Active Learning for Deep Object Detection

Preprint · September 2018

CITATIONS
0

READS
214

3 authors, including:

Clemens-Alexander Brust
Friedrich Schiller University Jena
10 PUBLICATIONS **85** CITATIONS

SEE PROFILE

Joachim Denzler
Friedrich Schiller University Jena
403 PUBLICATIONS **3,210** CITATIONS

SEE PROFILE

Active Learning for Deep Object Detection

Clemens-Alexander Brust[1], Christoph Käding[1,2] and Joachim Denzler[1,2]
[1]*Computer Vision Group, Friedrich Schiller University Jena, Germany*
[2]*Michael Stifel Center Jena, Germany*
{*f_author, s_author*}*@uni-jena.de*

Keywords: Active Learning, Deep Learning, Object Detection, YOLO, Continuous Learning, Incremental Learning

Abstract: The great success that deep models have achieved in the past is mainly owed to large amounts of labeled training data. However, the acquisition of labeled data for new tasks aside from existing benchmarks is both challenging and costly. Active learning can make the process of labeling new data more efficient by selecting unlabeled samples which, when labeled, are expected to improve the model the most. In this paper, we combine a novel method of active learning for object detection with an incremental learning scheme (Käding et al., 2016b) to enable continuous exploration of new unlabeled datasets. We propose a set of uncertainty-based active learning metrics suitable for most object detectors. Furthermore, we present an approach to leverage class imbalances during sample selection. All methods are evaluated systematically in a continuous exploration context on the PASCAL VOC 2012 dataset (Everingham et al., 2010).

1 Introduction

Labeled training data is highly valuable and the basic requirement of supervised learning. Active learning aims to expedite the process of acquiring new labeled data, ordering unlabeled samples by the expected value from annotating them. In this paper, we propose novel active learning methods for object detection. Our main contributions are *(i)* an incremental learning scheme for deep object detectors without catastrophic forgetting based on (Käding et al., 2016b), *(ii)* active learning metrics for detection derived from uncertainty estimates and *(iii)* an approach to leverage selection imbalances for active learning.

While active learning is widely studied in classification tasks (Kovashka et al., 2016; Settles, 2009), it has received much less attention in the domain of deep object detection. In this work, we propose methods that can be used with any object detector that predicts a class probability distribution per object proposal. Scores from individual detections are aggregated into a score for the whole image (see Fig. 1). All methods rely on the intuition that model uncertainty and valuable samples are likely to co-occur (Settles, 2009). Furthermore, we show how the balanced selection of new samples can improve the resulting performance of an incrementally learned system.

In continuous exploration application scenarios, *e.g.,* in camera streams, new data becomes available over time or the distribution underlying the problem changes itself. We simulate such an environment using splits of the PASCAL VOC 2012 (Everingham et al., 2010) dataset. With our proposed framework, a deep object detection system can be trained in an incremental manner while the proposed aggregation schemes enable selection of valuable data for annotation. In consequence, a deep object detector can explore unknown data and adapt itself involving minimal human supervision. This combination results in a complete system enabling continuously changing scenarios.

1.1 Related Work

Object Detection using CNNs An important contribution to object detection based on deep learning is R-CNN (Girshick et al., 2014). It delivers a considerable improvement over previously published sliding window-based approaches. R-CNN employs selective search (Uijlings et al., 2013), an unsupervised method to generate region proposals. A pre-trained CNN performs feature extraction. Linear SVMs (one per class) are used to score the extracted features and a threshold is applied to filter the large number of proposed regions. Fast R-CNN (Girshick, 2015) and Faster R-CNN (Ren et al., 2015) offer further improvements in speed and accuracy. Later on, R-CNN is combined with feature pyramids to enable efficient

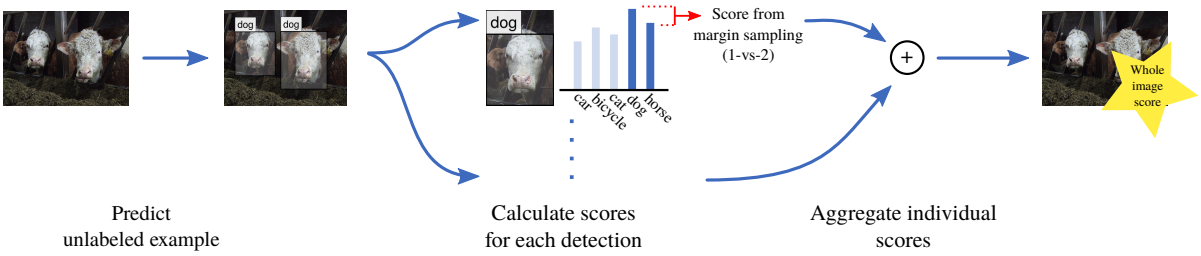

Figure 1: Our proposed system for continuous exploration scenarios. Unlabeled images are evaluated by an deep object detection method. The margins of predictions (*i.e.,* absolute difference of highest and second-highest class score) are aggregated to identify valuable instances by combining scores of individual detections.

multi-scale detections (Lin et al., 2017). YOLO (Redmon et al., 2016) is a more recent deep learning-based object detector. Instead of using a CNN as a black box feature extractor, it is trained in an end-to-end fashion. All detections are inferred in a single pass (hence the name "You Only Look Once") while detection and classification are capable of independent operation. YOLOv2 (Redmon and Farhadi, 2017) and YOLOv3 (Redmon and Farhadi, 2018) improve upon the original YOLO in several aspects. These include among others different network architectures, different priors for bounding boxes and considering multiple scales during training and detection. SSD (Liu et al., 2016) is a single-pass approach comparable to YOLO introducing improvements like assumptions about the aspect ratio distribution of bounding boxes as well as predictions on different scales. As a result of a series of improvements, it is both faster and more accurate than the original YOLO. DSSD (Fu et al., 2017) further improves upon SSD in focusing more on context with the help of deconvolutional layers.

