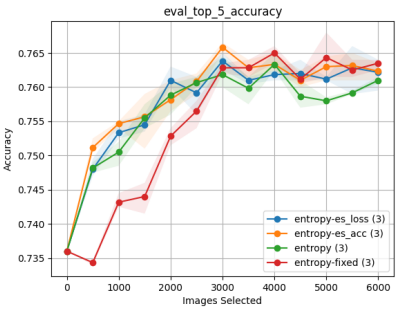


**Figure 4.5** Top 5 accuracy seems mostly unaffected.

Looking at the Figures 4.3, 4.4, and 4.5 we can draw some conclusions, here we are using the entropy scoring method. Using fixed steps per epoch, set to half the

selection data pool size, ensures that every epoch runs for the same amount of back-propagations. This seems to introduce an immense overtraining in the early stages of the evaluation, and undertraining towards the end. Early-stopping seems beneficial in comparing data quality, specifically noticeable in Figure 4.4. Early stopping on validation accuracy initially improves the learning rate a little, compared to early stopping on validation loss, only to perform worse a few selections later. We had noticed that accuracy often peaks a few epochs after loss does, meaning stopping on validation accuracy often results in a worse validation loss. This may in turn lead to worse selections later on. We conclude that early stopping on loss seems to be the best alternative.

### 4.1.3 Selection Pivot

Our data-scoring methods produce an ordered list of previously unseen data based on a trained model. Ideally, these methods would prioritize the images that will increase accuracy the most. However, we select them based on model uncertainty or loss. Is it reasonable to assume that the images with the highest uncertainty or loss will produce the largest accuracy gain?

To test the above assumption, we introduce a selection pivot, a float value between 0 and 1, around which the image selection will occur. Zero would represent selecting data that the model evaluates as "hardest" or would most likely improve accuracy, and a value of 1 would result in selecting the data that the model evaluates as the "easiest" and thus is least likely to improve accuracy. Since we have an ordered list of data, we can also choose to select around an arbitrary pivot, like 0.5, and chose images that are neither hard nor easy. A parameter sweep over this pivot value results in Figures 4.6 - 4.9.

The training results on entropy selection (Figure 4.6) show that selecting harder images (pivot = 0 => p0.0) makes it harder to achieve the same training performance compared to selecting easier (pivot = 1 => p1.0) images, given the same amount of epochs. This demonstrates that uncertainty sampling does indeed succeed in selecting challenging images and that the active learning concept is working as intended.

Interestingly, looking at evaluation accuracy (Figure 4.7) it seems to indicate that you can speed up learning by selecting slightly easier images instead of the most challenging ones. Improvement in early performance is observed with a pivot of 0.25. However, this pivot value's accuracy gain seems to stagnate halfway through, resulting in it being overtaken by pivot 0 at the end.

Let us now look at the same graph with early stopping, to ensure overtraining is not skewing the results. Let us also compare against random selection. Figure 4.8 looks at generalization loss, and surprisingly both p0.25 and p0.50 seem to beat random, while normal entropy p0.0 does not. Figure 4.8 looks at generalization accuracy, and just like previously observed without early stopping, entropy with a pivot of p0.25 is gaining accuracy faster, and stagnating after a few selections.

One explanation for the observed results could be that certain challenging im-

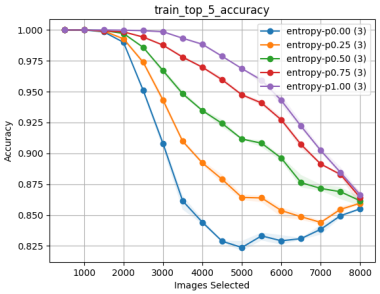


**Figure 4.6** Training accuracy falls predictably when problem difficulty increases given a fixed epoch size.

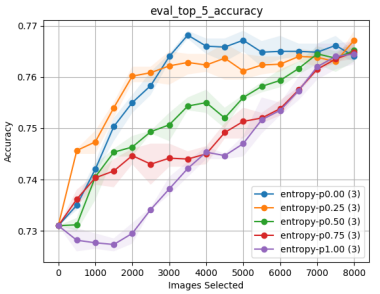


**Figure 4.7** Evaluation results with a variance of selection pivot points.

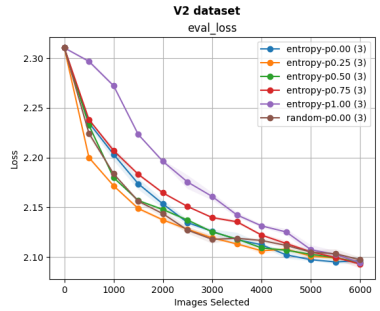


**Figure 4.8** Pivot loss. Both $p0.25$ and $p0.5$ seem to perform slightly better than random, while $p0.0$ does not.

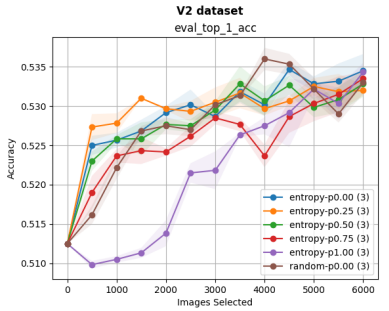


**Figure 4.9** Pivot accuracy. Entropy $p0.25$ seems to perform better than the others in the first few loops.

ages are hindering learning accuracy, while others are required to reach the best accuracy. The results could indicate that there are outliers present in our data, and that said outliers are producing high model uncertainty. Selecting these images results in worse generalization accuracy. Is this a reasonable hypothesis? One way to check is by using human validation on the first top 100 selected images and noting down any personal disagreements regarding data labeling or quality issues. Doing so resulted in Table 4.1.

Dataset-specific quality analysis is outside the scope of this report, and looking at 100 data points for a specific selection method is hardly proving statistical significance. However, looking at selected images for entropy-p0.0 compared to random results in a $\sim 70\%$ decrease in data quality. There is also a serious difference in "disagreements" between p0.0 and p0.25, meaning that label quality seems to increase when not selecting the images entropy selection considers hardest. This seems to

**Table 4.1** Human validation on top 100 selected images and their labels

| | random | entropy-p_0 | entropy-p_0.25 |
|---|---|---|---|
| disagreements | 23/100 | 39/100 | 29/100 |

indicate that active learning selection methods like entropy could be used for detecting outliers, improving annotation, or simply avoiding out-of-distribution data. This may also explain the results in Figure 4.7, since selecting points around the 0.25 pivot point results in significantly fewer outliers.

### 4.1.4 Budget Size

The term budget comes from the annotation world and defines how many images one can afford to annotate during the next active learning loop. In our case budget simply means how many images should be selected in between the training of the model. But what budget size is best suited for a given dataset and does budget size affect performance? During the testing of the VGG-16 model, we ran experiments on how budget size affects the active learning process. Those results can be seen in Figure 4.10 and 4.11. Looking at the y-axis, it seems that the budget size has no major impact on the training. However, going beyond a budget of 1500 images seems to start to degrade learning speed, i.e., it takes more data to achieve similar performance. This could very well be data-specific, but lower budget values seem to allow for faster improvement.

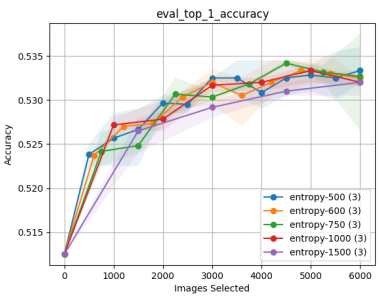


**Figure 4.10** Top 1 accuracy on the test data. Only a budget of 1500 seems to perform slightly worse.

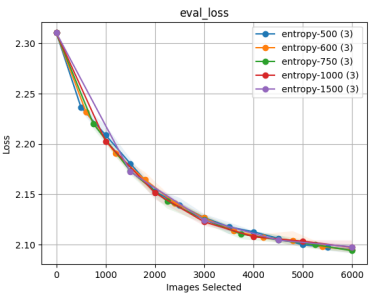


**Figure 4.11** Loss on the test data. No apparent effect.

### 4.1.5 Accumulation or Reselection

We had an idea that it might be beneficial if we let the model re-select all images after every loop. Instead of accumulative building a training pool, perhaps the model could create its pool every loop. The motivating thought here was that the model might pick images it is uncertain about in the first few loops, but later has no need for or has unnecessarily many examples of. Instead then, through re-selection, it could select images it considers hard at the moment. While this could be beneficial, training on certain images and not re-selecting them would essentially result in catastrophic forgetting, since the model gets reset in between loops.

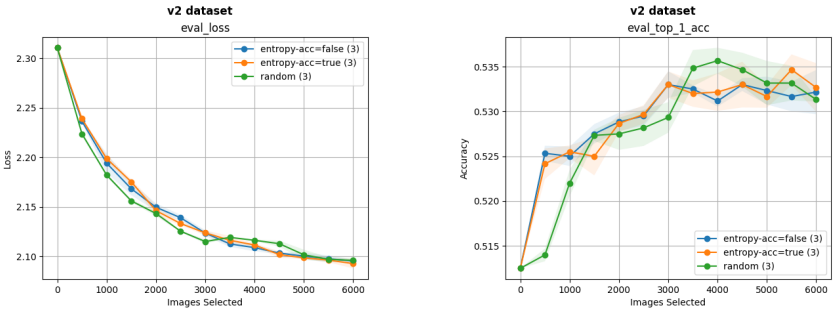


**Figure 4.12** Evaluation loss and top 1 accuracy. Comparing Accumulation on or off (blue vs orange) does not indicate a statistical difference.

