Rapid Annotation Tool to Train Novel Concept Detectors with Active Learning

Maaike H. T. de Boer, Henri Bouma, Maarten Kruithof and Bart Joosten

Data Science & Intelligent Imaging
TNO
The Hague, The Netherlands
e-mail: maaike.deboer@tno.nl

Abstract— **Annotating a large set of images, especially with bounding boxes, is a tedious task. In this paper, we propose an intuitive image annotation tool. This tool not only allows (non-expert) users to annotate images with novel concepts, but is also able to achieve acceptable performance with a smaller number of annotated images. The tool can also propose detections on unannotated images, to provide faster annotation and insight in the performance of the system. The tool is based on a Single Shot Multi-box Detector (SSD) neural network with active learning, based on showing the images with high-confidence detections first, to have a fast verification and re-training. An experiment on simulated data shows that this active learning method can achieve higher performance in a shorter expected annotation time with a small number of images (less than 500). A small experiment on user annotated data shows that the annotation tool allows faster annotation compared to without the annotation tool.**

Keywords-image annotation; concept localization; deep learning; active learning.

I. Introduction

Concept detection is relevant to automatically detect and localize concepts in images and facilitate user query by keyword to find relevant images. Some generic concepts are publicly available (e.g., in YOLO9000 [1] or SSD [2]), but this is not sufficient for many applications in the security domain. For example, when looking for radicalization in online videos or when looking for products on illegal market places, specific concept detectors are required. For law-enforcement agencies, it is important to adapt a concept detector for their own specific concepts. Therefore, it is important to have an annotation tool that assists users to flexibly train novel concepts with minimal annotation effort.

Our main contribution is that we demonstrate an annotation tool that can use different active-learning strategies to train novel concepts with minimal effort. High-confidence detection has the advantage that minimal adjustments are needed [3]. Uncertain detections have the advantage that they are close to the decision boundary and that only a minimal amount of detections is needed [4]. In our experiments on the Nexar Challenge dataset [42], we show that the high-confidence detections minimize the annotation time and that both approaches perform better than random selection of the data. In our experiment on traffic images, we show that working with the annotation tool and active learning is faster compared to without the annotation tool.

The outline of this paper is as follows. Section 2 gives an overview of related work, Section 3 describes the annotation tool, Section 4 describes the experiments with the different annotation techniques, Section 5 shows the results and Section 6 summarizes conclusions.

II. Related Work

In active learning, the results that are most informative for the system are displayed to a user to annotate and quickly learn a better model. We focus on active learning in which a large pool of unlabeled data is present and where the user may examine and select items from (pool-based sampling), as opposed to active learning based on streaming data (selective sampling) or synthesized data (query synthesis) [5][6]. Methods to measure informativeness include *uncertainty sampling* [7], *query-by-committee, expected gradient length, Fisher information* and *information density* [6]. Methods in uncertainty sampling include using the posterior probability or the entropy to measure the uncertainty and use the most uncertain items to learn from. Query-by-committee involves the Kullback-Leibler divergence [44] and voting of multiple classifiers to include items the classifiers disagree on. Expected gradient length uses the item that would create the largest change in the model if the label was known (largest expected gradient). Using the Fisher information [45], the item that minimizes the model variance is chosen. Information density weights the informativeness with the average similarity to all other items. While the other methods might favor outliers to select as most informative, this method does not.

In the computer-vision domain, active learning is typically used to train (or improve) concepts [8]. Active learning is distinguished from relevance feedback. In relevance feedback, the goal is to create a better model for a certain query by using positive and negative results, but not necessarily the most informative results. Typically in computer vision, the uncertainty sampling technique is used in which the items closest to the current boundary between the positive and negative items are perceived as the most uncertain items [9]-[13]. Zhao and Ding [14] use uncertainty sampling and use the top list as uncertain samples and the bottom list as fake negatives. Goh et al. [15] propose different sampling strategies for different semantic concepts based on scarcity, isolation and diversity, and Luan et al. [16] propose to start with items far from the boundary and move toward the items close to the boundary. Gavves et al. [17] propose to use zero-shot classifiers with priors to initialize and use a maximum conflict-label equality condition to select the most informative items. Holub et al. [18] and Kovashka et al. [19] use the entropy to determine the most informative items. Vondrick and Ramanan [20] use the Expected Gradient Length method. Dasgupta and Hsu [21] use hierarchical sampling and Zhu et al. [22] use a neighborhood graph on the unlabeled data.