Active Learning for Object Detection The authors of (Abramson and Freund, 2006) propose an active learning system for pedestrian detection in videos taken by a camera mounted on the front of a moving car. Their detection method is based on AdaBoost while sampling of unlabeled instances is realized by hand-tuned thresholding of detections. Object detection using generalized Hough transform in combination with randomized decision trees, called Hough forests, is presented in (Yao et al., 2012). Here, costs are estimated for annotations, and instances with highest costs are selected for labeling. This follows the intuition that those examples are most likely to be difficult and therefore considered most valuable. Another active learning approach for satellite images using sliding windows in combination with an SVM classifier and margin sampling is proposed in (Bietti, 2012). The combination of active learning for object detection with crowd sourcing is presented in (Vijayanarasimhan and Grauman, 2014). A part-based detector for SVM classifiers in combination with hashing is proposed for use in large-scale settings. Active learning is realized by selecting the most uncertain instances for labeling. In (Roy et al., 2016), object detection is interpreted as a structured prediction problem using a version space approach in the so called "difference of features" space. The authors propose different margin sampling approaches estimating the future margin of an SVM classifier.

Like our proposed approach, most related methods presented above rely on uncertainty indicators like least confidence or 1-vs-2. However, they are designed for a specific type of object detection and therefore can not be applied to deep object detection methods in general whereas our method can. Additionally, our method does not propose single objects to the human annotator. It presents whole images at once and requests labels for every object.

Active Learning for Deep Architectures In (Wang and Shang, 2014) and (Wang et al., 2016), uncertainty-based active learning criteria for deep models are proposed. The authors offer several metrics to estimate model uncertainty, including least confidence, margin or entropy sampling. Wang *et al.* additionally describe a self-taught learning scheme, where the model's prediction is used as a label for further training if uncertainty is below a threshold. Another type of margin sampling is presented in (Stark et al., 2015). The authors propose querying samples according to the quotient of the highest and second-highest class probability. The visual detection of defects using a ResNet is presented in (Feng et al., 2017). The authors propose two methods: uncertainty sampling (*i.e.,* defect probability of 0.5) and positive sampling (*i.e.,* selecting every positive sample since they are very rare) for querying unlabeled instances as model update after labeling. Another work which presents uncertainty sampling is (Liu et al., 2017). In addition, a query by committee strategy as well as ac-

tive learning involving weighted incremental dictionary learning for active learning are proposed. In the work of (Gal et al., 2017), several uncertainty-related measures for active learning are presented. Since they use Bayesian CNNs, they can make use of the probabilistic output and employ methods like variance sampling, entropy sampling or maximizing mutual information. Finally, the authors of (Beluch et al., 2018) show that ensamble-based uncertainty measures are able to perform best in their evaluation. All of the works introduced above are tailored to active learning in classification scenarios. Most of them rely on model uncertainty, similar to our applied selection criteria.

Besides estimating the uncertainty of the model, further retraining-based approaches are maximizing the expected model change (Huang et al., 2016) or the expected model output change (Käding et al., 2016a) that unlabeled samples would cause after labeling. Since each bounding box inside an image has to be evaluated according its active learning score, both measures would be impractical in terms of runtime without further modifications. A more complete overview of general active learning strategies can be found in (Kovashka et al., 2016; Settles, 2009).

2 Prerequisite: Active Learning

In active learning, a value or metric $v(x)$ is assigned to any unlabeled example x to determine its possible contribution to model improvement. The current model's output can be used to estimate a value, as can statistical properties of the example itself. A high $v(x)$ means that the example should be preferred during selection because of its estimated value for the current model.

In the following section, we propose a method to adapt an active learning metric for classification to object detection using an aggregation process. This method is applicable to any object detector whose output contains class scores for each detected object.

Classification For classification, the model output for a given example x is an estimated distribution of class scores $\hat{p}(c|x)$ over classes K. This distribution can be analyzed to determine whether the model made an uncertain prediction, a good indicator of a valuable example. Different measures of uncertainty are a common choice for selection, *e.g.,* (Ertekin et al., 2007; Fu and Yang, 2015; Hoi et al., 2006; Jain and Kapoor, 2009; Kapoor et al., 2010; Käding et al., 2016c; Tong and Koller, 2001; Beluch et al., 2018).

