A Practical Evaluation of Active Learning Approaches for Object Detection

Jan Schneegans[1], Maarten Bieshaar[2], and Bernhard Sick[1]

[1] Intelligent Embedded Systems, University of Kassel, Kassel, Germany
{jschneegans, bsick}@uni-kassel.de
[2] Robert Bosch GmbH, Corporate Research, Hildesheim, Germany
maarten.bieshaar@de.bosch.com

Abstract. The supervised training of deep learning models typically requires vast amounts of annotated data. With active learning, the annotation process can be made much more efficient by intelligently selecting the most valuable batches of samples to annotate and train on. Those samples are selected based on their utility regarding the training algorithm. In this work, we examine a wide range of such selection criteria for the task of object detection as performed by the widely applied Faster R-CNN model. We focus on the large and diverse BDD100K autonomous driving dataset, paying special attention to evaluate the model's performance regarding the dataset's meta information. Furthermore, we distinguish between approaches that select samples based on aleatoric or epistemic uncertainty. A selection of evaluation measures that cover specific error sources and the overall model performance suggests that there is little difference between the individual active learning approaches, even in regards to their specialized focus on different model parts and the object detection tasks of localization and classification. We conclude with a detailed discussion of the implied mechanisms regarding the active learning approaches that seem to affect model performances.

1 Introduction

Data annotation is costly both in time and resources (human and computational). The theoretical advantages of using active learning lie in a more efficient data annotation process by intelligently selecting a subset of samples that is thought to be most useful to train the machine learning model on. In this work, we perform a practical examination of active learning strategies, gaining insights into why certain approaches perform better than others regarding the task of 2D object detection. This is done on the very large BDD100K [1] dataset, which is one of the most diverse autonomous driving datasets in terms of scenarios, weather, uncommon objects, and other attributes, applying the popular Faster R-CNN model [2]. We describe the active learning process as a cycle of iteratively selecting a batch of samples to be annotated and training the Faster R-CNN model on the annotated portion of the data, cf. Figure 1. One batch consists of a set of images, each image containing one or more objects. The selection process consists of three parts: i) an utility function which estimates the

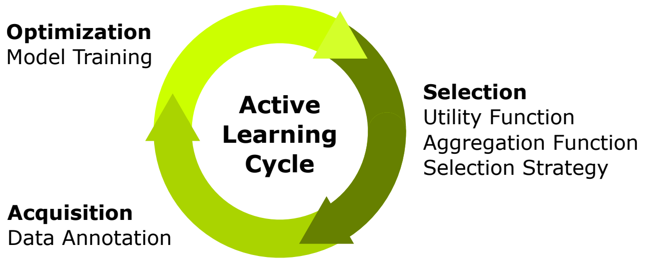

Fig. 1. The typical active learning cycle consisting of data selection, acquisition of annotations, and model training.

usefulness of each object, ii) an aggregation functions, which aggregates the usefulness for a complete image, and iii) a selection strategy, which selects the set of images deemed most useful. The examined variety of active learning strategies are based on the sub-tasks performed by the Faster R-CNN model, namely the separation of the annotated objects and the background, and the precise classification and localization of those objects. Furthermore, we consider and compare utility functions based on aleatoric and epistemic measures of uncertainty facilitated by Monte-Carlo dropout [3]. Multiple aggregation functions, e.g., mean and quantiles, are tested for each utility function to summarize the utilities over a whole image. We restrict the selection strategy to simply select the top k samples, i.e., images, with the highest utilities as aggregated from the individual object utilities for each image.

Our goal is to evaluate the practicality and benefits of utilizing active learning strategies in the training of large object detection models and to provide practical insights into the design of the corresponding machine learning pipeline. We present a novel utility function facilitated by the box predictions and their intersection-over-union (IOU) and experiment with approaches based on the different sub-tasks executed by the object detection model, i.e. utilizing the objectness, class, and box predictions of the model. We see a lack of a thorough, realistic and foremost practical evaluation of various active learning approaches for the task of object detection. In this work, we aim to close this gap and compare the actual annotation cost of each active learning approach and discuss the practicality and performance given the required computational effort. Moreover, we can get further insights into why certain active learning approaches perform better than others by examining the selection of meta-attributes, e.g. weather, time-of-day, scene, etc., of the BDD100K dataset.

In the following Section 2, we give an overview of active learning approaches as a whole and those specifically aimed at the task of object detection. Section 3 introduces our methodology, including the model setup and a thorough introduction of the examined active learning strategies. In Section 4, we provide further information regarding the dataset, experimental design, and evaluation measures. Section 5 discusses the experimental results, after which we summarize our findings and presents directions towards future work in Section 6.

2 Related Work

Active learning methods deal with the selection of data samples for annotation and subsequent model training. Much published work on the topic of active learning is concerned with proposing specific utility functions or selection criteria. Often these are based on a Bayesian Neural Network approach or Monte-Carlo sampling approaches through dropout [4]. Popular examples are uncertainty sampling [5], and entropy based ones [6], e.g. BALD [7] and Batch-BALD [8]. Siddahnt et al. [9] present a large-scale empirical study on deep active learning approaches, concluding that BALD can significantly outperform other approaches, using uncertainty estimates provided either by Dropout or Bayes-by-Backprop. Most techniques are made for classification tasks and only recently the spectrum of approaches was widened to encompass regression tasks [10,11,12,13]. For a survey on further aspects to consider in active learning, e.g. cost types and annotator performance, see [14]. As we do not consider any temporal information in the object detection tasks, we do not consider stream-based active learning methods, but instead focus on a variety of pool-based utility functions building on uncertainty estimation. Methods for query-synthesis, i.e. the generation of novel sample to annotate, are also beyond the scope of this work.