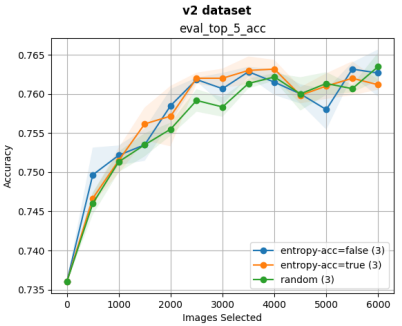


**Figure 4.13** Top 5 accuracy.

Figure 4.12 and Figure 4.13 are the results of this test. This quick test was mostly to see if there would be a change in performance with accumulation on or off. Random is there as a reference. The two scoring methods mostly lie inside each other's standard deviations. While this needs to be studied more to ensure we are

not missing some benefit here, we decided to end our testing on this topic with the takeaway that it seems to have little effect.

## 4.2 Image Classification

### 4.2.1 Model and Datasets

To test the different selection methods applied to the classification problem we used the VGG-16 network [Simonyan and Zisserman, 2014] with weights pre-trained on the training split of ImageNet. For our experiments, we have used the following different ImageNet test datasets. For clarity, these are separate from the ImageNet dataset but have a similar name because of their label space, i.e., they have been annotated using the same 1000 labels as ImageNet.

- **ImageNet-V2** is a dataset of 10000 images, with an even class distribution [Recht et al., 2019].
- **ImageNet-R** is a dataset with 30000 images that has a focus on art [Hendrycks et al., 2020].
- **ImageNet-A** contains 10000 especially 'hard' images [Hendrycks et al., 2019].

These test datasets were then used to create the following 4 new datasets. These are all arranged to be 6000 training images and 2000 validation and 2000 test images. They are presented in order of difficulty to classify.

- **V2** is our baseline dataset, with images only from ImageNetV2. This is a comparatively easy dataset as it contains similar images to ImageNet.
- **Mixed** is a mixed dataset, containing an equal split of data among ImageNet-V2, ImageNet-R, and ImageNet-A. This is a slightly harder dataset to classify.
- **Noisy** is the same as the mixed dataset, but the noise has been introduced to the images, forcing a larger domain shift from the original ImageNet dataset and resulting in additional difficulty. See Figure 4.14 for an example of the added noise.
- **Hard** is the same as Noisy but has every image repeated 4 times, with a small random rotation. The validation and test sets do not have repeated images. This is to imitate a surveillance scenario where similar images are common.

**Figure 4.14** ImageNet-V2 image: normal vs noisy

### 4.2.2 Diversity Sampling

Diversity sampling is when you sample data based on diversity aspects, like what dataset an image comes from or what label it has. It may be used in combination with confidence sampling or may simply be used by itself. To evaluate if diversity sampling could be beneficial for us, we ran a test with diversity sampling applied to the selection produced by random sampling and entropy, one of the confidence sampling methods. The results can be seen in Figures 4.15 - 4.18.

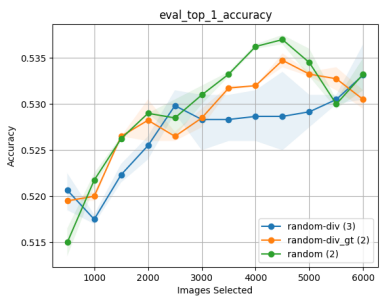


**Figure 4.15** Top 1 accuracy. Diversity sampling by itself using the v2 dataset. Random outperforms diversity sampling.

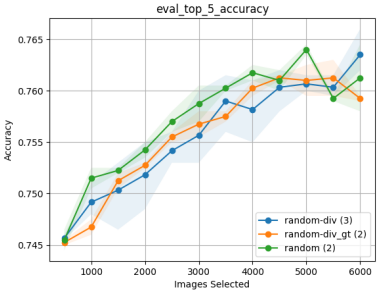


**Figure 4.16** Top 5 accuracy. Diversity sampling by itself using the v2 dataset. Random still outperforms diversity sampling, but not as much.

Some takeaways can be made here. Interestingly diversity sampling seems to hurt the accuracy performance when matched with random sampling. Ensuring class diversity given a random base order seems to worsen test accuracy. This effect is

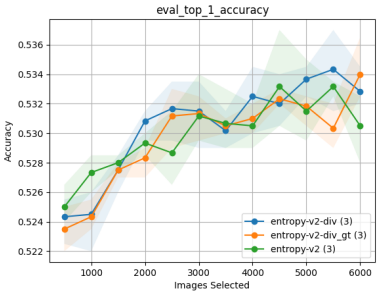


**Figure 4.17** Diversity together with entropy scoring using the v2 dataset. The results are too close to draw a meaningful conclusion here.

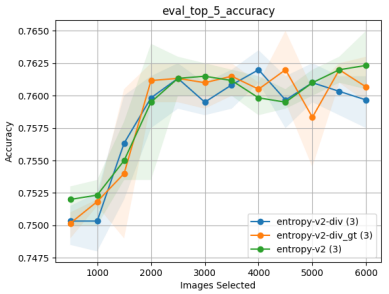


**Figure 4.18** Diversity sampling with entropy scoring using the v2 dataset. No real difference can be observed.

less noticeable when using a real selection method like entropy sampling. Adding diversity sampling does not improve the selection accuracy, however. Notice the y-axis on Figure 4.17 and 4.18, and how entropy and entropy-div's ranges overlap. There seems to be no significant difference between them.

Hypothetically, seeing a diverse dataset containing a wider range of classes should improve the training data as the model is less likely to forget or favor a subset of classes. So what could explain the results above? Since we are sampling based on class labels (or the predictions thereof), the class distribution in the dataset may be of importance. A balanced dataset already fulfills the requirement of a balanced selection even when using random selection. Our primary dataset, ImageNetV2 is a balanced dataset with 10 images per class (as seen in Figure 4.19). A more diverse dataset may be required to see the possible effect of diversity sampling.

We therefore make use of our mixed dataset, as both the ImageNet-R and ImageNet-A are imbalanced the final mix is also imbalanced with a class distribution as presented in Figure 4.20. Most classes now only have about 5 images, while certain classes have more than 80 examples, as opposed to the previous 10 per class in ImageNetV2.

Rerunning the same test as before on this unbalanced dataset results in Figures 4.21 - 4.24. Once again, using the selection method without diversity sampling applied afterward seems preferable, for both random and entropy selection. Even on an unbalanced dataset like ours, slightly modifying the priority order to be more class-representative does not appear to significantly improve training.

### 4.2.3 Data Selection Methods

It is time to compare the remaining selection methods, uncertainty selection, and loss selection. To compare the different methods, we permute all datasets and every selection method. We use a batch size of 20 and a budget of 500. We also use early

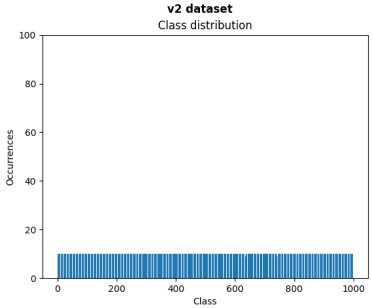


**Figure 4.19** Class distribution on the balanced v2 dataset.

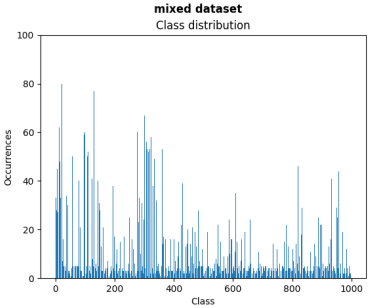


**Figure 4.20** Class distribution of our mixed dataset.

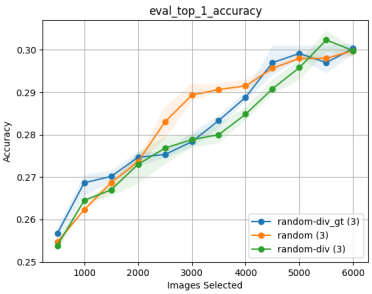


**Figure 4.21** Diversity sampling on random using the mixed dataset.

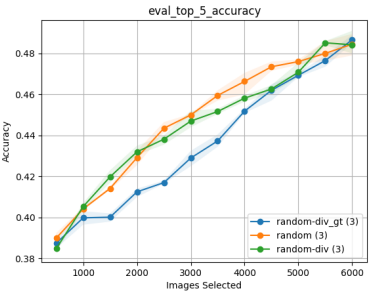


**Figure 4.22** Diversity sampling on random using the mixed dataset.

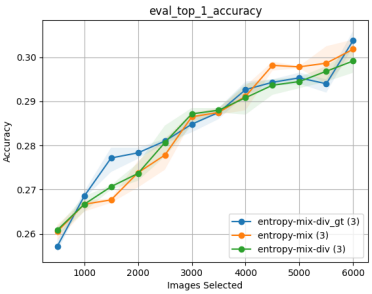


**Figure 4.23** Diversity sampling on entropy using the mixed dataset.

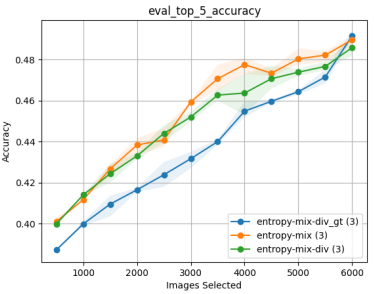


**Figure 4.24** Diversity sampling on entropy using the mixed dataset.

stopping looking at the validation loss with a patience value of 2. The training runs a maximum of 1000 epochs, early stops, and then resets to the best weights before evaluating the model. We plot the average of the runs with a line and the shaded area is the measured standard deviation from the average.