With the current advances in Deep Learning, active learning has also been used. The activation of the softmax can be interpreted as the distance from the decision boundary [23]. Wang et al. [24] use the softmax response and pseudo-labelling of `confident' samples in active learning with neural networks, and Zhou et al. [25] use the softmax response from Restricted Boltzmann machines. Stark et al. [26] use the highest output and divide this by the second highest to obtain an uncertainty. Geifman and El-Yaniv [23] and Sener and Savarese [27] propose to use coresets of the unlabeled data based on the activations in the neural network. Gal et al. [28] compare different informativeness measures, including maximum entropy, mutual information method named BALD by Houlsby et al. [29], and variation ratios, for Bayesian Neural Networks. Ducoffe et al. [30] use the query-by-committee strategy. They use a committee of partial Convolutional Neural Networks (CNNs) and batchwise dropout. The informativeness of an item is measured by the quantity of disagreement about the prediction of the label among the partial CNNs.

In concept localization, the goal is not only to correctly detect a concept, but to also localize this concept. There are several ways to handle concept localization [3] including drawing bounding boxes, segmentation, using point-click methods, using eye-tracking, using interactive annotation, using weakly-supervised object localization techniques and using active learning. In the weakly-supervised object localization techniques, Kolesnikov and Lampert [31] propose an annotation technique to improve object localization. This technique is based on the insight that objects and distractors form different clusters in the representation of a deep neural network. Cinbis et al. [32] use multi-fold multiple instance learning for the weakly supervised object localization. Konyushkova et al. [3] compare concept localization and annotation techniques such as weak and strong detectors, the difference between Drawing and Verification of the boxes, horizontal (re-training the whole detector) and vertical re-training (using a fixed detector and re-train with only the new part). The results show that horizontal training is better than vertical re-training. They used an annotation set of almost 5,000 images. Kao et al. [33] propose different evaluation metrics for localization: localization tightness (by estimating how tight the bounding box might enclose the true bounding box) and localization stability (by adding Gaussian noise) to select the items for active learning.

III. Annotation Tool

We developed an annotation tool where the user can annotate given concepts and train a deep neural network to detect and localize these concepts in an image. The user can select the concepts using a rectangle selection tool, as depicted in Figure 1. The user can upload images with the Graphical User Interface (GUI) to annotate or detect concepts. The user can also upload reference images for each concept. This will determine the concepts the tool is able to detect.

A. *Deep Learning Network*

The network we use for detecting the concepts is the Single Shot multi-box Detector (SSD) network [2]. We use the SSD300 network, which takes an image of 300 by 300 pixels as input and outputs the locations of detected concepts with a confidence score between 0 and 1. This confidence can be used to threshold the resulting detections. The number of output concepts is set to the number of concepts defined in the GUI. The network is pretrained on PASCAL VOC [34], MS-COCO [35], and ILSVRC [36].