Advancing active learning methodologies towards more complex prediction tasks, e.g., object detection and localization, requires more sophisticated active learning approaches. Brust et al. [15] select images in an uncertainty-based approach using bounding box and class metrics. In [16], the uncertainties of both classification and bounding box predictions are utilized, as well. Roy et al. [17] use a query by committee approach and the disagreement between the convolutional layers in the object detector backbone. [18] investigates continual learning aspects of an ensemble-based method incorporating both classification and localization aspects for 2D and 3D object detection. Multiple Instance Active Learning for Object Detection [19] adapts an adversarial training procedure to select informative images for detector training by observing instance-level uncertainty, although, this approach implicitly assumes that there is a dominating object in each image, hence it can attach a single label to each images (as in image classification). Haussmann et al. [20] evaluate the use of active learning on a large scale object detection dataset for autonomous driving, although, with a different choice of models, active learning strategies, and dataset.

The Faster R-CNN model is one of the most widely used object detection model due its good performance and many readily available implementations. Since the original publication of the Faster R-CNN model many improvements to its architectural design were proposed [21]. Since most of these approaches add more complexity to the models with minuscule performance improvements, we only utilize an additional feature pyramid network [22], whose multi-scale feature maps will take part in our active learning approaches. Aghdam et al. [23] perform active learning for object detection by aggregating different pixel-level scores on the output of a convolutional neural network, which bears resemblance to our application of utility functions on the objectness maps predicted by the region proposal network inside the Faster R-CNN.

3 Methodology

This section describes the required preliminaries and individual parts of the applied machine learning model and introduces the examined active learning approaches. First, we briefly describe the applied object detection model, i.e. a modified Faster R-CNN, which we augment with dropout layers to perform uncertainty estimations, i.e. Monte-Carlo Dropout. Then the active learning approaches, consisting of individual utility functions, aggregation functions, and selection strategies are introduced.

3.1 Faster R-CNN

The Faster R-CNN model is one of the most widely used object detection models due to it's reliable performance and readily available implementations; but due to it's two-stage approach it is also one of the slowest. Accordingly, incorporating active learning in the training pipeline is a natural match to reduce training times. The Faster R-CNN consists of three main parts: a ResNet [24] *backbone*, a *region proposal network* (RPN), and the *classification and regression heads*. Fig. 2 shows the three components of the model, the augmentation via the dropout layers, as well as exemplary predictions for each sub-task utilized in the active learning approaches.

The backbone used for features extraction consists of a ResNet50 as implemented by the torchvision framework [25], followed by a feature pyramid network (FPN) [22] to better handle objects of different scales. The FPN extracts features at five different scales and three aspect ratios (1:1, 1:2, and 2:1), which are all input to the region proposal network.

The region proposal network consists of convolutional layers and performs a foreground and background classification and an initial rough localization of potential objects. Due to the use of the FPN, this is performed at five scales (32^2, 64^2, 128^2, 256^2, and 512^2 pixels) and three aspect ratios, resulting in a total of 15 *objectness* maps containing pixel wise binary classifications. The objectness is treated as one of three model outputs, which are further utilized in the active learning approaches.

Based on the binary classification performed by the RPN, a set of highest scoring object proposals is selected. Together with the features extracted by the backbone the selected object proposals are subsequently processed in separate heads for the final object classification and localization, i.e. box prediction. Those are the second and third model outputs on which the active learning approaches are applied.

The dimensions of the last few layers of the Faster R-CNN need to be adjusted to the specific learning problem posed by the dataset, i.e. the thirteen object classes considered. The dimensions of the final fully connected layers are adjusted accordingly. We start each experiment on a pretrained Faster R-CNN model on the COCO [26] object detection dataset to facilitate faster learning.

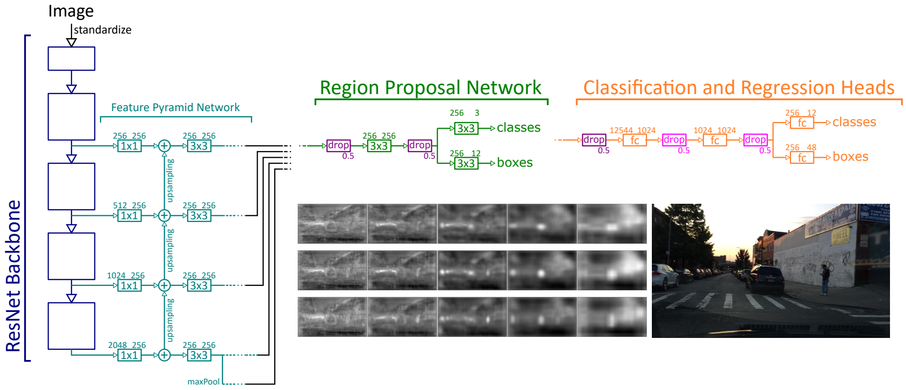

Fig. 2. Architecture of the Faster R-CNN model. The added dropout layers required to produce uncertainty estimates via Monte-Carlo dropout are shown in purple. The RPN is applied to all five scale dimensions output by the FPN individually. Exemplary predictions of the objectness (rows: aspect ratio, columns: scales), class and box predictions are shown. (We will colormap the objectness maps and depict matching model predictions in the final version)

Uncertainty Estimation Most active learning approaches are based on uncertainty estimates provided by a probabilistic model or sampled from an augmented model [27]. Typically a distinction between aleatoric and epistemic uncertainty is done [28,29]. Aleatoric uncertainty measures the uncertainty inherent in the data, produced by, e.g. noise, or in regards to the application it might also encompass sources of unpredictability such as motion blur or dirty lenses. Epistemic uncertainty measures the uncertainty of the model itself about its predictions and is typically harder to compute. To perform active learning this second kind of uncertainty is more useful, because one desires to select those samples, which the model struggles with, given the assumption that those samples provide the most benefit during training. The (pseudo-) probabilities produced by a neural network do not capture the epistemic uncertainty [28], therefore, the model architecture needs to be extended via an appropriate uncertainty estimation technique. We will utilize Monte-Carlo dropout as proposed in [4] and add respective dropout layers to the Faster R-CNN. More specifically they are added to the convolutional layers of the RPN and the classification and regression heads. Dropout refers to the CNN specific 2D variant that zeros-out entire channels, or in abstraction complete features. The model output can thus be sampled via multiple forward passes to estimate the epistemic uncertainty about the predictions. We draw 10 samples in each forward pass to maintain a reasonable inference time during the active learning cycles. The dropout layers are also kept active in those approaches that do not rely on the epistemic uncertainty estimates to avoid biasing the results, because we observed slightly lower performance while utilizing dropout, and we want to investigate the performance differences based on the utility functions and not due to adding dropout.