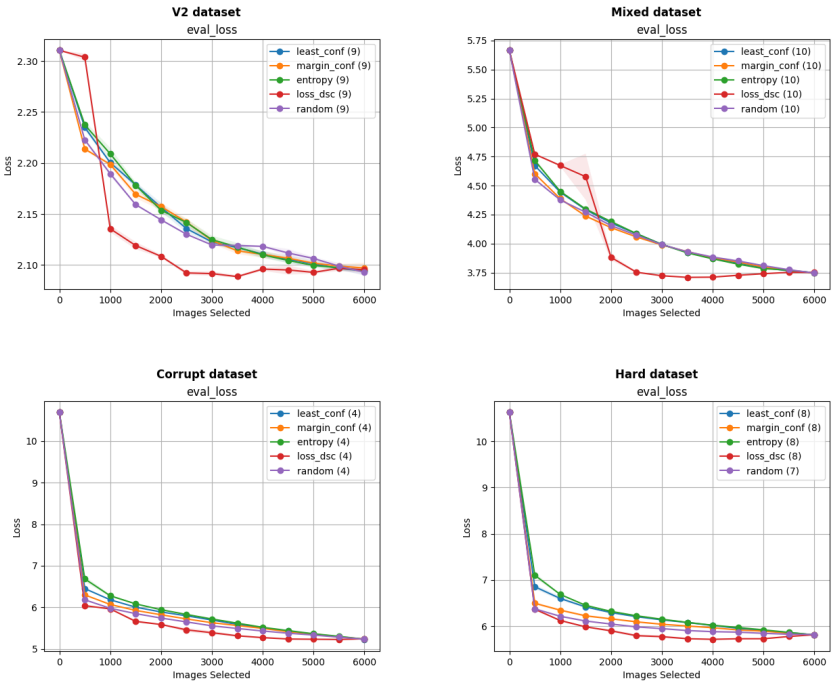


**Figure 4.25** Test loss on all datasets and methods. Note: Corrupt dataset corresponds to the noisy dataset.

We start our analysis by comparing our evaluation loss among the different datasets. Looking at the loss in Figure 4.25, we notice that loss descending seems to consistently be the best selection method to reduce loss. In the second place, random selection consistently seems to outperform uncertainty selection methods.

That loss selection performs well is logical, since we select images that produce high loss and train on them, resulting in a larger loss improvement. Random performs better than uncertainty selection methods, possibly due to a more fair sampling strategy, that across 1000 classes seems to improve the output vector overall. In the same way, uncertainty sampling's worse performance could be attributed to more targeted improvements, that could improve certain classification scenarios in a few classes, but not others. It is also worth mentioning, that among the uncertainty-based selection methods, entropy generally seems to perform worst in evaluation loss and margin of confidence seems to perform the best.

Let us focus on loss descending, especially on the v2 and mixed datasets. We notice a loss minima after selecting half of the training pool, 3000 images, after which the loss seems to rise again. When looking at random selection, we also notice a saddle point on v2 data. This further indicates the importance of 3000 images in our v2 dataset test. Even in the hard dataset, our most realistic dataset with a lot of images similar to each other, there seems to be an inflection point at 4000 images selected. One hypothesis we had going into this thesis work was that data prioritization may question the thought that more data is always better. The fact that our loss has reached a minimum before selecting all available data indicates that not only can active learning let you achieve higher accuracy with fewer data, but it can also indicate when you have exhausted your current data pool, or retroactively indicate that the model may be training on too much data. If that is the case, pruning images may improve generalization loss.

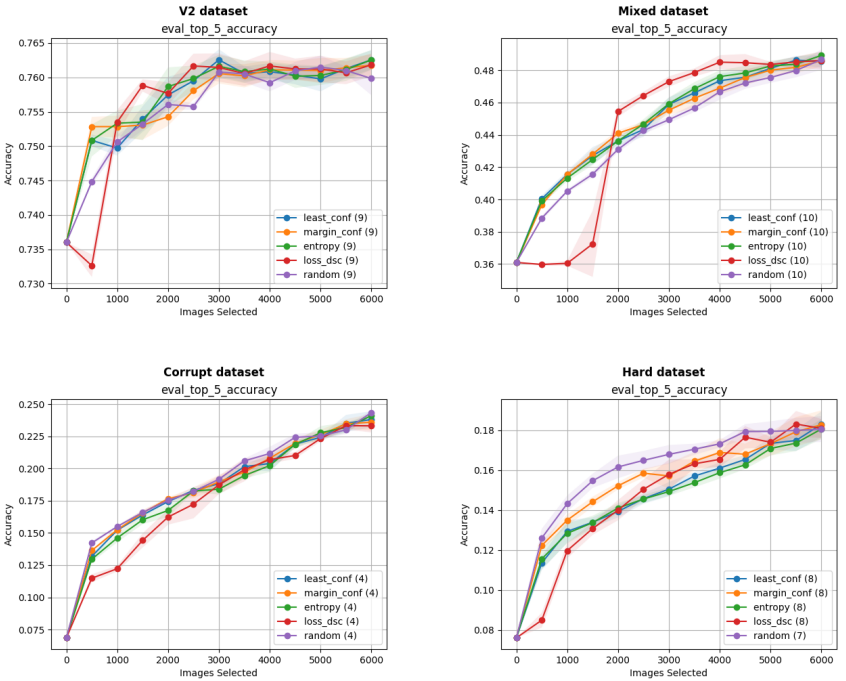


**Figure 4.26** Test accuracy on all datasets and methods. Note: Corrupt dataset corresponds to the noisy dataset.

Let us instead look at the accuracy comparison in Figure 4.26. For convenience reasons, we highlight the top 5 accuracies here, since the differences between selection methods are more pronounced. The top 1 accuracy figure can be found in

the appendix section 7 (Figure 7.1). Note the different scaling on the y-axis. While v2 may look noisy, it is mostly due to very minor accuracy improvements, resulting in a zoomed-in figure. We remind that the number next to the selection method indicates how many times the experiment was run.

Let us start with the promising aspects. In both the v2 and mixed datasets, the easier out of the four datasets, our selection methods seem to outperform random selection. Just like when we compared loss, loss descending seemingly needed a few selections to get going, but performed the best in those cases. However, with the two harder datasets, noisy and hard, random is equally good or better than our selection methods. This seemingly indicates that using an intelligent selection method only seems to outperform random selection given we already are somewhat good at the domain. One simple explanation for this could be that the worse our domain knowledge, the worse our selection, and the worse accuracy we can achieve. It is hard to draw a conclusion on performance for the rest of the methods, as we did for loss since they are not consistent. For example, looking at the hard dataset results, loss descending and entropy seem to perform the worst even though we know they have quite different loss values. We note that loss descending performs the best on the easier datasets, but draw no other conclusions.

How do our loss observations stack up while looking at accuracy? We previously noted that there was an optimum loss at around 3000 images on the v2 and mixed dataset. Looking at v2 accuracy, we see that indeed there seems to be no improvement beyond 3000 selected images. Looking at the mixed dataset, however, there is a clear improvement in accuracy beyond the 3000 image point, peaking and staying at 4000 images, even though 3000 is the optimal loss value. This is something we observed while testing early stopping as well. Our accuracy usually kept improving beyond the optimal loss value for a few epochs. Given that these datasets have 1000 label classes, and we are testing accuracy on only the top 1 or top 5 classes, this is not unreasonable. The model can improve the accuracy while worsening the loss if producing more noisy predictions. However, this indicates that our loss function, the function we optimize for, does not guarantee the best accuracy, the result we seek. While outside the scope of our report, perhaps cross-entropy is not the best loss function for these sets of data, since there seems to be a mismatch between accuracy performance and loss performance across multiple selection methods. Perhaps a loss function with a focus on top predictions would be more fitting, than one that treats all classes the same.

Some takeaways on evaluation loss and accuracy can be made:

- Loss selection performs best on loss, followed by random selection, followed by uncertainty selection methods.
- The results indicate that not training on all data may improve generalization loss
- Loss selection results in good accuracy on easier datasets.

### 4.2.4 Base Training Set

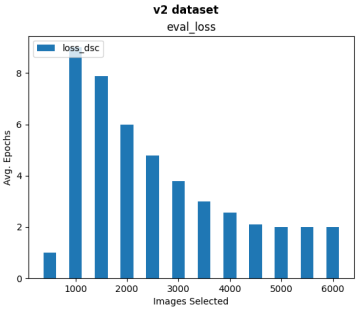


**Figure 4.27** Epochs analyst of suspicious loss scoring figure on the v2 dataset.

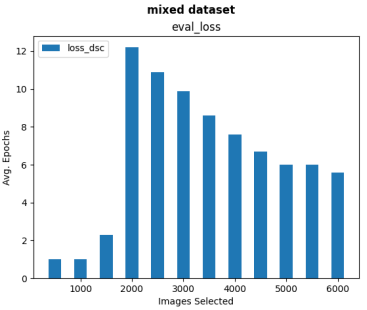


**Figure 4.28** Epochs analyst of suspicious loss scoring figure on the mixed dataset.

For the two easier datasets, v2 and mixed, loss descending seems to struggle in the beginning. On closer inspection, this is because the model early stopped almost instantly (shown in Figure 4.27-4.28). The images it selects in the first few loops on v2 and mixed seem to not be representative of their respective validation data, hence the early stopping. Why is that? We do not notice this on the two harder datasets, where there has been a bigger domain shift, so it is not the difficulty of the task that is the problem. Perhaps the opposite is true. Selecting images based on true loss on a well-known data domain might result in too good of a selection, meaning it successfully selected the hardest images from the pool. These images may be outliers, or could together form a different data domain and no longer generalize well to the original problem. Since we only train on the selected images, we may introduce a shift to a harder domain, resulting in it getting early stopped. This result may necessitate a revision in our comparison method. This may very well be a result of it only training on 500 images in the first loop. It seems a fairer way to compare methods would be to introduce a representative base pool of images, randomly selected, to avoid catastrophic forgetting on 'too good' early selections. This could also tie into our previous pivot results, where we observed not selecting the hardest images can result in faster training. That could have been connected to the mentioned above issues and should be re-evaluated with a base pool of images.