For example, if the difference between the two highest class scores is very low, the example may be located close to a decision boundary. In this case, it can be used to refine the decision boundary and is therefore valuable. The metric is defined using the highest scoring classes c_1 and c_2:

$$v_{1vs2}(x) \;=\; \Big(1 - \big(\max_{c_1 \in K} \hat{p}(c_1|x) - \max_{c_2 \in K \setminus c_1} \hat{p}(c_2|x)\big)\Big)^2 \;. \tag{1}$$

This procedure is known as *1-vs-2* or *margin sampling* (Settles, 2009). We use 1-vs-2 as part of our methods since its operation is intuitive and it can produce better estimates than *e.g.,* least confidence approaches (Käding et al., 2016a). However, our proposed aggregation method could be applied to any other active learning measure.

3 Active Learning for Deep Object Detection

Using a classification metric on a single detection is straightforward, if class scores are available. Though, aggregating metrics of individual detections for a complete image can be done in many different ways. In the section below, we propose simple and efficient aggregation strategies. Afterwards, we discuss the problem of class imbalance in datasets.

3.1 Aggregation of Detection Metrics

Possible aggregations include calculating the sum, the average or the maximum over all detections. However, for some aggregations, it is not clear how an image without any detections should be handled.

Sum A straightforward method of aggregation is the sum. Intuitively, this method prefers images with lots of uncertain detections in them. When aggregating detections using a sum, empty examples should be valued zero. It is the neutral element of addition, making it a reasonable value for an empty sum. A low valuation effectively delays the selection of empty examples until there are either no better examples left or the model has improved enough to actually produce detections on them. The value of a single example x can be calculated from the detections D in the following way:

$$v_{Sum}(x) \;=\; \sum_{i \in D} v_{1vs2}(x_i) \;. \tag{2}$$

Average Another possibility is averaging each detection's scores. The average is not sensitive to the

number of detections, which may make scores more comparable between images. If a sample does not contain any detections, it will be assigned a zero score. This is an arbitrary rule because there is no true neutral element *w.r.t.* averages. However, we choose zero to keep the behavior in line with the other metrics:

$$v_{Avg}(x) = \frac{1}{|D|} \sum_{i \in D} v_{1vs2}(x_i) \ . \quad (3)$$

Maximum Finally, individual detection scores can be aggregated by calculating the maximum. This can result in a substantial information loss. However, it may also prove beneficial because of increased robustness to noise from many detections. For the maximum aggregation, a zero score for empty examples is valid. The maximum is not affected by zero valued detections, because no single detection's score can be lower than zero:

$$v_{Max}(x) = \max_{i \in D} v_{1vs2}(x_i) \ . \quad (4)$$

3.2 Handling Selection Imbalances

Class imbalances can lead to worse results for classes underrepresented in the training set. In a continuous learning scenario, this imbalance can be countered during selection by preferring instances where the predicted class is underrepresented in the training set. An instance is weighted by the following rule:

$$w_c = \frac{\#\text{instances} + \#\text{classes}}{\#\text{instances}_c + 1}, \quad (5)$$

where c denotes the predicted class. We assume a symmetric Dirichlet prior with $\alpha = 1$, meaning that we have no prior knowledge of the class distribution, and estimate the posterior after observing the total number of training instances as well as the number of instances of class c in the training set. The weight w_c is then defined as the inverse of the posterior to prefer underrepresented classes. It is multiplied with $v_{1vs2}(x)$ before aggregation to obtain a final score.

4 Experiments

In the following, we present our evaluation. First we show how the proposed aggregation metrics are able to enhance recognition performance while selecting new data for annotation. After this, we will analyze the gained improvements when our proposed weighting scheme is applied. This paper describes work in progress. Code will be made available after conference publication.

Data We use the PASCAL VOC 2012 dataset (Everingham et al., 2010) to assess the effects of our methods on learning. To specifically measure incremental and active learning performance, both training and validation set are split into parts A and B in two different random ways to obtain more general experimental results. Part B is considered "new" and is comprised of images with the object classes `bird`, `cow` and `sheep` (first way) or `tvmonitor`, `cat` and `boat` (second way). Part A contains all other 17 classes and is used for initial training. The training set for part B contains 605 and 638 images for the first and second way, respectively. The decision towards VOC in favor of recently published datasets was motivated by the conditions of the dataset itself. Since it mainly contains images showing fewer objects, it is possible to split the data into a known and unknown part without having images containing classes from both parts of the splits.

Active Exploration Protocol Before an experimental run, the part B datasets are divided randomly into unlabeled batches of ten samples each. This fixed assignment decreases the probability of very similar images being selected for the same batch compared to always selecting the highest valued samples, which would lead to less diverse batches. This is valuable while dealing with data streams, *e.g.,* from camera traps, or data with low intra-class variance. The construction of diverse unlabeled data batches is a well known topic in batch-mode active learning (Settles, 2009). However, the construction of diverse batches could lead to unintended side-effects and an evaluation of those is beyond the scope of the current study. The unlabeled batch size is a trade-off between a tight feedback loop (smaller batches) and computational efficiency (larger batches). As side-effect of the fixed batch assignment, there are some samples left over during selection (*i.e.,* five for first way and eight for second way of splitting).

The unlabeled batches are assigned a value using the sum of the active learning metric over all images in the corresponding batch as a meta-aggregation. Other functions such as average or maximum could be considered too, but are also beyond the scope of this paper.