3.2 Active Learning Approaches for Object Detection

The term active learning encompasses strategies to select a subset of a given dataset with the goal of reducing costs in data annotation and model training. It does so in an iterative process of selecting and annotating data, and training on the available annotated data. We term one of those iterations as a *cycle*. Since we are working with the already annotated BDD100K dataset the annotation process is simulated by taking the annotations of the selected data into account.

For the application of object detection an active learning strategy consists of three main parts: a *utility function* that estimates the utility of an object or image, an *aggregation function* that aggregates the utilities of all objects in an image, and a *selection strategy* that selects the k most useful images.

Accordingly, one cycle consists of the model predicting object locations and classes, the application of the utility function, the application of the aggregations function, the application of the selection strategy, annotation of the selected data, i.e. moving data from the unlabeled dataset to the labeled dataset, retraining of the model based on the labeled dataset, and checking of a stopping criteria.

The initial condition (zeroth cycle) consist of an unlabeled dataset and the newly initialized (pretrained) model. Stopping criteria can be based on the amount of data that can be annotated, which might be restricted by available resources, e.g. financial budget, or based on the model performance, e.g., when a desired performance is reached or when the training saturates. Since we do not explicitly consider a fixed budget in the utility functions, we simply set the number of active learning cycles to 30 (based on the model convergence during the experiments) and compare different approaches based on the model performances over those iterations. The design of cost-sensitive utility functions, which explicitly consider the sample costs during estimation of their utility is still an open research-topic.

Utility Functions Given the model predictions about the object classes and locations, a utility function ascribes a usefulness to each object in the unlabeled dataset. Additionally, in case of the objectness predictions for an image, i.e. the feature maps showing the foreground-background classifications performed by the RPN, the utility of the entire image can be estimated directly. We further consider an approach utilizing all three predictions, i.e. objectness, class and location, combining multiple utility functions. The approaches based on the object classes consist of the following measures:

The normalized entropy provides a measure of the lack of model confidence based on the class predictions. It is obtained through normalization of the Shannon entropy [6] H over the maximum possible entropy $log(K)$, which is reached by a uniform distribution. Formally it is defined as

$$\eta(\hat{\mathbf{p}}) = \frac{H}{H_{max}} = -\sum_{k=1}^{K} \frac{\hat{p}_k \log(\hat{p}_k)}{\log(K)}, \tag{1}$$

where $\hat{\mathbf{p}}$ are the class predictions and K the number of classes. The predicted class probabilities $\hat{p}_k = \sigma_K(\mathbf{x})_c$ are given by applying the softmax function to the logits, i.e. the classification output of the model. The *normalized entropy* produces a high value when there is a strong disagreement between the different classes, i.e., when the distribution over the predicted classes approaches uniformity. Contrary, this entropy measure will be low, when there is a single class holding most of the distribution's mass, i.e., when the model in confident.

BALD [7], is also an entropy based measure. It aims to select samples that are expected to maximize the information gained about the model parameters [3]. BALD specifically utilizes the epistemic uncertainty of the model by sampling the model output via the applied Monte-Carlo dropout technique. The sampled predictions are clustered according to their location via an agglomerative clustering based on a distance threshold of 0.5 regarding their box IOU. The BALD utility function can then be applied to each cluster, i.e. each predicted object. We utilize a modified version of the BALD utility function that is normalized facilitating an unbiased combination of utility functions. Given a dataset $\mathcal{D}$ and model $\mathcal{M}$ with parameters ω_t as one of T random dropout configurations, the computationally tractable approximation of the utility function used during the experiments is formally given by

$$\begin{aligned}\alpha(\mathbf{x}|w) = & \frac{1}{log(K)} \left(-\sum_k \left[\left(\frac{1}{T} \sum_t \hat{p}_k^t \right) log \left(\frac{1}{T} \sum_t \hat{p}_k^t \right) \right] \right. \\ & \left. + \frac{1}{T} \sum_{k,t} \hat{p}_k^t \, log(\hat{p}_k^t) \right)\end{aligned} \tag{2}$$

where $\hat{p}_k^t$ is the probability of input $\mathbf{x}$ predicted by the model with parameters ω_t to take on class k, i.e., $\hat{p}_k^t = \sigma_K(\mathcal{M}_{\omega_t}(\mathbf{x}))_k$. p_k^t is given by the class predictions of the cluster. As with the entropy measure above, we normalize the BALD equation by the maximal possible entropy $log(K)$.

We will also apply both the normalized entropy and BALD utility functions to the objectness maps produced by the RPN.

As a utility function based on the object box regression, we propose a novel measure based on the Intersection-Over-Union (IOU). Similar to BALD we first need to cluster the proposed boxes per object before we calculate the IOU of each box within a cluster to the cluster mean, i.e., the mean box of the cluster. Subsequently those IOU values are averaged. Because we require the utility function to signify higher uncertainty with higher values we invert the expression by calculating 1 - the mean IOU. The *expected IOU* (eIOU) is thus formalized as

$$\bar{\mathbf{b}}_c = \frac{1}{|\mathcal{B}_c|} \sum_{\mathbf{b} \in \mathcal{B}_c} \mathbf{b} \tag{3}$$

$$\text{eIOU}(\mathcal{B}_c) = 1 - \frac{1}{|\mathcal{B}_c|} \sum_{\mathbf{b} \in \mathcal{B}_c} \text{IOU}(\mathbf{b}, \bar{\mathbf{b}}_c), \tag{4}$$

where $\mathcal{B}_c$ is the set of predicted boxes per cluster c, and $\bar{\mathbf{b}}_c$ the mean of the cluster. The expected IOU is normalized by definition, due to the IOU being normalized; this is advantageous compared to using the total or generalized variance of a cluster, because it is not influenced by the position and size of the object proposals. For the utility function should not be biased by those properties.