As we saw in the previous section, methods like loss selection may struggle to generalize well with very few images. To accommodate this we introduce a base training set of randomly selected images that all scoring methods will start with. This should prevent early selections to have too little data to generalize to the validation subset and result in a more fair comparison. This is a realistic scenario since most commonly one would have a base set of annotated data that one wishes to extend using active learning.

We focus especially on the mixed dataset, where the loss descending curve be-

haves the worst. In Figure 4.29 we look at the result of adding 500 random images as a base pool selection. For the sake of comparison, only one of the uncertainty selection methods is presented here, since most results up till this point have shown little difference between them. In the left column, the results from Section 4.2.3 are presented again, and in the right column, the results with the base set are shown.

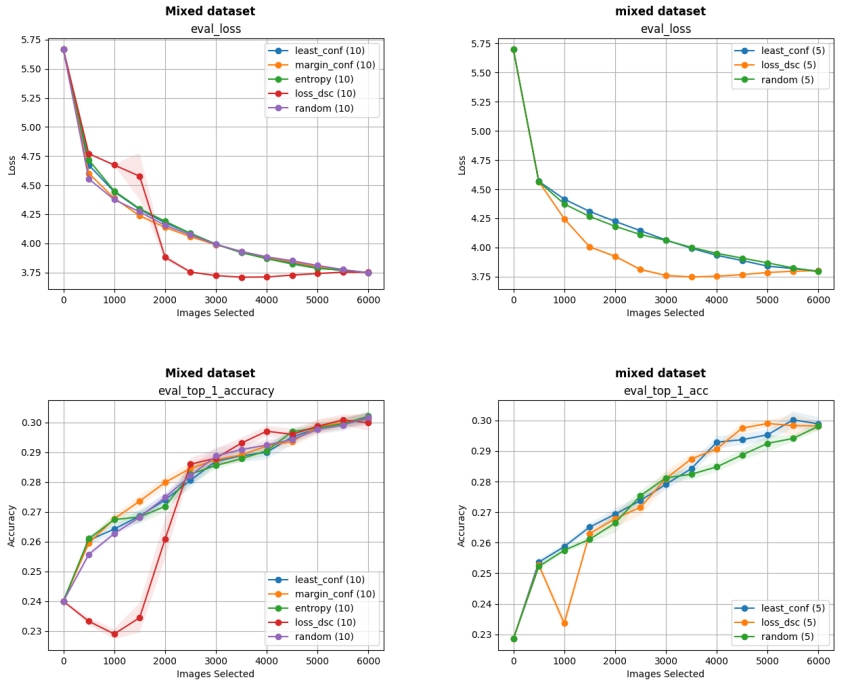


**Figure 4.29** In the left column we show methods without a base train pool of data. The right column shows the result if the first 500 images are randomly selected regardless of what selection method is used. We look at the mixed dataset.

We can see that introducing a randomly selected base pool of data resulted in a much more reasonable active learning curve for loss selection. This seems to indicate that our previous hypothesis of the selection generalized badly to the validation set may have been correct. The same can be observed in the v2 dataset, which we put in the appendix (Figure 7.5).

Looking at the top 1 accuracy in Figure 4.29 (bottom right) we still see a dip in accuracy after the base data set, at 1000 selected images. This may indicate that a larger base data set may be required to ensure the first loss selection is not worsening the accuracy.

We run a final test with 1000 images as a randomly selected base test, and on every method tested so far (Figure 4.30). There is still a slight dip downward, so

given a larger dataset, a bigger base pool might be recommended. However, all our other observations still hold. Loss entropy seems to find an optimum at 4000 images for example.

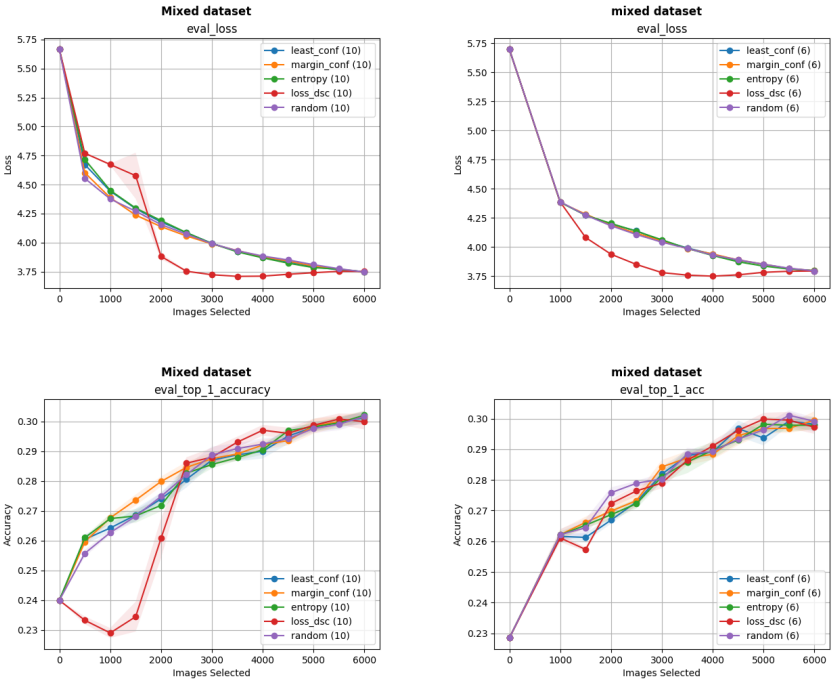


**Figure 4.30** The effect of adding a base set with 1000 images. The left column shows methods without base data. The right column shows methods with 1000 base images

We conclude that a base set should be provided when comparing different scoring methods for active learning. If accuracy goes down on the first iteration after the base set, a larger base set should be considered.

### 4.2.5 Statistical Significance

We notice some things looking at the results from Section 4.2.4, especially the Figure with 1000 randomly selected base images (Figure 4.30). Of the methods described in *Human-in-the-Loop Machine Learning* [Monarch, 2021], least-confidence, margin-of-confidence, and entropy are all quite similar in performance. They all fall under the umbrella term uncertainty sampling, defined by looking at the score vector the model produces. However, uncertainty sampling and loss sampling both seem to differ from random sampling in test performance. In this section, we try to prove that this difference is statistically significant.

**Table 4.2** AUC values for evaluation (test) dataset. AUC is scaled by $10^{-2}$.

| method | top 1 acc | top 5 acc | top 20 acc |
|---|---|---|---|
| least conf | 16.60 | 26.98 | 36.59 |
| margin conf | 16.64 | 26.98 | 36.62 |
| entropy | 16.61 | 27.02 | 36.65 |
| loss dsc | 16.63 | **27.31** | **37.62** |
| random | **16.70** | 26.76 | 36.19 |

**Table 4.3** t-test p-value for top 1 acc

| | least conf | margin conf | entropy | loss dsc | random |
|---|---|---|---|---|---|
| least conf | 1.00 | 0.02 | **0.42** | 0.02 | 0.00 |
| margin conf | 0.02 | 1.00 | **0.18** | **0.66** | 0.05 |
| entropy | **0.42** | **0.18** | 1.00 | **0.27** | 0.01 |
| loss dsc | 0.02 | **0.66** | **0.27** | 1.00 | 0.02 |
| random | 0.00 | 0.05 | 0.01 | 0.02 | 1.00 |

Since we know that 0% and 100% of the data should result in identical accuracy, regardless of the selection method, we can use the area under the curve (AUC) to determine the effectiveness of the data selection method. A larger AUC value indicates more effective learning with less data, which by extension would say something about the quality of our selection method and the selected data. Looking at the average AUC values of our results from the Base training section (Section 4.2.4, Figure 4.30) we get the Table 4.2. From this table, we see that loss descending selection seemingly generalizes well to the overall problem, consistently beating the other methods except on top 1 accuracy. Interestingly, in the top 5 and top 20 accuracy values, random selection performs the worst but still ends up on top in the top 1 accuracy score.

But are these results statistically significant? We ran the experiments multiple times to establish a standard deviation area. By collecting the different AUC values during different runs of the selection method, we can create an AUC set distribution that we can compare with other selection methods' AUC distribution in a t-test. In Table 4.3, Table 4.4, and Table 4.5 we have calculated the permutation table of p-values from the different ACU distributions per method.