The highest valued batch is selected for an incremental training step (Käding et al., 2016b). The network is updated using the annotations from the dataset in lieu of a human annotator. Please note, annotations are not needed for update batch selection but for the update itself. This process is repeated from the point of batch valuation until there are no unlabeled batches left. The assignment of samples to unlabeled batches

is not changed during an experimental run.

Evaluation We report mean average precision (mAP) as described in (Everingham et al., 2010) and validate each five new batches (*i.e.,* 50 new samples). The result is averaged over five runs for each active learning metric and way of splitting for a total of ten runs. As a baseline for comparison, we evaluate the performance of random selection, since there is no other work suitable for direct comparison without any adjustments as of yet.

Setup – Object Detector We use YOLO as deep object detection framework (Redmon et al., 2016). More precisely, we use the YOLO-Small architecture as an alternative to larger object detection networks, because it allows for much faster training. Our initial model is obtained by fine-tuning the *Extraction* model[1] on part A of the VOC dataset for 24,000 iterations using the Adam optimizer (Kingma and Ba, 2014), for each way of splitting the dataset into parts A and B, resulting in two initial models. The first half of initial training is completed with a learning rate of 0.0001. The second half and all incremental experiments use a lower learning rate of 0.00001 to prevent divergence. Other hyperparameters match (Redmon et al., 2016), including the augmentation of training data using random crops, exposure or saturation adjustments.

Setup – Incremental Learning Extending an existing CNN without sacrificing performance on known data is not a trivial task. Fine-tuning exclusively on new data leads to a severe degradation of performance on previously learned examples (Kirkpatrick et al., 2016; Shmelkov et al., 2017). We use a straightforward, but effective fine-tuning method (Käding et al., 2016b) to implement incremental learning. With each gradient step, the mini-batch is constructed by randomly selecting from old and new examples with a certain probability of λ or $1-\lambda$, respectively. After completing the learning step, the new data is simply considered old data for the next step. This method can balance known and unknown data performance successfully. We use a value of 0.5 for λ to make as few assumptions as possible and perform 100 iterations per update. Algorithm 1 describes the protocol in more detail. The method can be applied to YOLO object detection with some adjustments. Mainly, the architecture needs to be changed when new classes are added. Because of the design of YOLO's output layer, we rearrange the weights to fit new classes, adding 49 weights per class.

[1] `http://pjreddie.com/media/files/extraction.weights`

4.1 Results

We focus our analysis on the new, unknown data. However, not losing performance on known data is also important. We analyze the performance on the known part of the data (*i.e.,* part A of the VOC dataset) to validate our method. In worst case, the mAP decreases from 36.7% initially to 32.1% averaged across all experimental runs and methods although three new classes were introduced. We can see that the incremental learning method from (Käding et al., 2016b) causes only minimal losses on known data. These losses in performance are also referred to as "catastrophic forgetting" in literature (Kirkpatrick et al., 2016). The method from (Käding et al., 2016b) does not require additional model parameters or adjusted loss terms for added samples like comparable approaches such as (Shmelkov et al., 2017) do, which is important for learning indefinitely.

Performance of active learning methods is usually evaluated by observing points on a learning curve (*i.e.,* performance over number of added samples). In Table 1, we show the mAP for the new classes from part B of VOC at several intermediate learning steps as well as exhausting the unlabeled pool. In addition we show the area under learning curve (AULC) to further improve comparability among the methods. In our experiments, the number of samples added equals the number of images.

Quantitative Results – Fast Exploration Gaining accuracy as fast as possible while minimizing the human supervision is one of the main goals of active learning. Moreover, in continuous exploration scenarios, like faced in camera feeds or other continuous automatic measurements, it is assumed that new data is always available. Hence, the pool of valuable examples will rarely be exhausted. To assess the performance of our methods in this fast exploration context, we evaluate the models after learning learning small amounts of samples. At this point there is still a large number of diverse samples for the methods to choose from, which makes the following results much more relevant for practical applications than results on the full dataset.

In general, we can see that incremental learning works in the context of the new classes in part B of the data, meaning that we observe an improving performance for all methods. After adding only 50 samples, *Max* and *Avg* are performing better than passive selec-

Algorithm 1: Detailed description of the experimental protocol. Please note that in an actual continuous learning scenario, new examples are always added to $\mathfrak{U}$. The loop is never left because $\mathfrak{U}$ is never exhausted. The described splitting process would have to be applied regularly.

