# Localization-based active learning (LOCAL) for object detection in 3D point clouds

Moses, Aimee, Jakkampudi, Srikanth, Danner, Cheryl, Biega, Derek

Aimee Moses, Srikanth Jakkampudi, Cheryl Danner, Derek Biega, "Localization-based active learning (LOCAL) for object detection in 3D point clouds," Proc. SPIE 12099, Geospatial Informatics XII, 1209907 (27 May 2022); doi: 10.1117/12.2618513

SPIE.

Event: SPIE Defense + Commercial Sensing, 2022, Orlando, Florida, United States

# LOCalization-Based Active Learning (LOCAL) for Object Detection in 3D Point Clouds

Aimee Moses, Srikanth Jakkampudi, Cheryl Danner, and Derek Biega

Expedition Technology, Inc, 13865 Sunrise Valley Drive, Suite 350, Herndon, VA USA, 20171

## ABSTRACT

Deep learning-based object detection and classification in 3D point clouds has numerous applications including defense, autonomous driving, and augmented reality. A challenge in applying deep learning to point clouds is the frequent scarcity of labeled data. Often, one must manually label a large quantity of data for the model to be useful in application. To overcome this challenge, active learning provides a means of minimizing the manual labeling required. The crux of active learning algorithms is defining and calculating the potential added "value" of labeling each unlabeled sample. We introduce a novel active learning algorithm, LOCAL, with an anchor-based object detection architecture, a modified object matching strategy, and an acquisition metric designed for object detection in any dimension. We compare the performance of common acquisition functions to our novel metric that utilizes all of the model outputs—including both bounding box localizations and softmax classification scores—to capture both the classification and spatial uncertainty in the model. Finally, we identify opportunities for further exploration, such as alternative measures of spatial uncertainty as well as increasing the stochasticity of the model in order to improve robustness of the algorithm.

**Keywords:** Active learning, deep learning, point clouds, object detection, uncertainty estimation

## 1. INTRODUCTION

The objective of the overall Intelligent Classification for 3D (IC3D) architecture and algorithms such as LOCAL is analysis and labeling of point clouds. The IC3D algorithm solves an object detection problem in the point cloud domain, operating on lidar and synthetic aperture radar (SAR) point clouds to output a rotated bounding box, class label, and confidence score for each detected object. The deployed system could aid analysts by prioritizing point clouds that are likely to contain objects of interest, reducing the tedious manual effort needed and shortening the time between new raw data becoming available and having actionable analysis. Towards this objective, we have developed a deep learning object detection model that can learn to recognize objects in 3D point cloud data. Because model quality depends heavily on training data, we pair our deep learning model with an active learning pipeline to help quickly build up a high-quality labeled dataset.

Active learning algorithms aim to identify the most informative unlabeled data in order to minimize the amount of data that must be manually annotated for satisfactory supervised training. In the typical application, a user with a large unlabeled dataset can begin by manually labeling a small subset of data, then iteratively train a model and apply active learning to determine which point clouds should be labeled next to provide the largest performance improvement. This is outlined in Figure 1.

The main variation amongst different active learning methods is in how they identify which data to label at each iteration. Most methods fall into two categories of how to define what makes data useful to label: uncertainty or diversity. Diversity methods look for the data that will be the most representative of the complete dataset with the least number of samples. These methods tend to utilize clustering[1] and finding exemplars in the data.[2–5] On the other hand, uncertainty methods try to identify which data the model is unsure about. Uncertainty methods assess the model's understanding of each input by examining its outputs from one or more forward passes of the model, while diversity methods assess how representative the inputs are via the model's learned feature representations. Although diversity methods show promising results, we chose to focus

Further author information: (Send correspondence to Aimee Moses)
Aimee Moses: E-mail: amoses@exptechinc.com

Geospatial Informatics XII, edited by Kannappan Palaniappan, Gunasekaran Seetharaman, Joshua D. Harguess, Proc. of SPIE Vol. 12099, 1209907 · © 2022 SPIE
0277-786X · doi: 10.1117/12.2618513