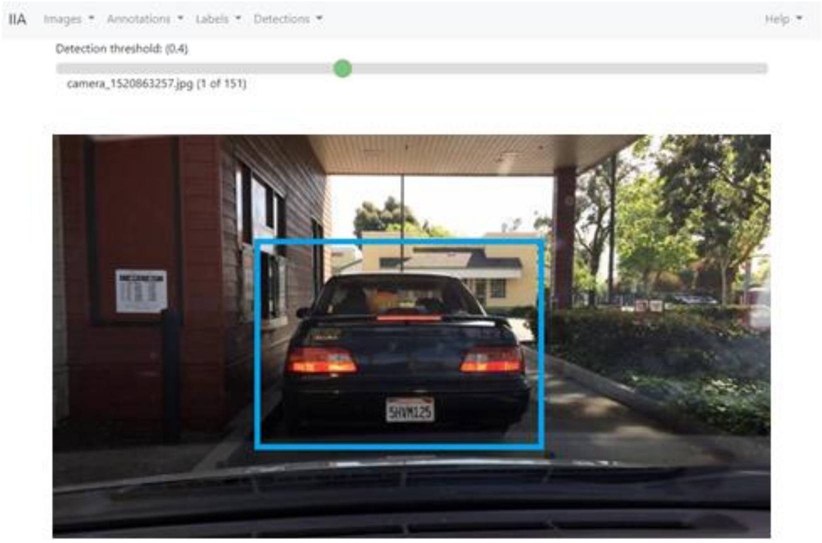
B. *(Re)Training for Concept Detection*

After annotating a number of images, a neural network can be trained to detect the annotated concepts. We take all images that have a detection as an input to the training of the network. The images are converted to images of 300 by 300 pixels and the detections are converted using detection priors for input of the network. We freeze the first 3 layers to decrease the chance of overtraining on the current dataset. We train for 20 epochs and store the weights for each epoch. We use horizontal flipping and saturation variance for image augmentation. The batch size is set to 4 images. The weight file with the smallest loss is chosen as the weights to detect the concepts. Each time the network is trained the weights are reset to the pretrained weights on PASCAL VOC, MS-COCO, and ILSVRC. After the network is trained, the tool can run the network on an image and show the concepts the network detected. The slider controls the threshold of which detections are visible in the GUI, as shown in Figure 2.

The resulting detections can now be corrected by moving or resizing them or they can be accepted as they are. This process can be repeated multiple times resulting in increasing performance of the model.

C. *Active Learning*

As active learning technique, we choose the method by Konyushkova et al. [13] (*high-confidence*). The items will be sorted from highest to lowest, so the most confident items will be shown to the user first.



Detection threshold: (0.4)

camera_1520863257.jpg (1 of 151)

Figure 1. Overview of the GUI of the annotation tool; two concepts are annotated in this image.

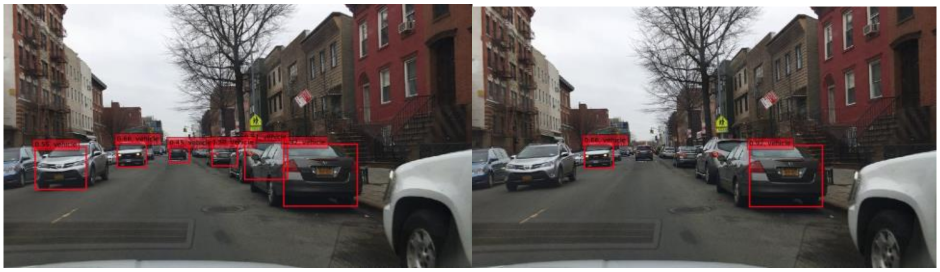


Figure 2. Detected concepts by the network with a low threshold (0.4) on the left and a high threshold (0.6) on the right.

IV. Experiments

In our experiments, we want to 1) verify our active learning choice and 2) validate that the annotation tool with active learning improves the annotation speed. In the first experiment, we use a vehicle dataset and calculate the anticipated performance and timing for each active learning method. In the second experiment, we use a street view dataset and ask four annotator to annotate the cars, bikes and persons.

A. *Simulation*

1) *Dataset*

We use the NEXET data from the Nexar Challenge 2 [42]This open dataset contains 50,000 diverse images from the rear of vehicles from different locations. The bounding box annotations are included. We use the 5,000 images taken at daylight from New York city, with approximately 16,900 detections in total. All classes (car, vehicle, truck, pickup_truck, van) are renamed to 'vehicle' to focus on just one class. We randomly select 60% as train set (10,200 detections) and 40% as a held-out test set (6,700 detections). As evaluation we use the evaluation script provided with the challenge, that calculates the mean Average Precision (mAP) with an Intersection over Union (IoU) of 0.75.