Require: Known labeled samples $\mathfrak{L}$, unknown samples $\mathfrak{U}$, initial model f_0, active learning metric v

$\mathfrak{U} = \mathfrak{U}_1, \mathfrak{U}_2, \ldots \leftarrow$ split of $\mathfrak{U}$ into random batches
$f \leftarrow f_0$

while $\mathfrak{U}$ is not empty **do**
  calculate scores for all batches in $\mathfrak{U}$ using f
  $\mathfrak{U}_{best} \leftarrow$ highest scoring batch in $\mathfrak{U}$ according to v

  $\mathcal{Y}_{best} \leftarrow$ annotations for $\mathfrak{U}_{best}$ *human-machine interaction*
  $f \leftarrow$ incrementally train f using $\mathfrak{L}$ and $(\mathfrak{U}_{best}, \mathcal{Y}_{best})$

  $\mathfrak{U} \leftarrow \mathfrak{U} \backslash \mathfrak{U}_{best}$
  $\mathfrak{L} \leftarrow \mathfrak{L} \cup (\mathfrak{U}_{best}, \mathcal{Y}_{best})$
end while

Table 1: Validation results on part B of the VOC data (*i.e., new classes only*). **Bold** face indicates block-wise best results, *i.e.,* best results with and without additional weighting ($\cdot + w$). Underlined face highlights overall best results.

	50 samples mAP/AULC	100 samples mAP/AULC	150 samples mAP/AULC	200 samples mAP/AULC	250 samples mAP/AULC	All samples mAP/AULC
Baseline						
Random	8.7 / 4.3	12.4 / 14.9	15.5 / 28.8	18.7 / 45.9	21.9 / 66.2	32.4 / 264.0
Our Methods						
Max	**9.2** / **4.6**	12.9 / **15.7**	15.7 / 30.0	19.8 / 47.8	22.6 / 69.0	32.0 / **269.3**
Avg	9.0 / 4.5	12.4 / 15.2	15.8 / 29.2	19.3 / 46.8	**22.7** / 67.8	**33.3** / 266.4
Sum	8.5 / 4.2	**14.3** / 15.6	**17.3** / **31.4**	**19.8** / **49.9**	22.7 / **71.2**	32.4 / 268.2
Max + *w*	**9.2** / **4.6**	13.0 / **15.7**	17.0 / 30.7	20.6 / 49.5	23.2 / 71.4	**33.0** / 271.0
Avg + *w*	8.7 / 4.3	12.5 / 14.9	16.6 / 29.4	19.9 / 47.7	22.4 / 68.8	32.7 / 267.1
Sum + *w*	8.7 / 4.4	**13.7** / 15.6	**17.5** / **31.2**	**20.9** / **50.4**	**24.3** / **72.9**	32.7 / **273.6**

tion while the *Sum* metric is outperformed marginally. When more and more samples are added (*i.e.,* 100 to 250 samples), we observe a superior performance of the *Sum* aggregation. But also the two other aggregation techniques are able to reach better rates than mere random selection. We attribute the fast increase of performance for the *Sum* metric to its tendency to select samples with many object inside which leads to more annotated bounding boxes. However, the target application is a scenario where the amount of unlabeled data is huge or new data is approaching continuously and hence a complete evaluation by humans is infeasible. Here, we consider the amount of images to be evaluated more critical as the time needed to draw single bounding boxes. Another interesting fact is the almost equal performance of *Max* and *Avg* which can be explained as follows: the VOC dataset consists mostly of images with only one object in them. Therefore, both metrics lead to a similar score if objects are identified correctly.

We can also see that the proposed balance handling (*i.e.,* $\cdot + w$) causes slight losses in performance at very early stages up to 100 samples. At subsequent stages, it helps to gain noticeable improvements. Especially for the *Sum* method benefits from the sample weighting scheme. A possible explanation for this behavior would be the following: At early stages, the classifier has not seen many samples of each class and therefore suffers more from miss-classification errors. Hence, the weighting scheme is not able to encourage the selection of rare class samples since the classifier decisions are still too unstable. At later stages, this problem becomes less severe and the weighting scheme is much more helpful than in the beginning. This could also explain the performance of *Sum* in general. Further details on learning pace are given later in a qualitative study on model evolution. Additionally, the *Sum* aggregation tends to select batches with many detections in it. Hence, it is natural that the improvement is noticeable the most with this aggregation technique since it helps to find batches with many rare objects in it.

Quantitative Results – All Available Samples In our case, active learning only affects the sequence of

unlabeled batches if we train until there is no new data available. Therefore, there are only very small differences between each method's results for mAP after training has completed. The small differences indicate that the chosen incremental learning technique (Käding et al., 2016b) is suitable for the faced scenario. In continuous exploration, it is usually assumed that there will be more new unlabeled data available than can be processed. Nevertheless, evaluating the long term performance of our metrics is important to detect possible deterioration over time compared to random selection. In contrast to this, the differences in AULC arise from the improvements of the different methods during the experimental run and therefore should be considered as distinctive feature implying the performance over the whole experiment. Having this in mind, we can still see that *Sum* performs best while the weighting generally leads to improvements.

Quantitative Results — Class-wise Analysis To validate the efficacy of our sample weighting strategy as discussed in Section 3.2, it is important to measure not only overall performance, but to look at metrics for individual classes. Fig. 2 shows the performance over time on the validation set for each individual class. For reference, we also provide the class distribution over the relevant part of the VOC dataset, indicated by number of object instances in total as well as number of images with at least one instance in it.

In the first row, we observe an advantage for the weighted method when looking at the performance of `cow`. Out of the three classes in this way of splitting `cow` has the fewest instances in the dataset. The performance of `tvmonitor` in the second row shows a similar pattern, where it is also the class with the lowest number of object instances in the dataset. Analyzing `bird` and `cat`, the top classes by number of instances in each way of splitting, we observe only small differences in performance. Thus, we can show evidence that our balancing scheme is able to improve performance on rare classes while it does not effect performance on frequent classes.