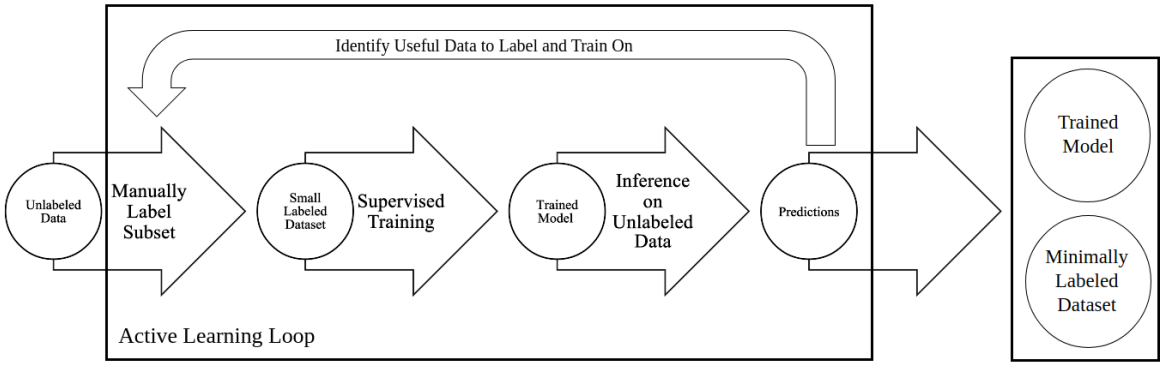


Figure 1. General structure of active learning algorithms: 1) supervised training on the available labeled data, 2) run inference with this model on the unlabeled pool of data, 3) identify the most informative unlabeled data points, 4) label those data, 5) repeat.

on uncertainty methods for ease of implementation with our existing object detector. Settles' "Active Learning Literature Survey"[6] has a comprehensive overview of both uncertainty and diversity methods, particularly those that are not specific to deep learning. There has been a surge of research and new methods in the last decade, going hand in hand with the development of deep learning.

While there are many active learning methods for deep learning classification problems, there are few for object detection and even fewer for object detection in 3D. Object detection generally includes the classification of detected objects, so most active learning methods for object detection adapt those designed for classification to the additional task of object localization. A popular approach to uncertainty metrics for classification is to use an ensemble of models or Monte Carlo dropout[7] to obtain multiple predictions for each input and combine or compare them when assessing the model's understanding of the input. Object detectors complicate this technique by outputting a variable number of objects detected for each input. To compare the predictions across multiple forward passes and obtain a single uncertainty score per input, one must either aggregate the objects' predictions before scoring or compute scores object-wise and then aggregate the scores. Additionally, object detectors output bounding box specifications that uncertainty metrics designed for classification do not incorporate.

In this paper, we propose a localization-based active learning framework, LOCAL, designed for object detection in any dimension. To our knowledge, there are only two existing papers on active learning for object detection in point clouds,[8,9] and both rely on additional RGB images for object detection. We build upon the high-achieving VoxelNet[10] to develop an object detector that requires only the 3D point clouds as inputs. We then modify an object matching algorithm to address the challenge of variable numbers of objects detected for the same input and enable one to capture this variability within traditional uncertainty metrics for classification. Finally, we propose a novel uncertainty metric that reflects the model's uncertainty in both classification and bounding box predictions.

We begin in Section 2 by summarizing the existing research on object detection in 3D and active learning. In Section 3, we detail our object detection model, object matching algorithm, and localization-based scoring function. We then outline our dataset and experiments testing our uncertainty metric against the standard metrics in Section 4. We conclude with a discussion of the results and conclusions in Sections 5 and 6.

## 2. RELATED WORKS

### 2.1 Object Detection in 3D Point Clouds

Deep learning models have revolutionized computer vision in the domain of 2D electro-optical (EO) images, achieving state-of-the-art results in classification, object detection, segmentation, and other tasks. Although the level of maturity lags behind that of the EO domain, deep learning solutions have similarly improved state-of-the-art performance in the point cloud domain in recent years. A ground-breaking paper in lidar object

detection is VoxelNet,[10] significant for its detection-centric approach using learned features. Later approaches built upon the success of the VoxelNet architecture, including SECOND,[11] which makes a number of improvements including loss function changes and a new sparse convolution implementation; and CenterPoint,[12] which applies the "anchorless" object detection approach from "Objects as Points"[13] to the VoxelNet and PointPillars[14] architectures.