The top 1 accuracy t-test permutation table shows that except for random, which is statistically unique, the other methods are not different enough to be statistically differentiable. Getting the right class out of 1000 images seems to be a difficult

**Table 4.4** t-test p-value for top 5 acc

| | least conf | margin conf | entropy | loss dsc | random |
|---|---|---|---|---|---|
| least conf | 1.00 | **0.94** | **0.25** | 0.00 | 0.00 |
| margin conf | **0.94** | 1.00 | **0.28** | 0.00 | 0.00 |
| entropy | **0.25** | **0.28** | 1.00 | 0.00 | 0.00 |
| loss dsc | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 |
| random | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 |

**Table 4.5** t-test p-value for loss

| | least conf | margin conf | entropy | loss dsc | random |
|---|---|---|---|---|---|
| least conf | 1.00 | **0.14** | 0.00 | 0.00 | 0.00 |
| margin conf | **0.14** | 1.00 | **0.58** | 0.00 | **0.07** |
| entropy | 0.00 | **0.58** | 1.00 | 0.00 | 0.04 |
| loss dsc | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 |
| random | 0.00 | **0.07** | 0.04 | 0.00 | 1.00 |

problem, and trying to select a few thousand images to improve on this does not seem to be enough. This probably boils down to the issue of a smaller dataset, since random seems to outperform the other methods here.

Looking at the top 5 and top 20 accuracies we see that the different uncertainty scoring methods, least confidence, margin of error, and entropy all are very similar in performance and are not significantly differentiable. Which of these ends up getting chosen does not have a significant difference in the results of this experiment. They are however mostly differentiable from random and loss.

## 4.3 Object Detection

### 4.3.1 Model and Datasets

For the problem of object detection, we are using a Single-Shot Detector (SSD) model [Liu et al., 2015] with a backbone for extra feature extraction. The model handles 8 classes and one extra class set as background. For the experiments in this section, we have used parts of three different datasets:

- **Large**: This dataset is a general large-scale dataset that like the ImageNet is multi-class. It depicts many real-life situations.

- **Surveillance**: The second dataset also consists of real images but from a surveillance perspective and it mainly depicts cars and people. This dataset has fewer classes than the Large dataset.

- **Synthetic**: The third and final dataset is a computer-rendered synthetic dataset that contains images of city environments, also from a surveillance perspective. This dataset has fewer classes than the Large dataset.

As this model is larger and takes longer to test compared to VGG-16 used in the image detection case, we have chosen to use the knowledge we have obtained from our previous experiments with said VGG-16 network and mainly used the same hyper-parameters henceforth.

Throughout the experiments, we used a few different sets of pre-trained weights, which we call baselines. These baselines have been trained for 25 epochs, comparatively a rather small and quick training given the model complexity, on different sets of images from a given dataset. Further, due to hardware limitations, we had to fix the image size to 512x288, which is not the resolution the model was designed for.

It is important to understand how we have implemented the selection methods for the object detection model. The traditional selection methods, least confidence, the margin of confidence, and entropy, similar to the image classification model only work with class scores. With object detection, we also have to take the box encoding prediction into account. Another difference is that we get thousands of predictions for different boxes in the image, but disregard all but the top detections, calculate the uncertainty score for the detections individually, and then take the average as the final score.

For the loss descending selection method, we are however looking directly at the sum of the classification- and location loss of an image and using that as the uncertainty score. As for the graphs, note that the evaluation loss now is a function of class prediction loss and box encoding loss. We are using mAP and individual class AP with an IOU score threshold in the range 50% - 100% as a measure of accuracy.

### 4.3.2 Data Selection Methods

Using the parameters observed to achieve the best performance from the experiments on the VGG-16 network, we applied the selection methods to the object detection model. To repeat, those parameters were a batch size of 20, a budget of 1000 images, and early stopping set to look at the validation loss with the patience of 3 with a maximum number of epochs set to 100. For this experiment, we used a pre-trained model trained on about 70000 images of the Large dataset for 25 epochs

and used the Surveillance dataset for selection, validation, and evaluation. The results may be viewed in Figures 4.31 - 4.34. These graphs are less stable than the image classification graphs, which could be due to the harder problem it tries to solve, yet they show a promising trend that active learning is working. We also note that loss descending and entropy drops rapidly and achieve a low evaluation loss throughout. However, the margin of confidence method seems to accomplish a better mAP and AP score faster in Figure 4.33 and 4.34 respectively. While the trend looks promising, no real comparison between scoring methods can be drawn here.

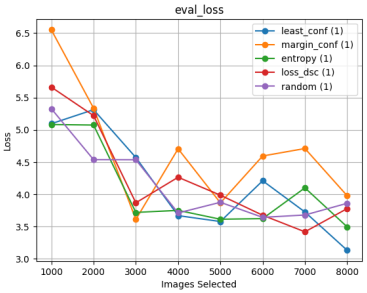


**Figure 4.31** Loss as a function of class prediction loss and box encoding loss.

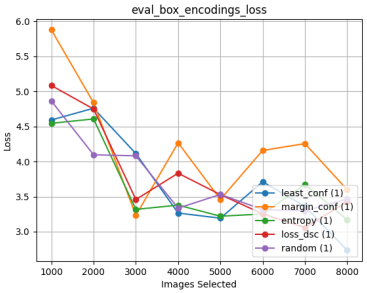


**Figure 4.32** Main contributor to the evaluation loss, the box encoding loss.

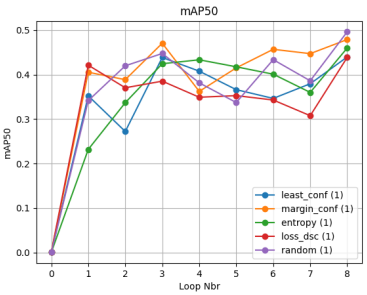


**Figure 4.33** mAP score

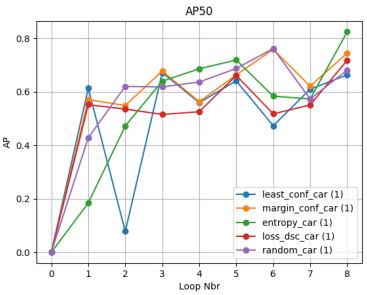


**Figure 4.34** AP score of the class "car".

### 4.3.3 Baseline Matters

In this experiment, we wanted to see how different baselines may affect the data selection and the training. We define a baseline as a set of pre-trained weights, that we use as a backbone for training. We created three "small" baselines:

- **Real**: This baseline is trained on about 10000 images from the Large dataset for 25 epochs.
- **Synthetic**: This baseline is trained on about 10000 synthetic images from the Synthetic dataset for 25 epochs.
- **Untrained**: This baseline has not been trained and thus has random weights.

These baselines are considered small since they are trained on a small dataset for a short amount of time. They were then tuned, evaluated, and tested on the Surveillance dataset. The results are shown in Figures: 4.35 - 4.38 for the Real baseline, 4.39 - 4.42 for the Synthetic baseline and 4.43 - 4.46 for the Untrained baseline.

***Real Baseline.*** We see in Figures 4.35 - 4.38 that the selection methods entropy, loss descending and random for most iterations are within each other's standard deviation from the average (shaded area behind the line). This means that the methods perform essentially the same over time. Entropy has a smaller deviation, indicating a more certain selection, after 4000 selected images in Figure 4.36. Entropy also maintains a better score than random until the end. We see a different case with loss descending, it instead deviates from the average and gets more uncertain after 4000 images. Most notably it starts with the best scores in all graphs but then does not improve much over time. Both entropy and random behave quite similarly to each other in their mAP50 score except for 1 data point where random is better than entropy. However, for the individual AP score of class "car" in Figure 4.38 we can observe that entropy makes a better selection in the first 3 iterations and achieves a better score than random. Both methods thereafter stagnate and do not further improve the AP score notably. Regarding the number of epochs each training takes, we see that both entropy and loss descending are taking longer than random generally. This is to be expected as both methods pick "harder" images to train upon, this is most apparent in the first iteration where loss descending trains for the longest. After the first iteration, entropy takes the lead and trains longer.

***Synth Baseline.*** Moving on, we apply the same parameters and tuning dataset to the Synthetic baseline and observe a very different result. The curves are even less stable (bigger deviations) and the results in Figures 4.39, 4.40 and 4.42 do not come near in achieving the same scores when comparing to the Real baseline. We see, however, in Figure 4.41 that in the end, the model can reach about the same mAP score as the Real baseline. Nevertheless, the slope is not as steep as we would want it to be. This could be due to there being too big of a difference between the Synthetic dataset which the model trained on and the Surveillance dataset which was used as the selection pool. In other words, the domain shift between the datasets could be too big. If that is the case, we see an interesting change in the selection methods' behavior in the evaluation loss graphs.

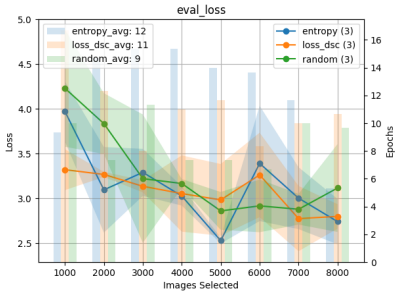


**Figure 4.35** Evaluation loss. The Real baseline was tuned and evaluated on the Surveillance dataset. The number of epochs is also shown.