Intuitively, these observations are in line with our expectations regarding our handling of class imbalances, where examples of rare classes should be preferred during selection. We start to observe the advantages after around 100 training examples, because, for the selection to happen correctly, the prediction of the rare class needs to be correct in the first place.

Qualitative Results – Sample Valuation We calculate whole image scores over `bird`, `cow` and `sheep` samples using our corresponding *initial* model trained on the remaining classes for the first way of splitting.

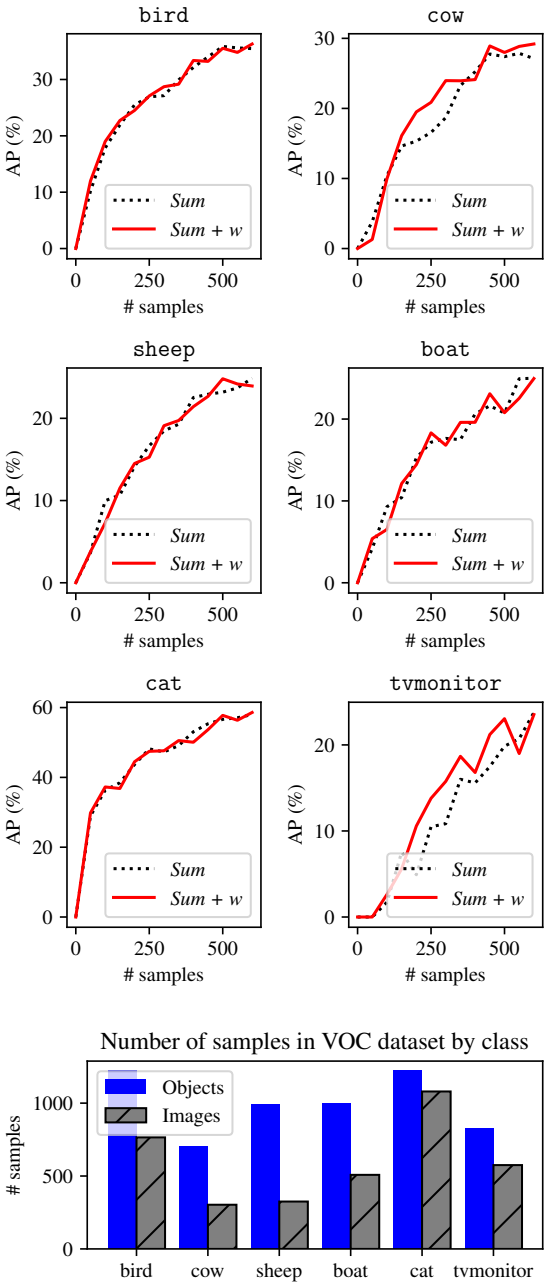

Figure 2: Class-wise validation results on part B of the VOC dataset (*i.e.*,, unknown classes). The first row details the first way of splitting (`bird`,`cow`,`sheep`) while the second row shows the second way (`boat`,`cat`,`tvmonitor`). For reference, the distribution of samples (object instances as well as images with at least one instance) over the VOC dataset is provided in the third row.

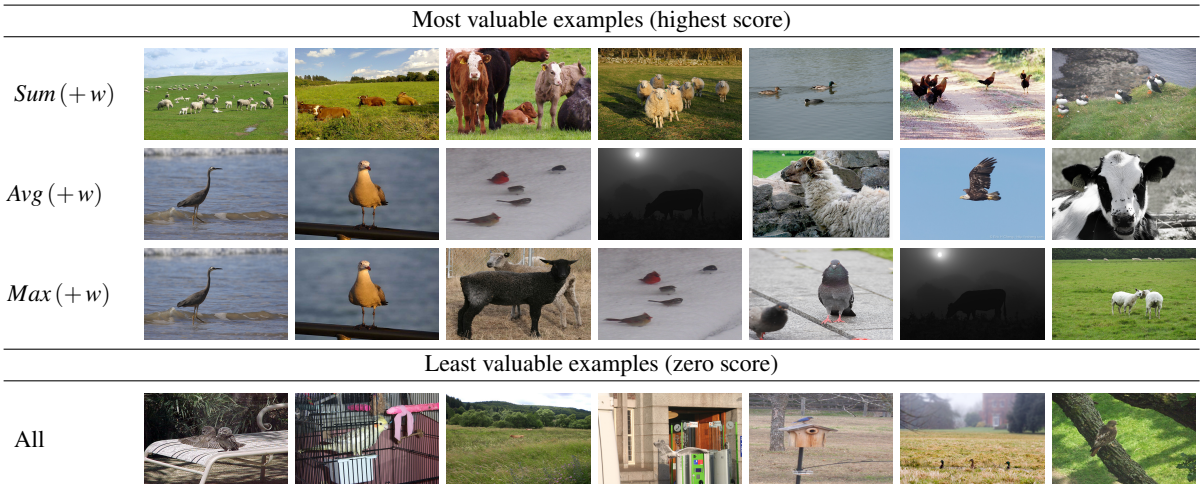

Figure 3: Value of examples of `cow`, `sheep` and `bird` as determined by the *Sum*, *Avg* and *Max* metrics using the *initial* model. The top seven selection is not affected by using our weighting method to counter training set class imbalaces.