In order to evaluate a utility function utilizing all three model outputs, i.e., objectness, classes and boxes, we define a combined measure consisting of the normalized BALD approach applied to the objectness and the class, together with the eIOU.

The presented selection of utility functions comprise the most general and widely applied approaches based on estimated model uncertainties with the addition of a similarly inspired box-based version, i.e. the expected IOU. Having defined the utility functions, we can measure the utility per predicted object, or pixel in case of the objectness maps.

Aggregation Functions An aggregation function summarizes the output produced by a utility function to describe the utility of a complete sample, i.e. an entire image.

Intuitively the mean over the utility function output provides a measure of the average utility in annotating a certain sample. The median is not a good option as it is not influenced by outliers, e.g. objects the model is especially uncertain about, but we want the utility measure to be explicitly influenced by those parts of the image, assuming that these outliers are particularly interesting and useful.

Accordingly, applying the max function gives further priority to especially high values of the utilities produced by the utility function.

Although, simply applying the max aggregation function on the utility functions applied to the objectness maps would not work well due to fact that almost always the objectness maps contains maximum values of 1, thus every image would be ascribed the same maximum utility. To solve this issue we ignore those very high values by only considering the 95th and 99th percentiles.

Selection Strategies The selection strategy decides which of the samples from the unlabeled dataset are selected for annotation, given the aggregated utilities inferred through the application of a utility and aggregation function. We want to train the model on those samples that are deemed most useful, naturally, the selection strategy will simply consist of the max function over all unlabeled samples, selecting those samples with the maximum utility as aggregate per image. Depending on the utilized utility functions those samples are also the ones the model is most uncertain about.

4 Evaluation Methodology

This section details preliminary information regarding the dataset, the experimental setup, and the applied evaluation measures necessary to investigate the object detection as well as the active learning performances.

Dataset The experiments are performed on the BDD100K dataset, which is is one of the largest object detection datasets in the autonomous driving domain. It contains a variety of scenarios, sceneries, and annotated objects. Due to varying conditions such as time-of-day, weather, as well as noise, motion blur, and lens flares, the dataset poses a challenge towards current machine learning models. This naturally befits the use of active learning techniques to select different and useful samples, with the goal of reducing both annotation costs and training time.

There are five object categories with overall 13 classes: bike: bicycle, motorcycle; person: pedestrian, rider; vehicle: bus, car, truck; distractor: other person, other vehicle, trailer, train; signal: traffic light, traffic sign.

We perform experiments on all of the 13 classes as well as an easier subset of three classes summarized by bike, person, and vehicle, which supported the results on the larger set of classes. Additionally, we investigate the connection between the available meta-attributes with the active learning approaches to see if the approaches display certain preferences in selecting data samples in regards to these attributes: weather: clear, foggy, overcast, partly cloudy, rainy, snowy, undefined; scene: city street, gas stations, highway, parking lot, residential, tunnel, undefined; time-of-day: dawn/dusk, daytime, night, undefined; occluded: False, True; truncated: False, True.

Experimental Design Our goal is to evaluate the individual active learning approaches, consisting of combinations of the introduced acquisitions functions, aggregation functions, and selection strategy. For each of those combination we train a Faster R-CNN model for 30 active learning cycles. The pretrained model gets trained from scratch in each cycle as to not overfit on the data annotated the earliest, which we observed upon initial experimentation. Each experiment is performed twice, to make sure the experimental results and discussion thereof are reliable.

The BDD100K dataset contains 100.000 images, although, the annotations of the original test set are not available so we took the last 10.000 images from the training set to form an annotated test set. The splits are illustrated in Figure 3 with the original validation set, containing 10k images.

¡Through preliminary experiments the main hyper-parameters were determined. Those include: learning rate = 1e-5, batch size = 20, epochs = 10 (per cycle), and Mish activation functions. We utilize the Ranger optimizer [30] for faster convergence, which incorporates AdaBelief, RAdam, Lookahead, and Gradient Centralization. An acquisition size of 512, i.e. the number of acquired samples after each cycle, was determined to balance a reasonable annotation cost

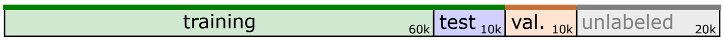

Fig. 3. Training, test, and validation splits of the BDD100K dataset.

and training progress on the growing annotated dataset. This means the final models (at cycle 30) were trained on $30 * 512 = 15360$, which corresponds to 25.6% of the training data.

During training we utilize image augmentation by alteration of the brightness and contrast, or by adding Gaussian noise. The augmentations are applied with a random probability, order, and intensity (within previously determined bounds).

Performance Measures We distinguish between two kinds of measures that evaluate the object detection performance and further aid the investigation of the active learning approaches, respectively.

The most commonly applied evaluation measure for the task of object detection is the *mean Average Precision* (mAP). It describes the area under the precision-recall curve derived from the statistics of the model predictions. Since the mAP score only provides a single number, we additionally apply separate measures to evaluate each of 6 possible kinds of errors: class error, location error, class and location error, duplicate predictions, background prediction, missed objects, as proposed by [31]. A predicted box is considered correct if its IOU with the ground-truth box is higher than 0.5. To be able to compare class and location errors independently, an additional lower IOU threshold is needed, so that if a box is in the wrong location the classes can still be reasonably compared. This lower IOU threshold is set to 0.1, as suggested by [31]. Arguably, predictions recognized as class or location errors could also be counted as duplicate predictions if they can be matched to a ground-truth box, but since we want to count every prediction only once, only otherwise correct predictions count towards duplicate errors. By investigating the individual error sources, we can for example examine if approaches based on the predicted classes produce less classification errors, or if approaches based on the box predictions perform better in the localization sub-task.