2) *Conditions*

In our experiments we compare three conditions: 1. our chosen active learning technique based on Konyushkova et al. [13] (*high-confidence*), 2. the baseline (*random* selection of images) and 3. the uncertainty sampling technique (*uncertainty*).

In the *uncertainty* condition, the items closest to the current boundary between the positive and negative items are perceived as the most uncertain items. In this experiment, we select the items around the confidence value of 0.4 as most uncertain (based on experience):

$$u_i = |\,0.4 - \; c_i|\,,$$

where c_i is the confidence of item i and u_i is the uncertainty of item i.

Based on the uncertainty items, the images are sorted in the order of lowest to highest, so the images with the most uncertain items will be shown to the user first. The images are, thus, selected based on a single uncertain detection. All detections of this image, including the possibly more certain detection, are shown to the user. In the *random* selection, random images are selected and in *high-confidence* the values of c_i will be sorted from highest to lowest, so the most confident items will be shown to the user first.

3) Active Learning runs

For each of the three conditions explained in the previous section, we start with a model that is trained on 125 randomly selected images. We then apply the trained model on the train images again to get the detections. We select 50 new images according to the condition and train a model on the 125 + 50 images. We thus train three new models, one based on each condition, with a different set of 175 images of which 50 are new. We apply this new model for this specific condition on the train images again and select 75 new images according to the condition. We train a model on the 125 + 50 + 75 images. We increase the number of images added, because a larger trainset requires a larger number of images being added to this set to make a difference. We keep on adding images with increasing step size until 2000 images.

4) Simulated Timing

In our experiments, we can automatically calculate the performance using the different active learning techniques, but we need an estimation of annotation time to simulate the timing. In the literature, different annotation times are mentioned [37]-[40], varying from 1.6 seconds to verify a bounding box to 25 second to draw a bounding box. An explanation for these differences in timing are the quality of the bounding box. Based on the results from these papers, we assume that it will take at least twice as long to draw a bounding box compared to verify a bounding box. If the bounding box is, however, not correct, our tool allows users to adjust the bounding box. In previous experiments [41], we found that adjusting a bounding box takes on average twice as long as drawing a new bounding box. We use these proportions to indicate the timing.

Besides the timing to verify and modify a bounding box, we need a definition of when a bounding box is correct. We use the IoU for this purpose. If the IoU is higher than or equal to 0.9, the bounding box is perceived correct. If the IoU is between 0.5 and 0.9, the bounding box should be modified. In the cases that the IoU is lower than 0.5, no close enough match is found and a new bounding box should be drawn.

Based on literature and our own previous experiment, we take the following annotation times (Table 1).

TABLE I. TIMING ESTIMATES

	Definition	Time (seconds)
ValidateCorrectBBox	IoU => 0.9	0.5
ModifyBBox	0.5 <= IoU < 0.9	2.0
CreateBBox	IoU < 0.5	1.0

B. User Experiment

We use the vehicle dataset from the H2020 InDeV ("In-Depth understanding of accident causation for Vulnerable road users") project [43]. The dataset consists of 269 images and in total 1424 annotated vehicle bounding boxes. Of this dataset, 2%, 10% or 50% is used for training and 66 images (25%) are used for performance estimation. Four volunteers each annotated the same 66 images from this dataset four times. The first experiment is a manual mode and the other experiments are in assisted mode. The second experiment is based on a detector that is trained on a random selection of 2% of the data. In the third experiment, the detector is trained twice. The detector is first trained on 2% of the data, then the uncertainty-based active-learning approach is used to select the next 8% of data and the detector is trained again on the total 10%. In the fourth experiment, the detector is trained three times: first on random 2%, then on 10% and 50%, to allow reordering with active learning. To compensate for a learning effect, we use a Latin square.

The dataset is fully annotated. Therefore, is was possible to prepare all the data and perform training offline. So, the users only had to annotate the 66 images during the experiment.