Figure 3 shows those images that the three aggregation metrics consider the most valuable. Additionally, common zero scoring images are shown. The least valuable images shown here are representative of all proposed metrics because they do not lead to any detections using the *initial* model. Note that there are more than seven images with zero score in the training dataset. The images shown in the figure have been selected randomly.

Intuitively, the *Sum* metric should prefer images with many objects in them over single objects, even if individual detection values are low. Although VOC contains mostly of images with a single object, all seven of the highest scoring images contain at least three objects. The *Average* and *Maximum* metric prefer almost identical images since the average and maximum are used to be nearly equal for few detections. With few exceptions, the most valuable images contain pristine examples of each object. They are well lit and isolated. The objects in the zero scoring images are more noisy and hard to identify even for the human viewer, resulting in fewer reliable detections.

Qualitative Results – Model Evolution Observing the change in model output as new data is learned can help estimate the number of samples needed to learn new classes and identify possible confusions. Fig. 4 shows the evolution from initial guesses to correct detections after learning 150 samples, corresponding to an fast exploration scenario. For selection, the *Sum* metric is used.

The class confusions shown in the figure are typical for this scenario. `cow` and `sheep` are recognized as visually similar `dog`, `horse` and `cat`. `bird` is often classified as `aeroplane`. After selecting and learning 150 samples, the objects are detected and classified correctly and reliably.

During the learning process, there are also *unknown* objects. Please note, being able to mark objects as *unknown* is a direct consequence of using YOLO. Those objects have a detection confidence above the required threshold, but no classification is certain enough. This property of YOLO is important for the discovery of objects of new classes. Nevertheless, if similar information is available from other detection methods, our techniques could easily be applied.

5 Conclusions

In this paper, we propose several uncertainty-based active learning metrics for object detection. They only require a distribution of classification scores per detection. Depending on the specific task, an object detector that will report objects of unknown classes is also important. Additionally, we propose a sample weighting scheme to balance selections among classes.

We evaluate the proposed metrics on the PASCAL VOC 2012 dataset (Everingham et al., 2010) and offer quantitative and qualitative results and analysis. We show that the proposed metrics are able to guide the annotation process efficiently which leads to superior performance in comparison to a random selection baseline. In our experimental evaluation, the *Sum* metric is able to achieve best results overall which can

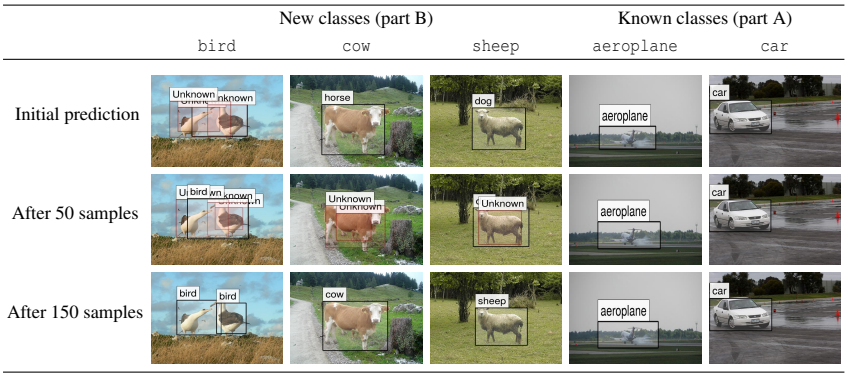

Figure 4: Evolution of detections on examples from validation set.

be attributed to the fact that it tends to select batches with many single objects in it. However, the targeted scenario is an application with huge amounts of unlabeled data where we consider the amount of images to be evaluated as more critical than the time needed to draw single bounding boxes. Examples would be camera streams or camera trap data. To expedite annotation, our approach could be combined with a weakly supervised learning approach as presented in (Papadopoulos et al., 2016). We also showed that our weighting scheme leads to even increased accuracies.

All presented metrics could be applied to other deep object detectors, such as the variants of SSD (Liu et al., 2016), the improved R-CNNs *e.g.,* (Ren et al., 2015) or the newer versions of YOLO (Redmon and Farhadi, 2017). Moreover, our proposed metrics are not restricted to deep object detection and could be applied to arbitrary object detection methods if they fulfill the requirements. It only requires a complete distribution of classifications scores per detection. Also the underlying uncertainty measure could be replaced with arbitrary active learning metrics to be aggregated afterwards. Depending on the specific task, an object detector that will report objects of unknown classes is also important.

The proposed aggregation strategies also generalize to selection of images based on segmentation results or any other type of image partition. The resulting scores could also be applied in a novelty detection scenario.

REFERENCES

Abramson, Y. and Freund, Y. (2006). Active learning for visual object detection. Technical report, University of California, San Diego.

Beluch, W. H., Genewein, T., Nürnberger, A., and Köhler, J. M. (2018). The power of ensembles for active learning in image classification. In *Computer Vision and Pattern Recognition (CVPR)*.