To evaluate the performance of the active learning strategies, we are not only interested in the final model performances after 30 cycles, but also in the annotation costs (as measured by the number of annotated objects), which are often neglected in the current literature, and in the learning behavior over all active learning cycles. Notably, while the same number of samples, i.e., images, is selected in each cycle, different amounts of objects in the selected images lead to different annotation costs of the active learning approaches. Splitting the performance evaluation according to the available attributes is very useful to investigate whether a model performs better on samples that are considered more difficult, e.g., when time-of-day is night. Another assumption often made is that difficult samples are most useful to train on and that active learning approaches based on uncertainty measures are supposed to select those difficult samples. To investigate if this is the case, we sort all samples by the average mAP score over all models and compare them with the selection by the models.

5 Results

This section presents the results of the experiments and discusses the various insights into the applied active learning approaches. This encompasses three main parts: the learning behavior, the final model performances, and the acquisition characteristics.

The different approaches are each abbreviated by three letters, indicating: the type of sub-task predictions used for the active learning approach, the utility function, and the aggregation function. For example, `cbm` is the approach comprised of the BALD utility function applied to the predicted object classes and aggregated by the mean aggregation function. See all abbreviations in Table 1. `rdm` denotes random sampling, and `all` stands for the approach that utilizes all three kinds of predictions, applying the normalized BALD utility function to the class and objectness predictions, and the eIOU to the boxes.

Table 1. Active learning approaches abbreviations.

prediction		utility		aggregation	
c	class	e	entropy	m	mean
b	box	b	BALD	x	max
o	objectness	i	eIOU	5	95-percentile
a	all			9	99-percentile

We first compare the various active learning approaches by considering their performance on the test set. Fig. 4 shows the mAP performance for each cycle, averaged over both random seeds. Remember that each cycle adds 512 images to the labeled portion of the training set and in each cycle the models are trained anew. With this in mind we can see that training on more data does improve the model performances consistently for all approaches. Although, the convergent behavior is similar for all approaches, i.e. no approach provides considerably faster learning based on the selected data, and they all end up with around the same performance after 30 cycles. We performed the same set of experiments with a reduced set of annotation containing only three classes (vehicle, pedestrian, and cyclist), resulting in very much the same performance and learning behavior.

Second, we compare the performance of the trained models after 30 cycles. Fig. 5 shows the performance of each approach sorted by the mAP score on the test set, and the different error sources according to the TIDE measures [31]. Again, the numbers represent the average over both random seeds. Generally, we observe no significant differences between the active learning approaches. Most perform slightly better than random sampling. The approaches based on the objectness seem to perform best, for which the reason will be explored in the next subsection. There is no clear winner between the utility functions or the aggregation function. The performance of models trained via utility functions based on class or box predictions, do not show a clear correlation for their specific sub-tasks, as shown by the detailed evaluation of the different kinds of errors.

To investigate which kind of images each active learning approach acquires, we compare the selection for each model and for each cycle for the following object- and image-level attributes: class label, box size (5 bins, logarithmic),

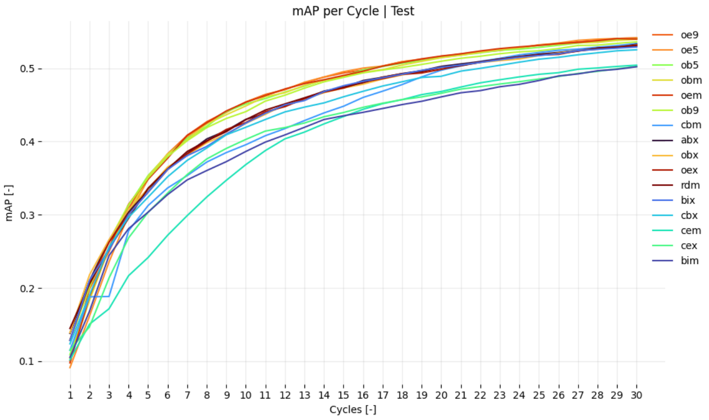

Fig. 4. Learning curves showing the mAP performances per cycle on the test set. All approaches show similar learning behavior, although, *cem*, *cex*, and *bim* perform slightly worse than the other approaches.

and the attributes presented by the BDD100K dataset. Additionally, we will assert if the approaches select especially difficult samples, as measured by the average mAP score of the models.

Fig. 6 shows the distribution of attributes in the training set (black dashed) from which the active learning approaches iteratively select a subset to train on. The attribute distributions of the selected subsets are shown for each cycle, whereby lower cycles are blue and higher cycles are red. One would assume, that the approaches should select a higher proportion of attributes that occur less often in the dataset, given the assumption that those samples are probably more difficult for the models. If this were the case, one should see a balancing between the individual instances of the attributes, respectively.