Object detection performance for these and other approaches is commonly reported for one or more public datasets, such as the KITTI Vision Benchmark Suite[15] released in 2012, the Waymo Open Dataset[16] released in 2019, and the nuScenes dataset[17] released in 2019. These datasets, assembled to train and evaluate algorithms for autonomous driving, include lidar point clouds captured at ground level with annotated bounding boxes for vehicles, pedestrians, and other objects of interest. Voxel-based approaches were initially among the top performers on the KITTI leaderboard, won the Waymo 2021 3D detection challenges in 2020[18] and 2021,[19] and continue to be highly-ranked in the nuScenes detection task leaderboard.[12] While our objective is similar to those of these autonomous driving detection challenges, our point cloud data is collected by airborne sensors and has different characteristics compared to KITTI, Waymo, and nuScenes point clouds. There is likely significant overlap in architectures and techniques that lead to better performance on both overhead and ground-level point cloud object detection, but we have been benchmarking on the government-owned datasets that our algorithms ultimately need to perform on. Described in Section 3.1, our custom deep learning model is based on VoxelNet with modifications that led to improved performance on our datasets.

## 2.2 Active Learning

### 2.2.1 Active learning for classification

A key component of uncertainty approaches in active learning is the scoring, or acquisition, function. This scoring function assigns each data point a score for how well the model interprets that point, and it is the main difference between different approaches. For classification problems, one readily available resource is the predictive probabilities of class membership, generally obtained in deep learning via the softmax function. The simplest approach is to use these outputs directly as a proxy for model confidence. That is, choose to label the data whose maximum probabilities of class membership are the lowest.[20–22] In order to use more of the information available, other scoring functions use the predictive probabilities for other classes, not just the most probable. Margin sampling is the most naïve of these, in which the probability of the second-best class is subtracted from that of the best class. The smaller this difference, the less clarity the model has on how to treat the data.[21,23,24] Leveraging the softmax values as approximate posterior probabilities for a random variable, we can also calculate the Shannon entropy of the random variable's distribution.[25] Entropy is highest when every class is equally probable, and lowest when a single class is the clear favorite. It is widely used in recent work on active learning.[4,21,26] Entropy of the softmax vector is defined in Equation (1) for an input $x$, train data $D_{train}$, and output $y$ with possible classes $c \in C$.

$$\mathcal{H}(y|x, D_{train}) = -\sum_{c} \mathrm{P}(y = c|x, D_{train}) \log \mathrm{P}(y = c|x, D_{train}) \tag{1}$$

### 2.2.2 Bayesian active learning

All of the aforementioned acquisition functions rely on the class confidences as reliable indicators of model certainty. However, most models are not designed for that purpose and generally overestimate the confidence of their predictions.[27] One way to better calibrate the class confidences as measures of model uncertainty is Monte Carlo dropout. In Monte Carlo dropout, dropout layers are added to a neural network and left on at inference time. Inference-time dropout adds randomness to the forward pass and results in non-deterministic outputs. Gal and Ghahramani[7] showed that averaging the outputs of these stochastic inference runs is a probabilistically valid method of predicting model uncertainty. Their reasoning extends to other ways of introducing variation to the inference outputs such as inference time data augmentations.[22,28] Another approach is to train the same model multiple times with random initializations, creating an ensemble of models.[4] Ensembles can be used with machine learning methods other than deep learning, but they are computationally costly since you are required

to train the model multiple times at every iteration of active learning. For $T$ stochastic forward passes, let $M_t$ be the $t^{th}$ variation of the model. We define the average softmax scores as:

$$\mathrm{P}(\mathbf{y}|x, D_{train}) = \frac{1}{T}\sum_{t=1}^{T}\mathrm{P}(y|x, M_t). \tag{2}$$