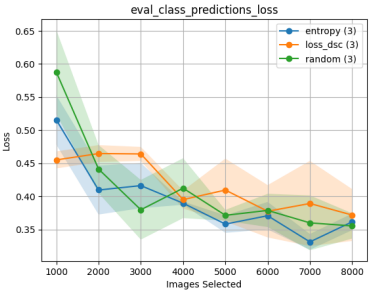


**Figure 4.36** Class prediction loss using Real baseline tuned and evaluated on the Surveillance dataset.

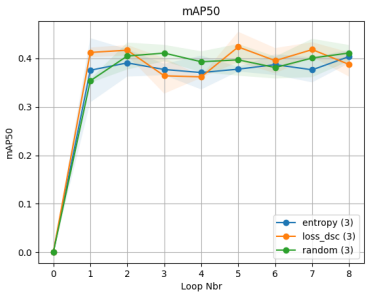


**Figure 4.37** mAP score of the Real baseline tuned and evaluated on the Surveillance dataset.

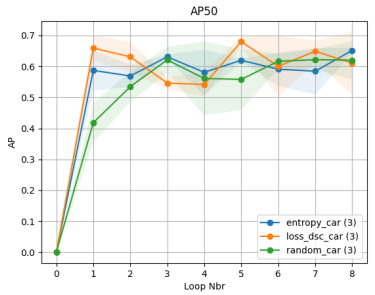


**Figure 4.38** AP score of the class "car" using the Real baseline tuned and evaluated on the Surveillance dataset.

In Figures 4.39 and 4.40, entropy performs slightly worse than random in general suggesting that the model has a hard time solving the problem. This is consistent with the number of epochs not declining over time, even with random selection it is difficult for the model to learn. However, we see that loss descending has a much better evaluation loss for the first three iterations than the other two selection methods. Remember the difference between entropy and loss descending, the first is based on class predictions whereas the second is based on both class predictions and box encoding. Both achieve the same training loss as seen in Figure 7.4 in the Appendix section 7, but since loss descending has more information to work with it makes a superior selection which in this case helps the generalization in Figure 4.39. In line with the Real baseline, we see a break around 4000 images where the model does not improve much after, again suggesting that there is not a need to train

on the whole selection pool. Though in Figure 4.41 and Figure 4.42, the model still improves after the fourth iteration for entropy and random. We see that the average entropy curve seems to be slightly better than random when it comes to the mAP and AP score than it is at achieving a low evaluation loss, which is contradictory as they are tested on the same dataset split. Then again, these results have a high deviation and thus are less reliable to make conclusions from.

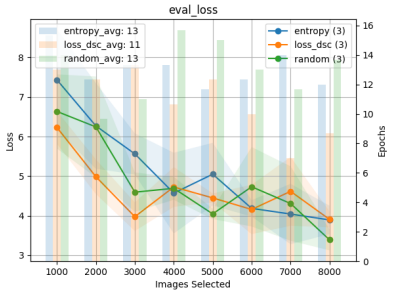


**Figure 4.39** Evaluation loss. The Synthetic baseline was tuned and evaluated on the Surveillance dataset.


**Figure 4.40** Class prediction loss using the Synthetic baseline tuned and evaluated on the Surveillance dataset.

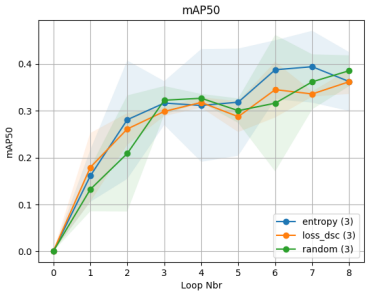


**Figure 4.41** mAP score of the Synthetic baseline tuned and evaluated on the Surveillance dataset.

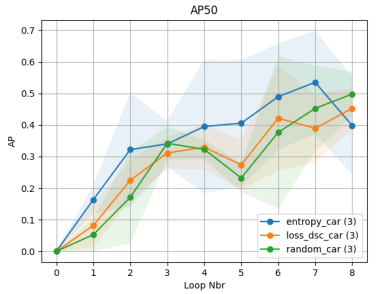


**Figure 4.42** AP score of the class "car" using the Synthetic baseline tuned and evaluated on the Surveillance dataset.

***Untrained Baseline.*** Finally, we have the experiment parameters applied to the Untrained baseline. We see that this model has a high evaluation loss in Figure 4.43, and in this case the main contributor to the loss is the classification of images which is contradictory with the other models where the box encoding loss is higher. Still, it is to be expected that this model would perform worse than the other models as it has no previous information to work with. Again, all selection methods are rather

close to each other, meaning we cannot draw a real conclusion from this graph, but the entropy average seems to perform slightly better than random. Also note that both entropy and loss descending starts above the loss of random which could be due to the lack of class information, but then manages to keep a similar loss score to random. However, the mAP score remains rather unstable which suggests that some information may still be lacking and allowing outliers in the selection. We see that the Untrained model achieves only slightly worse mAP and AP of class "car" than the real baseline does and much faster than the Synthetic baseline. Moreover, we see that loss descending performs well and is stable until the fourth iteration. When it comes to the number of epochs we can conclude that with more data the model needs less training.

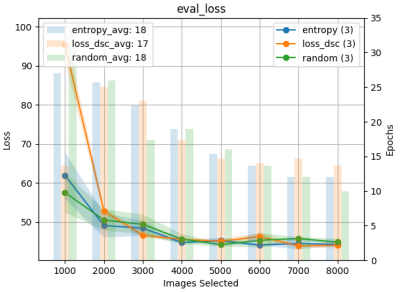


**Figure 4.43** Evaluation loss. The Untrained baseline was tuned and evaluated on the Surveillance dataset.

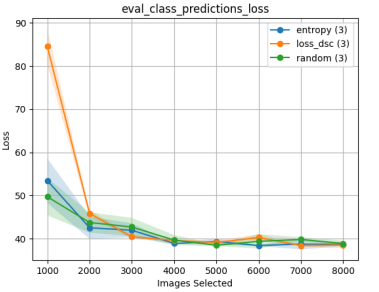


**Figure 4.44** Class prediction loss using the Untrained baseline tuned and evaluated on the Surveillance dataset.

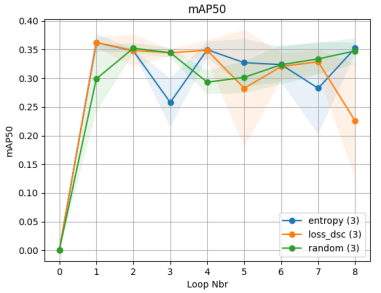


**Figure 4.45** mAP score of the Untrained baseline tuned and evaluated on the Surveillance dataset.

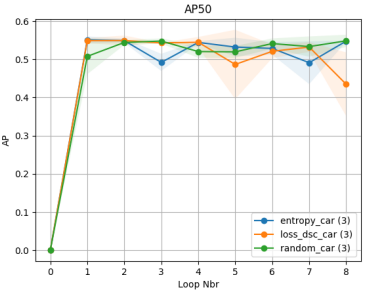


**Figure 4.46** AP score of the class "car" using the Untrained baseline tuned and evaluated on the Surveillance dataset.

***Object Detection Summary.*** Even though these baselines were tuned on the same dataset and used the same hyperparameters, there were many differences in outcome. So what can we take away from this experiment? To summarize, we have that:

- Most importantly we can conclude that the previous training of the baseline does matter. All baselines were tuned on the same dataset yet achieved very different results. Also, a pre-trained model will produce better selections.
- Using the whole selection pool may not be necessary as the effect wears off after about 4000 images.
- A bigger domain shift makes the selection harder and thus the effect of active learning is less.
- With no prior information, that is the untrained model, the random selection performs rather well.

# 5

# Conclusion

In this thesis, we have applied our implementation of the active learning method to mainly three big experiments. Summarized below we have our most important findings from these different experiments.

## 5.1 Parameter Optimization

After having finished our parameter tests for our active learning pipeline, we concluded that for fair comparisons between selection methods, the following parameters should be used.

- Shuffle after each selection step should be on. Failing to do so results in a difficulty gradient among the training batches, which seemingly results in catastrophic forgetting, i.e. quickly forgetting what you've learned because you are training on different data, even though all data is seen once during an epoch.
- The highest scoring (hardest, p0.0) data should be selected. Selecting the non highest scoring (pretty hard, p0.25) data results in faster learning in the beginning, but hinders the accuracy in later phases. There is potential for improvement here.
- Early stopping based on validation loss should be used to avoid over and undertraining. Fixing the training time, gradient steps, or epochs during training may seem fair, but does not measure data quality and results in overtraining issues.
- Lower budget is better, but what exact budget is not very important. Too high of a budget will comparatively limit the learning speed, by physically lowering the resolution of the resulting active learning graph.
- We tried evaluating if reselecting data every loop could be beneficial over accumulative collected data, but saw no improvement. More tests could be done on this topic.

- A base set of randomly selected training data should be used to ensure good performance on active learning methods. Without it, certain scoring methods might not generalize well to the test data and early stop early.

## 5.2 Image Classification

Applying our results from the parameter optimization section, we permuted through all methods and databases defined on the image classification problem and the VGG-16 model and could draw the following conclusions.

- Diversity sampling did not appear to affect accuracy, even on unbalanced datasets, and was therefore not used in the rest of the experiments. More experiments can be done here.
- Uncertainty sampling generally was slightly better in accuracy than random sampling, but not in loss. The different uncertainty methods we analyzed generalized very similar results, and they were not always significantly separable from each other.
- Loss selection performed very well, beating out the other methods of scoring and sampling the data on the dataset without a domain shift. Loss selection is the best active learning scoring method given the VGG-16 network.
- For active learning methods to beat out random sampling, the new data must be similar to the data the backbone is trained on. Introducing a base set might mitigate this effect.
- Not selecting all images may result in higher accuracy. We noticed a worsening of accuracy beyond a certain point given the loss selection. This may indicate pruning your dataset may result in higher accuracy.