V. RESULTS

A. Simulation

1) mAP Performance

Figure 3 shows the mAP performance for different conditions (average over 10 runs). The plot shows that *high-confidence* stably increases with an increasing number of images. At 350 images, high-confidence reaches a mAP that is 22% higher than the mAP for random. However, this technique flattens out at the end. This is in agreement with the expectations, because high-confidence detections are becoming less and less informative. *Uncertainty* is closer to random with a smaller number of images, but improves with more images compared to *high-confidence.* At 2000 images, uncertainty reaches a mAP that is 3% higher than the mAP of high-confidence. This is also in agreement with the expectations, because initially the low-confidence detections can be confusing, but in the end the uncertain detections appear most informative.

2) Simulated Timing

Figure 4 shows the timing for the different performances (average over 10 runs). Random_baseline is without using the detections and Random is with using the detections.

Random_baseline is slower than *random* (including detections) with a smaller number of images. *High-confidence* is the techniqu that achieves a high performance in the least time. The high-confidence approach reaches an mAP of 0.2 70% faster than random. Because this technique uses the detections with the highest confidence, detections were more often validated as correct (with minimal annotation time) without necessity to modify the detections.

Uncertainty is not faster compared to random. The results are summarized in Table II. The table shows that high-confidence sampling reaches higher mAP in less time than the alternative methods.

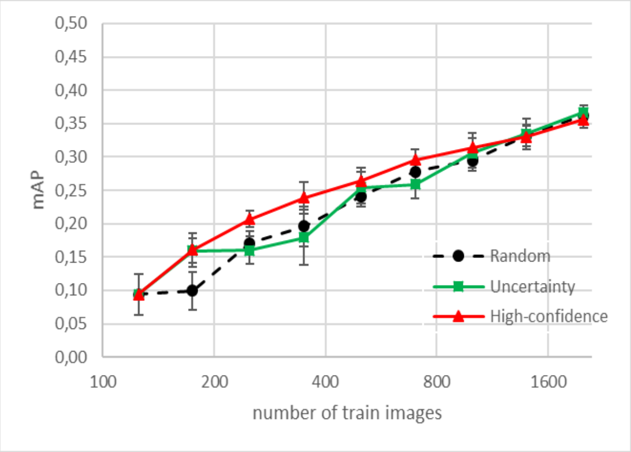


Figure 3. mAP performance for different number of train images, for all three conditions.

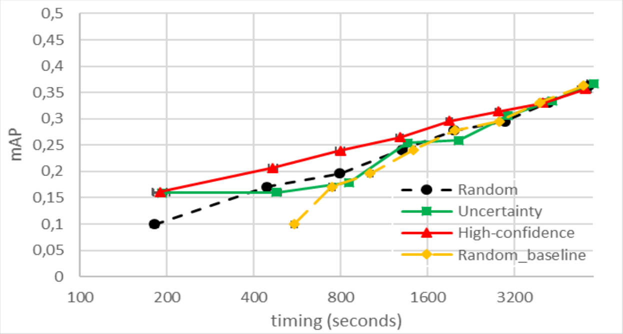


Figure 4. Estimated timing in seconds with respect to the mAP performance.

TABLE II. SUMMARY SIMULATION RESULTS

	mAP (%) at 250 im.	Time (min) at mAP=20%
Random sampling	17	13
Uncertainty sampling	16	16
High-confidence sampling	**21**	**7**

B. User Experiment

Tables III and IV show the results for the user experiment. Manual mode is significantly slower than the assisted mode, and the 50% active learning approach is significantly faster than random 2%. Table IV shows that there is a learning effect: the first experiment is 25% slower than the average annotation time. This is expected, because the same 66 images are annotated in each condition. If we compensate for the learning effect by dividing the time by the effect (i.e. for first experiment divide by 1.25), the conclusion on manual vs. assisted is strengthened and the difference between random 2% and active 50% is also strengthened.