Averaging the confidences of the predictions of an ensemble of models or a single model with stochasticity provides more robust and better calibrated measures of uncertainty and has improved the performance of existing uncertainty methods in active learning.[4,29,30] Instead of averaging, one can also compare the models' predictions for the same data. Samples whose predicted outputs vary heavily may lie near a decision boundary of the model and would be useful to label for better differentiation between the classes. One form of this is query by committee, in which each set of predictions counts as a "vote" in the committee. The inputs with the least agreement between the committee members are selected for labeling.[31–33] How disagreement among the committee is quantified varies, but a popular metric is the variation ratio—the ratio of the number of votes against the majority to the total.[4,26,34] Mutual information also finds the data with the most disagreement between the models, but it is able to incorporate the whole softmax vector from each model by building off entropy.[4,26,29,30,35] We can write it as the entropy of the average softmax vector minus the average of the entropy of the individual softmax vectors:

$$\begin{aligned}\mathcal{MI}(\mathbf{y}, \mathbf{M}|x, D_{train}) &= -\sum_{c}\mathrm{P}(\mathbf{y}=c|x, D_{train})\log\mathrm{P}(\mathbf{y}=c|x, D_{train}) \\ &\quad - \frac{1}{T}\sum_{t=1}^{T}\sum_{c} -\mathrm{P}(y=c|x, M_t)\log\mathrm{P}(y=c|x, M_t) \\ &= \mathcal{H}(\mathbf{y}|x, D_{train}) - \mathbb{E}_{\mathrm{p}(M|D_{train})}[\mathcal{H}(\mathbf{y}|x, M)].\end{aligned} \tag{3}$$

### 2.2.3 Estimating spatial uncertainty

For the problem of object detection, the active learning methods and metrics are less straightforward. Not only is there a variable number of predictions per image, but those predictions also go beyond softmax scores to include bounding box locations and dimensions. There are far fewer studies on active learning for object detection than for classification and only two to our knowledge for object detection in 3D. One approach to estimating uncertainty in object detectors is to ignore the spatial uncertainty of the bounding boxes and use the aforementioned metrics for classification uncertainty, aggregating across the boxes in the image.[36–38] Scores are aggregated by taking the average, sum, or maximum of the boxes' scores. Alternatively, for 2D object detection models that output probability maps for each class, one can begin to incorporate location by calculating classification uncertainty per pixel and then aggregating.[5] Aghdam et al.[39] went one step further by comparing softmax scores of adjacent pixels instead of across an ensemble, reasoning that objects span a neighborhood of pixels and those pixels should have similar outputs from the model.

Other authors have developed new metrics to address the uncertainty in object locations. Roy et al.[38] took what they call a "white box" approach and used the different convolution layers of their object detector as the committee members and comparing the confidence of overlapping predictions from each layer. Haussmann et al.[5] also utilized the model itself in one of their uncertainty calculations. They used the magnitude of the gradients of the output layer as an indicator of model stability, but it performed no better than the traditional methods of entropy and mutual information. Schmidt et al.[8] identified similar predictions across the inference outputs of an ensemble via Intersection over Union (IoU) in a method that they refer to as Region of Interest (RoI) Matching. They then used the entropy of the classification scores for each of the matched boxes. The authors also introduce a "consensus score" in which one minus the IoU of ensembles' predictions for each object are aggregated as the score itself. They found that RoI Matching with entropy had the most success, followed by the consensus score weighted by the classification variation ratios. In Kao et al.,[22] the authors used various combinations of classification confidence, localization tightness, and localization stability to define uncertainty. Localization tightness is another white box method in which they take the IoU of region proposals from the first

stage of their network and the final predictions from the second stage. For localization stability, they add various levels of Gaussian noise to the inputs and compute the IoU of the ordinary predictions with the noisy predictions, weighted by the classification score of the ordinary predictions. They found that combining localization stability or localization tightness with classification confidence were the most effective active learning strategies.

3D object detection presents its own challenges and there are only two papers we are aware of that have attempted active learning with the data. Feng et al.[9] used entropy and mutual information with each of Monte Carlo dropout and model ensembles to assess classification uncertainty. For the 3D component of their study, Schmidt et al.[8] used Monte Carlo dropout to obtain the classification variation ratio. Neither of these approaches addresses the spatial uncertainty in the regressed bounding box location and dimensions.

Additional strategies for estimating spatial uncertainty can be drawn from outside of the research specifically on active learning for object detection. Feng et al.[40] aimed to evaluate the uncertainty in 3D object detection for autonomous driving. They assess spatial uncertainty by the trace of the covariance matrix, or total variance, of the regressed bounding box dimensions. Miller et al.[41,42] also use the total variance of 2D bounding boxes along with a variety of clustering techniques when estimating epistemic spatial uncertainty in open-set detection. Using similar prediction matching strategies as Miller et al.,[41,42] Morrison et al.[43] and Blok et al.[44] multiply classification and spatial uncertainty to obtain one all-encompassing measure of uncertainty for instance segmentation. In the absence of regressed bounding box values, they use the average IoU of similar segmentation masks as their measure of spatial uncertainty.