## 5.3 Object Detection

The problem of object detection requires further testing, there was not enough time to make as extensive experiments as for the image classification problem. Nevertheless, these initial tests show promise for the application of active learning on the object detection problem. We conclude that:

- The different selection methods applied to the object detection problem show a promising trend that active learning may be of use for model improvement. This is also apparent in the graphs produced for section 4.3.3.
- However, if there is a big domain shift between the new data and the data the model has been previously trained on, the effect of active learning may be less.

- The choice of a baseline is essential for the outcome, a better-trained model is capable of better predictions and in turn better selections.

# 6

# Future Work

Our pipeline code is easily extendable and allows for more complex and modern active learning scoring methods to be implemented without any major modification. One could build on our code to compare state-of-the-art scoring methods on any pre-trained model and a dataset. Especially newer scoring methods on the object detection problem could lead to better AUC accuracy values.

While we have demonstrated a minor improvement over random selection given certain circumstances, more data would most certainly make this difference more pronounced. While we were limited by time and compute power, future work could try to evaluate this on larger datasets like the whole of ImageNet, instead of the smaller test datasets as was done in this report. Given a larger sample size, it may be easier to evaluate if all data is required for the best performance of the model and perhaps would allow for more insight into the effectiveness of dataset pruning.

We saw a clear improvement in not always choosing the most prioritized images for selection. Since this did only produce a gain in the beginning, but hindered later selections, this was something we decided against using. The improvement in the beginning could be the result of a bad selection method, it does fail in its goal of selecting the data most likely to improve model accuracy. This could also be a new way of approaching active learning. Perhaps the scoring method should define the potential gain but not the difficulty of a certain image or data point, and a different method should be used to perform the final selection. Regardless, more work on the topic could be beneficial and interesting.

More work is needed for the problem of object detection. The experiments done were short and small in size due to time constraints, computational power and limited by the few and small subsets available. Many tests did not make it into the report as the results were too unstable to make any conclusions from. The cause of this could be that the models used were not pre-trained enough or to achieve a certain accuracy, more training data and a larger number of epochs to train for would then have been needed. If the models had not learned anything from their base training, then it is easy to learn something from any new data. Meaning, that the model is not "good enough" to select images of better quality, and the selection becomes somewhat random which could explain the large deviations. As for the baseline

test, to see if the different baselines would have done a different selection from each other and which would have produced the best selection we would have to save the selections and cross-examine them on the different baselines. This would mean a minor addition to the code and likely a bigger selection pool would be needed.

Additionally, it would be of interest to do further testing using different mixing strategies to possibly achieve better results when using synthetic data. We have throughout our experiments mainly used the datasets separately, where one dataset was used for pre-training the model, another for the selection pool, and possibly another for validation and testing. As discussed before, this may lead to a too big domain shift for the model to handle or to an unwanted domain shift where the target domain is different from what we would want. By mixing the datasets we could perhaps avoid this, by not forgetting the source domain and learning from the new domain.

Finally, a more complex selection method could further improve results. As specified before, the uncertainty methods mainly look at the detected classes and calculate a score, while the loss descending method uses the average sum of the model-specific classification and location losses. These methods may not hold enough information or prove to be unfair between images since they may contain a different amount of detections compared to the next. A scoring method that considers classification and location (and possibly other) information and where the number of detections is part of the score but not decisive, could then result in better scoring of images.

# 7

# Appendix

As to complete the experiment in Section 4.2.3 we also present the top 1 accuracy graphs on the VGG-16 model when run on the 4 different datasets and all selection methods in Figure 7.1.

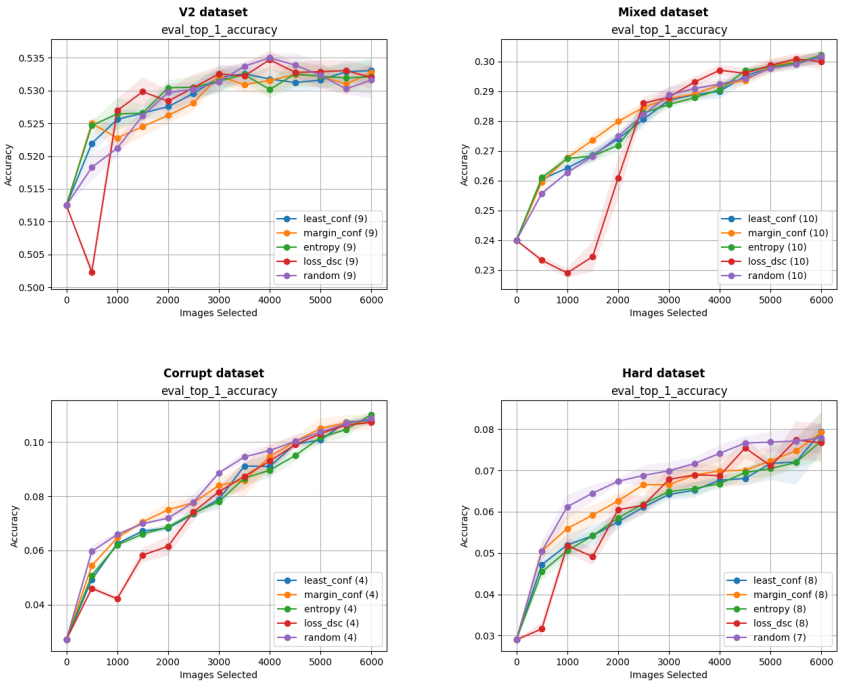


**Figure 7.1** Test accuracy on all datasets and methods

To further demonstrate how the active pipeline works, and give the reader a sense of what images we are working with, we present a typical run on the v2

dataset. The easiest and hardest images per loop per selection method are plotted with their label. While it is hard to draw any conclusions from these Figures, they help visualize what the methods look for.

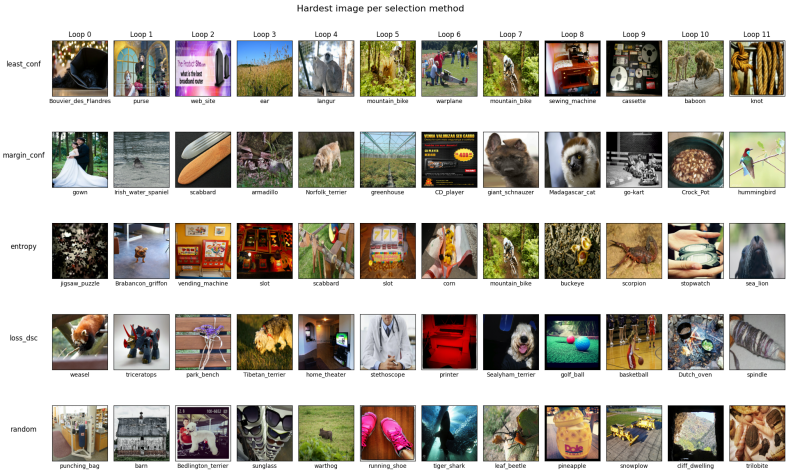


**Figure 7.2** A typical run on the v2 dataset. Most prioritized (hardest) images per selection method

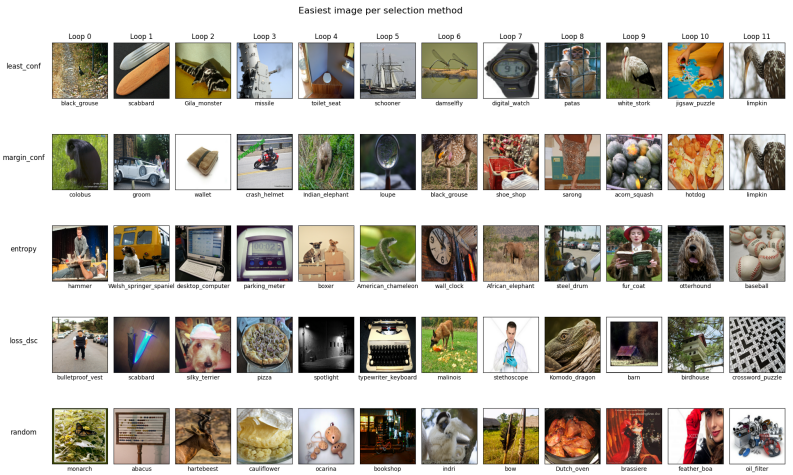


**Figure 7.3** A typical run on the v2 dataset. Least prioritized (easiest) images per selection method

To further illustrate how the model is learning and that the image selection does matter we have Figure 7.4. The synthetic baseline was used together with dataset 2 to produce the graph.