Regarding the label distributions, the approaches mostly adhere to the training distributions, with the exception of *bim* and *cbm*, which even exaggerate the label imbalances by selecting many images with cars in them. The approaches based on the objectness maps show more promising behavior by selecting less cars and more pedestrians, with the exception of the approaches utilizing the max aggregation function. The reason being, that they practically reduce to random random sampling due to the effect explained in Sec. 3.2. The box size selection is very consistent between every approach, oversampling small boxes. The weather selection looks very similar to the label attribute, with the *cem* approach also oversampling the most prominent weather instance (*clear*). Interestingly, the scene attribute shows an inverse behavior compared to the label and weather distributions. *bim* and *cbm* select more images showing *residential* and fewer showing *city street*. *cem* and *cex* have a similar preference towards *highway* scenes. Contrary to the label selection, the objectness based approaches oversample *city street* scenes and avoid *highway* scenes; we will discover why when we look at the number of acquired objects. Regarding the time-of-day, most ap-

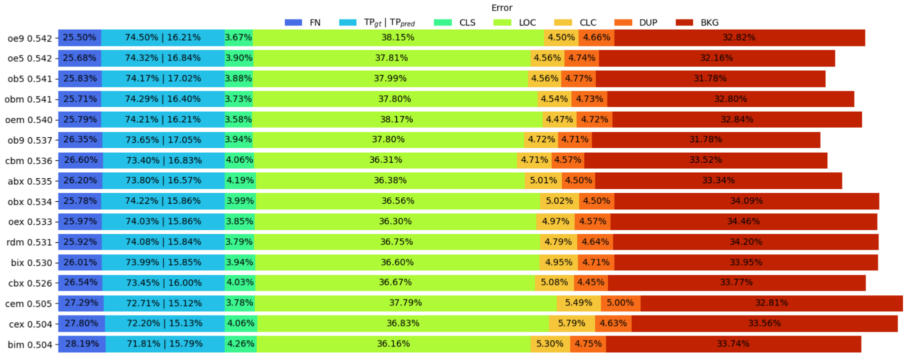

Fig. 5. Model performance after 30 cycles, sorted by the mAP score on the left. FN and TP$_{\text{gt}}$ are in proportion to the number of ground truth, all else in proportion to the number of predictions. There is no score threshold applied to the predictions, which is why the percentage TP$_{\text{pred}}$ seems relatively low. FN: false negatives, TP: true positives, CLS: class error, LOC: location error, CLS: class and location error, DUP: duplicate predictions, BKG: false positives/background predictions.

proaches exaggerate the day- and night-time imbalance in the distribution by selecting primarily day-time images, while some seem to prefer night-time images (*bim*, *cem*). The distributions of the occlusion and truncation attributes show no significant behavior, except for some approaches, apparent in the figure.

Contrary to the assumption, we generally observe little balancing and the attribute distributions of the selected images mostly vary around the training data distribution. We make the general observation that if there are deviations from the training distributions, they are more pronounced in the early cycles (blue), and the selection distributions are closing in on the training distributions towards later cycles (red), often matching them in the final cycles. For the attribute instances with very few sample we barely see any selection; enabling the approaches to oversample during sample selection might help in those cases. The attribute distributions of the data selected by random sampling (*rdm*, last row) is consistent with the overall data distribution, as expected.

Another kind of attribute often implicitly talked about is the difficulty of the samples, based on the assumption that more difficult samples are particularly useful for model training and should thus be selected by active learning approaches; especially by uncertainty based ones if we relate uncertainty to difficulty. To check this assumption we sorted all images in the training set according to their average mAP score over all models to estimate their difficulty.

Fig. 7 shows the four kinds of behaviors observed when we look at the image selections in each cycle over the mAP score. To create the depiction all 60k images in the training set, sorted by their average mAP score, were binned into 128 bins, shown along the x-axis (low to high mAP, from left to right). This includes the selection from both random seeds. The rows depict the cycles (early to late, from bottom to top). The approaches based on the class predictions and the BALD utility function, as well as the proposed box based eIOU approach,

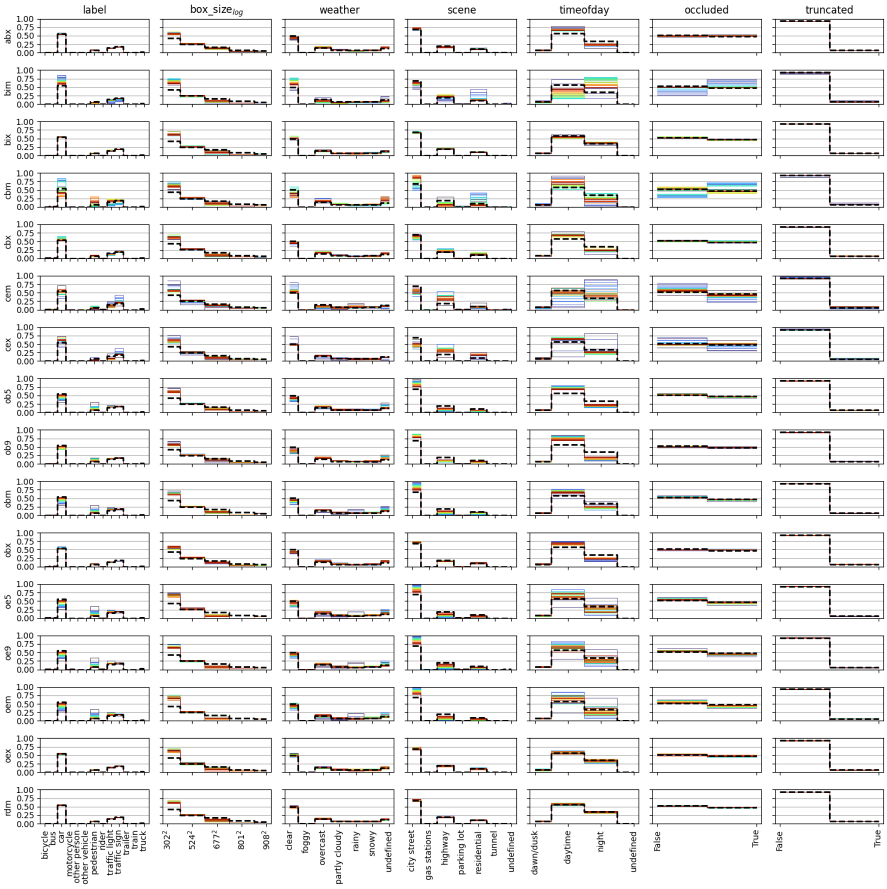

Fig. 6. Distributions of selected object and class level attributes compared to the training set distributions (black, dashed). For each active learning approach and attribute the distributions for all cycles and both random seeds are shown. One cycles consists of the attribute distribution of the 512 acquired images or the objects contained therein. Early to late cycles are colored blue to red.