## 3. METHODOLOGY

### 3.1 Object Detection Model

Since our 3D learned feature detection model is not the focus of this paper, we will provide a non-comprehensive overview of it in this section. Development of our deep learning object detection model began in late 2017, based on the VoxelNet paper[10] that was first released at that time. Building from that foundation, we experimented with many aspects of the model architecture and training process.

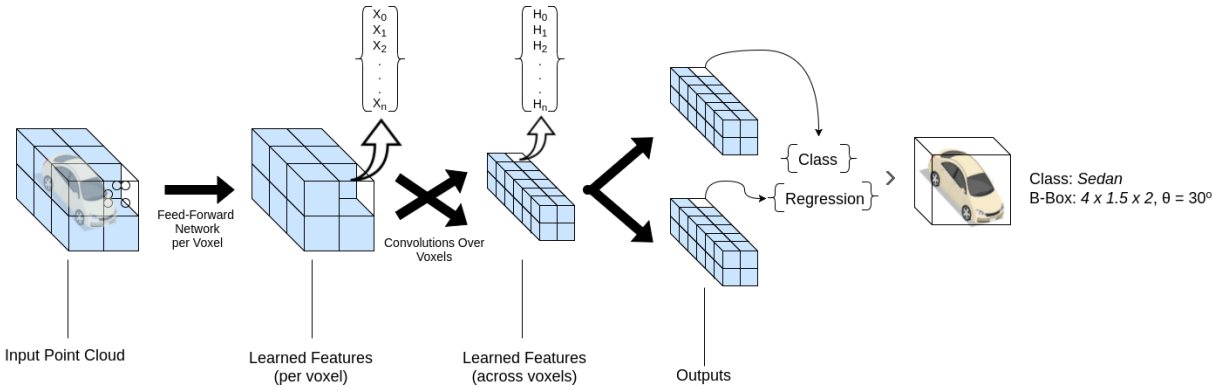


Figure 2. High-level depiction of a VoxelNet-based neural network architecture.

Although our voxel feature encoding module is based heavily on that described in VoxelNet, we found that comparable performance could be achieved with a less computationally intensive design. Our version of the Voxel Feature Encoding layer is consistent with the VoxelNet version in that it begins with a fully-connected layer, followed by batch norm and a ReLU activation. However, after max pooling, we do not concatenate the voxel-wise features and point-wise features, keeping the feature representation at the per-voxel level. As a result, the input to the next fully-connected layer is considerably smaller and more efficient.

Our data augmentation strategy differs entirely from that described in VoxelNet. Instead of applying all data augmentation operations on the fly, we precompute the voxelized features and ground truth anchors for each input example resulting in a significant speed-up in training time.

We achieved a large performance gain in validation set average precision at IoU 0.5 (AP50) by replacing the standard cross-entropy loss with focal loss, which is often employed to improve performance on datasets with class imbalance and/or high incidence of difficult examples. To reduce the risk of overfitting, we apply L2 weight regularization throughout the network and dropout to the voxel feature encoder.

When running inference, our model predicts a list of detections, where each detection has a vector of classification scores that sum to 1 across all classes (including a "background" class) and the following bounding box parameters: x, y, and z center; x, y, and z extent; and rotation angle (theta) in the x-y plane. The number of output predictions is fixed, but predictions with a background score above a threshold are discarded.

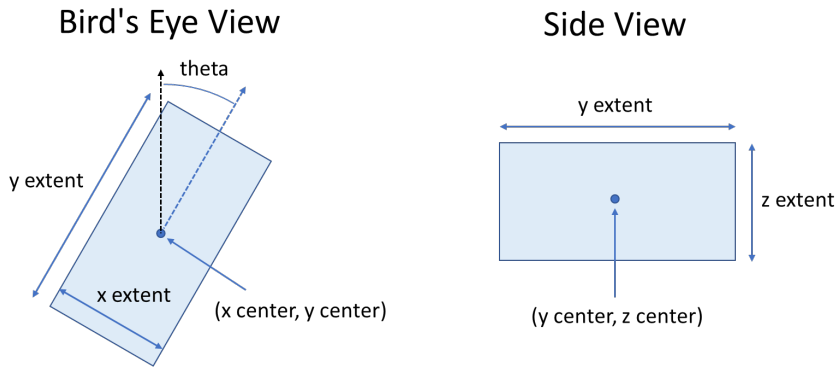


Figure 3. Two views depicting the seven parameters that we use to define a bounding box.