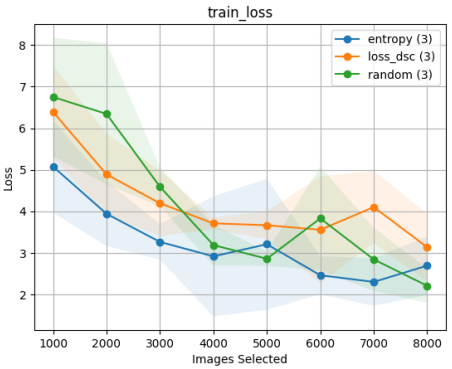


**Figure 7.4** The training loss on synthetic baseline tuned on dataset 2.

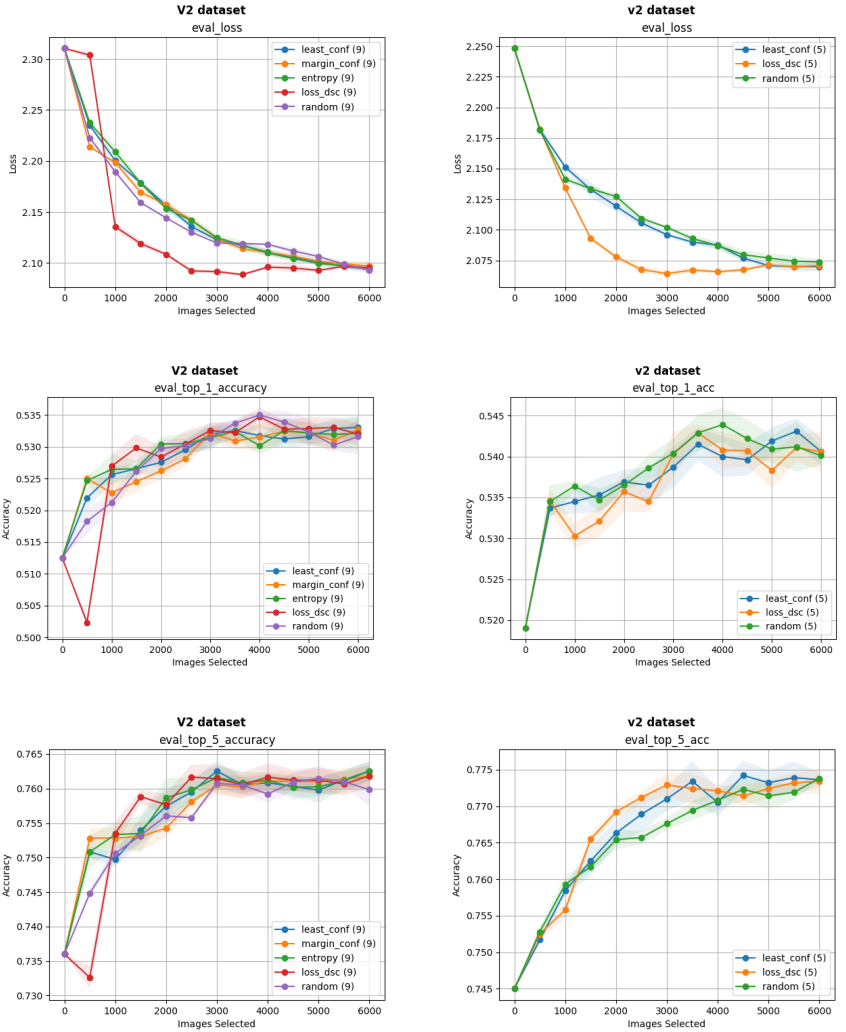


**Figure 7.5** Test accuracy and loss on v2 before and after replacing the first selection step with a randomly selected base pool of data

# Bibliography

Bochinski, E., V. Eiselein, and T. Sikora (2016a). "Training a convolutional neural network for multi-class object detection using solely virtual world data." *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Advanced Video and Signal Based Surveillance (AVSS), 2016 13th IEEE International Conference on*, pp. 278–285. ISSN: 978-1-5090-3811-4.

Bochinski, E., V. Eiselein, and T. Sikora (2016b). "Training a convolutional neural network for multi-class object detection using solely virtual world data." *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Advanced Video and Signal Based Surveillance (AVSS), 2016 13th IEEE International Conference on*, pp. 278–285. ISSN: 978-1-5090-3811-4.

Choi, J., I. Elezi, H.-J. Lee, C. Farabet, and J. M. Alvarez (2021). "Active learning for deep object detection via probabilistic modeling".

Hendrycks, D., S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, D. Song, J. Steinhardt, and J. Gilmer (2020). "The many faces of robustness: a critical analysis of out-of-distribution generalization". *arXiv preprint arXiv:2006.16241*.

Hendrycks, D., K. Zhao, S. Basart, J. Steinhardt, and D. Song (2019). "Natural adversarial examples". *arXiv preprint arXiv:1907.07174*.

Hinterstoisser, S., O. Pauly, H. Heibel, M. Marek, and M. Bokeloh (2019). *An annotation saved is an annotation earned: using fully synthetic training for object instance detection*. arXiv: 1902.09967 [cs.CV].

Jeon, H.-M., L. H. Pham, D. N.-N. Tran, H.-H. Nguyen, and J. W. Jeon (2022). "3d synthetic image training and testing data for autonomous driving." *2022 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Consumer Electronics-Asia (ICCE-Asia), 2022 IEEE International Conference on*, pp. 1–4. ISSN: 978-1-6654-6434-5.

Kao, C.-C., T.-Y. Lee, P. Sen, and M.-Y. Liu (2018). *Localization-aware active learning for object detection*. arXiv: 1801.05124 [cs.CV].

Kwon, Y., J.-H. Won, B. J. Kim, and M. C. Paik (2020). "Uncertainty quantification using bayesian neural networks in classification: application to biomedical image segmentation". *Computational Statistics  Data Analysis* **142**, p. 106816. ISSN: 0167-9473.

Lewis, D. D. and W. A. Gale (1994). "A sequential algorithm for training text classifiers."

Liu, W., D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg (2015). "Ssd: single shot multibox detector."

Ma, C., X. Pan, Q. Ye, F. Tang, W. Dong, and C. Xu (2023). "Crossrectify: leveraging disagreement for semi-supervised object detection". *Pattern Recognition* **137**, p. 109280. ISSN: 0031-3203.

Monarch, R. M. (2021). *Human-in-the-Loop Machine Learning*. Manning Publications. ISBN: 9781617296741.

Nikolenko, S. I. (2021). "Synthetic data for basic computer vision problems". In: *Synthetic Data for Deep Learning*. Springer International Publishing, Cham, pp. 161–194. ISBN: 978-3-030-75178-4. DOI: 10.1007/978-3-030-75178-4_6. URL: https://doi.org/10.1007/978-3-030-75178-4_6.

Nowruzi, F. E., P. Kapoor, D. Kolhatkar, F. A. Hassanat, R. Laganiere, and J. Rebut (2019a). *How much real data do we actually need: analyzing object detection performance using synthetic and real data*. arXiv: 1907.07061 [cs.CV].

Nowruzi, F. E., P. Kapoor, D. Kolhatkar, F. A. Hassanat, R. Laganiere, and J. Rebut (2019b). "How much real data do we actually need: analyzing object detection performance using synthetic and real data."

Rajpura, P. S., H. Bojinov, and R. S. Hegde (2017). *Object detection using deep cnns trained on synthetic images*. arXiv: 1706.06782 [cs.CV].

Recht, B., R. Roelofs, L. Schmidt, and V. Shankar (2019). "Do imagenet classifiers generalize to imagenet?" In: *International Conference on Machine Learning*, pp. 5389–5400.

Sijin, C., Y. Yingyun, and H. Yan (2023). "Semi-supervised active learning for object detection." *Electronics* **12**:375, p. 375. ISSN: 2079-9292.

Simonyan, K. and A. Zisserman (2014). "Very deep convolutional networks for large-scale image recognition."

Simpson, M., N. K. N. Aznan, J. Brennan, P. Watson, P. James, and J. Jonczyk (2022). "Generating synthetic images  data to improve object detection in cctv footage from public transport". In: *2022 IEEE 18th International Conference on e-Science (e-Science)*, pp. 391–392. DOI: 10.1109/eScience55777.2022.00053.

Yoo, D. and I. S. Kweon (2019). *Learning loss for active learning*. arXiv: 1905.03677 [cs.CV].

Yu, W., S. Zhu, T. Yang, and C. Chen (2022). *Consistency-based active learning for object detection*. arXiv: 2103.10374 [cs.CV].

**Lund University**
**Department of Automatic Control**
**Box 118**
**SE-221 00 Lund Sweden**

*Document name*
MASTER'S THESIS

*Date of issue*
June 2023

*Document Number*
TFRT-6198

*Author(s)*
Linnea Allander
Torben Nordtorp

*Supervisor*
Joel Sjöbom, Axis Communications, Sweden
Emma Person, Axis Communication, Sweden
Pontus Giselsson, Dept. of Automatic Control, Lund University, Sweden
Karl-Erik Årzen, Dept. of Automatic Control, Lund University, Sweden (examiner)

*Title and subtitle*
Qualitative Data Selection with Active Learning

*Abstract*

In this work, an active learning pipeline is presented that allows for comparative tests between mainly three different types of data scoring methods: uncertainty selection, diversity selection, and perfect loss selection. Tests and parameter sweeps indicate that hyperparameters like reshuffling and early stopping are required to ensure fair comparisons. We then apply our pipeline to test some naive scoring methods on two Computer Vision problems: Image Detection, and Object Detection. We conclude that comparing these scoring methods against random selection gives minor evaluation loss improvements on our datasets in the right circumstances, and discuss other aspects of data quality along the way. The repository for our pipeline can be found on GitHub at https://github.com/voxlz/Qualitative-Data-Selection-with-Active-Learning.

*Keywords*

*Classification system and/or index terms (if any)*

*Supplementary bibliographical information*

*ISSN and key title*
0280-5316

*ISBN*

*Language*
English

*Number of pages*
1-56

*Recipient's notes*

*Security classification*

http://www.control.lth.se/publications/