TABLE III. SUMMARY OF USER RESULTS

	Manual	Assisted Random 2%	Assisted Active 10%	Assisted Active 50%
Average Timing (sec)	1078 ± 182	634 ± 207	562 ± 246	443 ± 87

TABLE IV. TIMING PER EXPERIMENT (ORDER)

	Exp 1	Exp 2	Exp 3	Exp 4
Average Timing (sec)	807	637	646	629

VI. Conclusion

In this paper, we explained our annotation tool and compared active learning techniques in an experiment with a baseline of random image selection. The results of this experiment on a vehicle detection and localization dataset show that the High-confidence technique is faster than the uncertainty and random technique and performs better with a smaller number of images (<500).

In our second experiment we tested our annotation tool with four annotators and we can conclude that the annotation tool in assisted mode with active learning is faster than an annotation tool in manual mode.

Acknowledgment

The ASGARD (“Analysis System for Gathered Raw Data”) project has received funding from the EU H2020 Secure Societies program (FCT-01-2015) under grant agreement number 700381. The InDeV (“In-Depth understanding of accident causation for Vulnerable road users”) project received funding from H2020-EU3.4 under grant agreement number 635895. Any opinions expressed in this paper do not necessarily reflect the views of the European Community. The Community is not liable for any use that may be made of the information contained herein.

REFERENCES

[1] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *IEEE CVPR*, pp. 6517-6525, 2017.

[2] W. Liu et al., "SSD: Single Shot MultiBox Detector," ECCV, pp. 21-37, 2016.

[3] K. Konyushkova, J. Uijlings, C. H. Lampert and V. Ferrari, "Learning Intelligent Dialogs for Bounding Box Annotation," in *IEEE CVPR*, pp. 9175-9184, 2018.

[4] G. Burghouts, K. Schutte, H. Bouma and R. den Hollander, "Selection of negative samples and two-stage combination of multiple features for action detection in thousands of videos," *Machine Vision Applications*, vol. 25(1), pp. 85-98, 2014.

[5] T. S. Huang et al., "Active learning for interactive multimedia retrieval," *Proc. IEEE,* vol. 96, no. 4, pp. 648-667, 2008.

[6] B. Settles, "Curious machines: Active learning with structured instances," *Thesis Univ. Wisconsin,* 2008.

[7] D. Lewis and W. Gale, "Training text classifiers by uncertainty sampling," *ACM SIGIR, pp. 3-12,* 1994.

[8] C. Snoek and M. Worring , "Concept-based video retrieval," *Foundations and Trends in Information Retrieval*, 2009.

[9] G. Schohn and D. Cohn, "Less is more: Active learning with support vector machines," in *ICML*, pp. 839-846, 2000.

[10] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *ICML*, pp. 107-118, 2001.

[11] M. Chen, M. Christel, A. Hauptmann and H. Wactlar, "Putting active learning into multimedia applications: dynamic definition and refinement of concept classifiers," in *Proc. ACM Int. Conf. on Multimedia*, pp. 902-911, 2005.

[12] G. Nguyen, M. Worring and A. Smeulders, "Interactive search by direct manipulation of dissimilarity space," *IEEE Trans. Multimedia,* 9(7), 1404-1415, 2007.

[13] S. Vijayanarasimhan and K. Grauman, "Large-scale live active learning: Training object detectors with crawled data and crowds," *IJCV*, vol. 108, pp. 97-114, 2014.

[14] X. Zhao and G. Ding, "Query expansion for object retrieval with active learning using BoW and CNN feature," *Multimedia Tools and Appl.,* 76(9), pp. 12133-12147, 2017.

[15] K.-S. Goh, E. Y. Chang and W.-C. Lai, "Multimodal concept-dependent active learning for image retrieval," in *Proc. ACM Int. Conf. on Multimedia*, pp. 564-571, 2004.

[16] H. Luan et al., "Segregated feedback with performance-based adaptive sampling for interactive news video retrieval," in *ACM Int. Conf. MM*, pp. 293-296, 2007.

[17] E. Gavves, T. Mensink, T. Tommasi, C. Snoek and T. Tuytelaars, "Active transfer learning with zero-shot priors: Reusing past datasets for future tasks," in *IEEE ICCV*, pp. 2731-2739, 2015.

[18] A. Holub, P. Perona and M. C. Burl, "Entropy-based active learning for object recognition," in *IEEE CVPR,* 2008.