Bietti, A. (2012). Active learning for object detection on satellite images. Technical report, California Institute of Technology, Pasadena.

Ertekin, S., Huang, J., Bottou, L., and Giles, L. (2007). Learning on the border: active learning in imbalanced data classification. In *Conference on Information and Knowledge Management*.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision (IJCV)*.

Feng, C., Liu, M.-Y., Kao, C.-C., and Lee, T.-Y. (2017). Deep active learning for civil infrastructure defect detection and classification. In *International Workshop on Computing in Civil Engineering (IWCCE)*.

Fu, C.-J. and Yang, Y.-P. (2015). A batch-mode active learning svm method based on semi-supervised clustering. *Intelligent Data Analysis*.

Fu, C.-Y., Liu, W., Ranga, A., Tyagi, A., and Berg, A. C. (2017). Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*.

Gal, Y., Islam, R., and Ghahramani, Z. (2017). Deep bayesian active learning with image data. *arXiv preprint arXiv:1703.02910*.

Girshick, R. (2015). Fast R-CNN. In *International Conference on Computer Vision (ICCV)*.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR)*.

Hoi, S. C., Jin, R., and Lyu, M. R. (2006). Large-scale text categorization by batch mode active learn-

ing. In *International Conference on World Wide Web (WWW)*.

Huang, J., Child, R., Rao, V., Liu, H., Satheesh, S., and Coates, A. (2016). Active learning for speech recognition: the power of gradients. *arXiv preprint arXiv:1612.03226*.

Jain, P. and Kapoor, A. (2009). Active learning for large multi-class problems. In *Computer Vision and Pattern Recognition (CVPR)*.

Käding, C., Freytag, A., Rodner, E., Perino, A., and Denzler, J. (2016a). Large-scale active learning with approximated expected model output changes. In *German Conference on Pattern Recognition (GCPR)*.

Käding, C., Rodner, E., Freytag, A., and Denzler, J. (2016b). Fine-tuning deep neural networks in continuous learning scenarios. In *ACCV Workshop on Interpretation and Visualization of Deep Neural Nets (ACCV-WS)*.

Käding, C., Rodner, E., Freytag, A., and Denzler, J. (2016c). Watch, ask, learn, and improve: A lifelong learning cycle for visual recognition. In *European Symposium on Artificial Neural Networks (ESANN)*.

Kapoor, A., Grauman, K., Urtasun, R., and Darrell, T. (2010). Gaussian processes for object categorization. *International Journal of Computer Vision (IJCV)*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv: 1412.6980*.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N. C., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. (2016). Overcoming catastrophic forgetting in neural networks. *arXiv preprint arXiv:1612.00796*.

Kovashka, A., Russakovsky, O., Fei-Fei, L., and Grauman, K. (2016). Crowdsourcing in computer vision. *Foundations and Trends in Computer Graphics and Vision*.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection. In *CVPR*.

Liu, P., Zhang, H., and Eom, K. B. (2017). Active deep learning for classification of hyperspectral images. *Selected Topics in Applied Earth Observations and Remote Sensing*.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). SSD: Single shot multibox detector. In *European Conference on Computer Vision (ECCV)*.

Papadopoulos, D. P., Uijlings, J. R. R., Keller, F., and Ferrari, V. (2016). We dont need no bounding-boxes: Training object class detectors using only human verification. In *Computer Vision and Pattern Recognition (CVPR)*.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Computer Vision and Pattern Recognition (CVPR)*.

Redmon, J. and Farhadi, A. (2017). Yolo9000: Better, faster, stronger. In *Computer Vision and Pattern Recognition (CVPR)*.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Neural Information Processing Systems (NIPS)*.

Roy, S., Namboodiri, V. P., and Biswas, A. (2016). Active learning with version spaces for object detection. *arXiv preprint arXiv:1611.07285*.

Settles, B. (2009). Active learning literature survey. Technical report, University of Wisconsin–Madison.

Shmelkov, K., Schmid, C., and Alahari, K. (2017). Incremental learning of object detectors without catastrophic forgetting. In *International Conference on Computer Vision (ICCV)*.

Stark, F., Hazırbas, C., Triebel, R., and Cremers, D. (2015). Captcha recognition with active deep learning. In *Workshop New Challenges in Neural Computation*.

Tong, S. and Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research (JMLR)*.

Uijlings, J. R., Van De Sande, K. E., Gevers, T., and Smeulders, A. W. (2013). Selective search for object recognition. *International Journal of Computer Vision (IJCV)*, 104(2):154–171.

Vijayanarasimhan, S. and Grauman, K. (2014). Large-scale live active learning: Training object detectors with crawled data and crowds. *International Journal of Computer Vision (IJCV)*.

Wang, D. and Shang, Y. (2014). A new active labeling method for deep learning. In *International Joint Conference on Neural Networks (IJCNN)*.

Wang, K., Zhang, D., Li, Y., Zhang, R., and Lin, L. (2016). Cost-effective active learning for deep image classification. *Circuits and Systems for Video Technology*.

Yao, A., Gall, J., Leistner, C., and Van Gool, L. (2012). Interactive object detection. In *Computer Vision and Pattern Recognition (CVPR)*.