select difficult examples in the earlier cycles, which then diffuses towards the region of average difficulty after the first few cycles. *cbx* maintained a slightly stronger preference towards high mAP scores throughout all cycles class-based approaches utilizing the normalized entropy utility function have a very strong tendency to sample very easy or very difficult images, as estimated by the mAP score. Here the focus tapers off towards later cycles as well. The approaches based on the objectness maps show a more independent distributions over the mAP score, with a slight tendency to not sample very difficult images. There is barely any variation over the cycles. Lastly, some approaches show similar random behavior as random sampling. Notably, all of those approaches use the

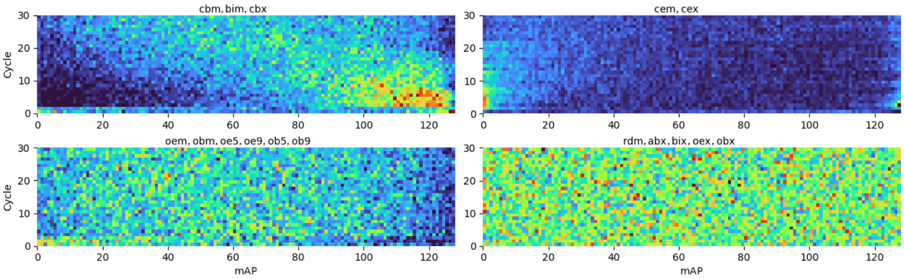

Fig. 7. The four kinds of selection behaviors expressed by the active learning approaches, in regards to the difficulty of the available images, estimated by the average mAP score of each image. It shows a two-dimensional histogram indicating the number of images in that region of the sorted mAP score. The mAP score is sorted from left to right, i.e. less difficult to most difficult.

max aggregation function. As already mentioned before, these approaches assign the same maximum utility to most images, leading to a random selection.

Finally, we compare the performance of each approach with the annotation costs of their acquired images. For this shows the actual practical applicability of the approaches. The performance is estimated by the mAP score of the trained models after 30 cycles. The annotation costs are estimated by the number of objects contained in the acquired images, because annotation companies usually calculate the annotation effort and costs per object. Fig. 8 shows both performance and the annotation costs for each approach, relative to random sampling. The approaches are sorted by their performance.

We observe a strong correlation between annotations costs, i.e. the number of objects, with the model performance. If we allow for a small decrease in performance compared to random sampling, the annotation costs can be reduced immensely, depending on the approach. In contrast, to achieve a slight increase in performance the annotation costs rise by about 40%. Notably, we see that the objectness based approaches consistently acquire images with more objects.

This is consistent with our previous observations that the objectness based approaches preferably select *city street* scenes with many *pedestrian* labels compared to the usually high number of *car* labels, during *daytime*. Otherwise, the results do not seem to correlate with the results depicted in Fig. 6 or Fig. 7. If we compare the annotation costs with the more detailed TIDE scores, we notice that the model performance correlates well with the overall amount of errors.

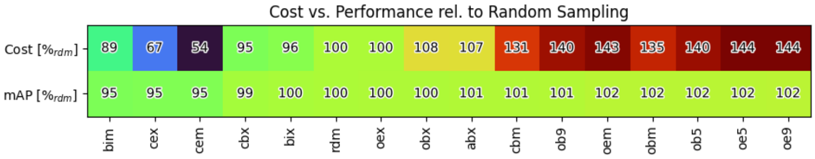

Fig. 8. cost vs performance rel. to random sampling

Overall, a high number of objects is beneficial to the model performance, and the objectness based approaches represent a proxy to find images containing many objects. We attribute it to the observation that the objectness maps produce high values around object edges, because there the uncertainty, whether a pixel in the objectness map corresponds to and object, is very high. Accordingly, the more objects are in the image, the more edges there will be, and the higher the aggregated utilities are. This naturally leads to the question, whether simply approaches like an edge detector could provide a good basis for active learning strategies, reducing the computationally effort by a large margin. To further note: random sampling takes drastically less compute effort, because one does not need to estimate the uncertainties, e.g. by sampling a model multiple times.

6 Conclusion

We applied a variety of active learning approaches to the task of object detection on a large and varied autonomous driving dataset. The approaches, comprising combinations of multiple utility functions and aggregations functions, utilized different kinds of model predictions based on the sub-tasks performed by the Faster R-CNN model. Overall, the approaches performed similarly, but showed some differences in how they functioned. An investigation into the attributes of the selected images lead to the observation that the objectness based approaches perform an elaborate proxy-task to estimate the number of objects per image. A main insight is the clear correlation between number of objects in the selected images and performance of the models trained on them.

It remains questionable if the uncertainty based approaches evaluated in this work justify the added complexity in the implementation and computational costs, compared to random sampling. Therefore, active learning approaches must further strive to be applicable to complex, real world datasets and difficult learning tasks such as object detection. Although, we discovered a promising direction of utilizing more primitive and efficient proxy-tasks, e.g. estimating the number of object per image, to base the active learning approaches on.

The assumption that, for example, night-time images are more difficult and should thus be selected by active learning approaches could not be confirmed. Which either means that the assumption is not true, which could be further verified by looking at the error scores of individual images with the respective attributes, or that the approaches simply do not select samples according to the assumption.

The experiments can be extended to include more datasets and a wider range of active learning approaches, since in the time past since the conduction of the experiments more utility functions and active learning strategies were proposed. Likewise, the application domain as well as the object detection task further warrant the additional use of other sensor modalities, e.g. Lidar or Radar. We also did not consider temporal information, e.g. video data, for stream based active learning.