[19] A. Kovashka, S. Vijayanarasimhan and K. Grauman, "Actively selecting annotations among objects and attributes," in *ICCV*, 2011.

[20] C. Vondrick and D. Ramanan, "Video annotation and tracking with active learning," in *Advances in Neural Information Processing Systems*, pp 28-36, 2011.

[21] S. Dasgupta and D. Hsu, "Hierarchical sampling for active learning," in *ICML*, pp. 208-215, 2008.

[22] X. Zhu, J. Lafferty and Z. Ghahramani, "Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions," in *ICML 2003 workshop*, 2003.

[23] Y. Geifman and R. El-Yaniv, *Deep Active Learning over the Long Tail.*, arXiv:1711.00941, 2017.

[24] K. Wang, D. Zhang, Y. Li, R. Zhang and L. Lin, "Cost-effective active learning for deep image classification," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 27, no. 12, pp. 2591-2600, 2017.

[25] S. Zhou, Q. Chen and X. Wang, "Active deep learning method for semi-supervised sentiment classification," *Neurocomputing,* vol. 120, pp. 536-546, 2013.

[26] F. Stark, C. Hazrbas, R. Triebel and D. Cremers, "Captcha recognition with active deep learning," in *GCPR Workshop on New Challenges in Neural Computation*, 2015.

[27] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," arXiv, 2018.

[28] Y. Gal, R. Islam and Z. Ghahramani, "Deep bayesian active learning with image data," *arXiv:1703.02910,* 2017.

[29] N. Houlsby, F. Huszár, Z. Ghahramani and M. Lengyel, "Bayesian active learning for classification and preference learning," *arXiv:1112.5745,* 2011.

[30] M. Ducoffe and F. Precioso, "Active learning strategy for CNN combining batch-wise Dropout and Query-By-Committee,," in *Proc. Europ. Symp. Artificial Neural Networks*, pp. 595-600, 2017.

[31] A. Kolesnikov and C. Lampert, "Improving weakly-supervised object localization by micro-annotation," *BMVC,* 2016.

[32] R. Cinbis, J. Verbeek and C. Schmid, "Weakly supervised object localization with multi-fold multiple instance learning," *IEEE Trans. PAMI,* vol. 39, no. 1, pp. 189-203, 2017.

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

[34] M. Everingham et al., "The Pascal visual object classes (voc) challenge," *Int. J. of Comp. Vision*, 88(2), pp. 303-338, 2010.

[35] T. Lin et al., "Microsoft COCO: Common objects in context.," in *ECCV*, pp. 740-755, 2014.

[36] A. Krizhevsky, I. Sutskever and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Adv. NIPS*, pp. 1097-1105, 2012.

[37] O. Russakovsky, L. Li and L. Fei-Fei, "Best of both worlds: Human-machine collaboration for object annotation", *IEEE CVPR*, pp. 2121-2131, 2015.

[38] D. P. Papadopoulos, J. R. R. Uijlings, F. Keller and V. Ferrari, "Extreme clicking for efficient object annotation," in *IEEE ICCV*, pp. 4940-4949, 2017.

[39] D. Papadopoulos, J. Uijlings, F. Keller and V. Ferrari, "We don't need no bounding-boxes: Training object class detectors using only human verification," in *IEEE CVPR*, 2016.

[40] H. Su, J. Deng and L. Fei, "Crowdsourcing annotations for visual object detection," in *AAAI Workshop*, 2012.

[41] H. Bouma et al., "Flexible image analysis for law enforcement agencies with deep neural networks to determine: where, who and what," in *Proc. SPIE*, vol. 10802, 2018.

[42] [Online]. Available: https://www.getnexar.com/challenge-2/.

[43] [Online]. [Accessed 03 2019] Available: https://www.indev-project.eu/InDeV/EN/Workpackages/WP_node.html.

[44] A. Nigam and K. McCallum, "Employing EM in pool-based active learning for text classification," *ICML,* 1998.

[45] M. Schervish, Theory of Statistics, Springer, 1995.