7 Code Availability

The code base of this work is available to reproduce, verify, or extend the experiments conducted for this work under `https://git.ies.uni-kassel.de/public_code/a_practical_evaluation_of_active_learning_approaches_for_object_detection`.

8 Acknowledgment

This work results from the project KI Data Tooling (19A20001O) funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK).

References

1. F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, “BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning,” *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2633–2642, 2020.
2. S. Ren, K. He, R. B. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1137–1149, 2015.
3. Y. Gal, R. Islam, and Z. Ghahramani, “Deep Bayesian active learning with image data,” in *34th International Conference on Machine Learning, ICML 2017*, vol. 3, 2017, pp. 1923–1932.
4. Y. Gal and Z. Ghahramani, “Dropout as a Bayesian Approximation: Appendix,” in *33rd International Conference on Machine Learning, ICML 2016*, vol. 3, 2016, pp. 1661–1680.
5. D. D. Lewis and W. A. Gale, “A sequential algorithm for training text classifiers,” in *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1994*, 1994, pp. 3–12.
6. C. E. Shannon, “A Mathematical Theory of Communication,” *Bell System Technical Journal*, vol. 27, no. 6, 10, pp. 379–423, 623–656, 1948.
7. N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel, “Bayesian Active Learning for Classification and Preference Learning,” *ArXiv*, vol. abs/1112.5, 2011.
8. A. Kirsch, J. van Amersfoort, and Y. Gal, “BatchBALD: Efficient and diverse batch acquisition for deep Bayesian active learning,” in *Advances in Neural Information Processing Systems*, vol. 32, 2019.
9. A. Siddhant and Z. C. Lipton, “Deep Bayesian Active Learning for Natural Language Processing: Results of a Large-Scale Empirical Study,” *ArXiv*, vol. abs/1808.0, 2018.
10. D. Wu, C. T. Lin, and J. Huang, “Active learning for regression using greedy sampling,” *Information Sciences*, vol. 474, pp. 90–105, 2019.
11. D. Wu, “Pool-Based Sequential Active Learning for Regression,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, pp. 1348–1359, 2019.
12. C. Käding, E. Rodner, A. Freytag, O. Mothes, B. Barz, and J. Denzler, “Active learning for regression tasks with expected model output changes,” in *British Machine Vision Conference 2018, BMVC 2018*, 2019.

13. J. Goetz, A. Tewari, and P. Zimmerman, “Active Learning for Non-Parametric Regression Using Purely Random Trees,” in *NeurIPS*, 2018.
14. M. Herde, D. Huseljic, B. Sick, and A. Calma, “A survey on cost types, interaction schemes, and annotator performance models in selection algorithms for active learning in classification,” *IEEE Access*, vol. 9, pp. 166 970–166 989, 2021.
15. C. A. Brust, C. Käding, and J. Denzler, “Active learning for deep object detection,” in *VISIGRAPP 2019 - Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, vol. 5, 2019, pp. 181–190.
16. C. C. Kao, T. Y. Lee, P. Sen, and M. Y. Liu, “Localization-Aware Active Learning for Object Detection,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11366 LNCS, 2019, pp. 506–522.
17. S. Roy, A. Unmesh, and V. P. Namboodiri, “Deep active learning for object detection,” in *British Machine Vision Conference 2018, BMVC 2018*, 2019.
18. S. Schmidt, Q. Rao, J. Tatsch, and A. Knoll, “Advanced active learning strategies for object detection,” in *Intelligent Vehicles Symposium (IV)*, Las Vegas, NV, 2020, pp. 871–876.
19. T. Yuan, F. Wan, M. Fu, J. Liu, S. Xu, X. Ji, and Q. Ye, “Multiple instance active learning for object detection,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, virtual, 2021, pp. 5330–5339.
20. E. Haussmann, M. Fenzi, K. Chitta, J. Ivanecký, H. Xu, D. Roy, A. Mittel, N. Koumchatzky, C. Farabet, and J. M. Alvarez, “Scalable Active Learning for Object Detection,” in *IEEE Intelligent Vehicles Symposium (IV)*, Las Vegas, NV, USA, 2020, pp. 1430–1435.
21. Z. Cai and N. Vasconcelos, “Cascade R-CNN: Delving Into High Quality Object Detection,” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6154–6162, 2018.
22. T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, 2017, pp. 936–944.
23. H. Habibi Aghdam, A. Gonzales-Garcia, A. M. Lopez, and J. van de Weijer, “Active learning for deep detection neural networks,” in *International Conference on Computer Vision (ICCV)*, Seoul, South Korea, 2019, pp. 3671–3679.
24. K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, 2016, pp. 770–778.
25. S. Marcel and Y. Rodriguez, “Torchvision the machine-vision package of torch,” in *Proceedings of the 18th ACM International Conference on Multimedia*, ser. MM ’10. New York, NY, USA: Association for Computing Machinery, 2010, p. 1485–1488.
26. T.-Y. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *ECCV*, 2014.
27. P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, X. Chen, and X. Wang, “A survey of deep active learning,” *ACM Computing Surveys (CSUR)*, vol. 54, pp. 1 – 40, 2022.
28. D. Huseljic, B. Sick, M. Herde, and D. Kottke, “Separation of aleatoric and epistemic uncertainty in deterministic deep neural networks,” in *25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 9172–9179.

29. A. Kendall and Y. Gal, “What uncertainties do we need in bayesian deep learning for computer vision?” in *Conference on Neural Information Processing Systems (NIPS)*, Long Beach, CA, 2017, pp. 5574–5584.
30. L. Wright, “Ranger-deep-learning-optimizer,” 2020.
31. D. Bolya, S. Foley, J. Hays, and J. Hoffman, “Tide: A general toolbox for identifying object detection errors,” *ArXiv*, vol. abs/2008.08115, 2020.