### 3.2 Object Matching Across Inference Runs

One challenge in the jump from active learning for classification problems to object detection is how to combine or compare outputs across stochastic forward passes. The majority of existing works have found ways to side-step this hurdle. Some do so by not using dropout or ensemble methods and relying on single-run metrics such as least confidence, margin sampling, and entropy.[36–38] In 2D object detection, others calculate uncertainty by pixel when using dropout or ensembles.[5,39] Kao et al.[22] and Roy et al.[38] designate one set of boxes to be the reference boxes and guarantee a corresponding box from the other inference results. Alternatively, Feng et al.[9] and Schmidt et al.[8] assume the same number of objects will be detected in each forward pass. Feng et al.[9] only have one object per image, so they can reduce the problem to location regression and classification without any object matching needed. Schmidt et al.[8] describe a matching process in which each prediction for every inference run is compared to the predictions in the other runs via IoU. Still, their method implies that every prediction has exactly one clear match from each of the other inference outputs.

However, aligning detected objects across multiple outputs is not unique to active learning for object detection. Miller et al.[42] explore methods for clustering detections from MC dropout to improve open-set object detection performance. In typical clustering, there are no restrictions on which data points can belong to the same clusters, but they want to restrict the clusters to having a maximum of one detection per forward pass. Miller et al.[42] suggest an exclusive Basic Sequential Algorithmic Scheme (BSAS) clustering with IoU thresholding for matching detections that complies with this restriction. Their method is used in Blok et al.[44] and Morrison et al.[43] for assessing uncertainty in their semantic segmentation models. After matching the segmentation masks across multiple forward passes with the exclusive BSAS clustering, they weigh the uncertainty score for each set of matches according to the number of forward passes with a segmentation mask in the set.

We extend the exclusive BSAS clustering to further our assessment of uncertainty. This method is flexible with respect to the number of detections output by an ensemble of models or models with stochastic outputs. Let $T$ be the number of forward passes performed on a particular scene. For each $t \in \{1, ..., T\}$, we calculate the IoU of each prediction, $r \in R_t$, with the predictions of each of the other inference outputs, $R_{t_0}$ for all $t_0 \in \{1, ..., T\}$ such that $t_0 \neq t$. We identify the best "match" from $R_{t_0}$ as the prediction with the highest IoU with $r$. In contrast to existing works, if there are no objects detected in the $t_0$ forward pass or none of the detections in

$R_{t_0}$ have sufficient overlap with $r$, we assign it a fabricated prediction, $\hat{r}$, as the match. After iterating through every detection from every forward pass, we are left with $\sum_{t=1}^{T} |R_t|$ objects, each of which is characterized by exactly $T$ detections.

**Algorithm 1:** Exclusive BSAS with Fabricated Detections

*threshold* $\in [0, 1]$
$S \leftarrow [\,]$
$\hat{r} \leftarrow$ fabricated detection
**for** *every forward pass* $t \in \{1, ..., T\}$ **do**
  $R_t \leftarrow$ detections from forward pass $t$;
  **for** *every detection* $r \in R_t$ **do**
    $s_r \leftarrow \{r\}$;
    **for** *every forward pass* $t_0 \in \{1, ..., T\}$, *s.t.* $t_0 \neq t$ **do**
      $IOU \leftarrow IOU(r, R_{t_0})$;
      **if** *any(IOU)* $\geq$ *threshold* **then**
        $s_r \leftarrow s_r \cup \{\underset{r_0 \in R_{t_0}}{\text{argmax}}\ IOU\}$;
      **else**
        $s_r \leftarrow s_r \cup \{\hat{r}\}$ ;
      **end**
    **end**
    $S \leftarrow S + s_r$;
  **end**
**end**
**return** *List of matched detection sets, S;* $|S|$ = *number of objects,* $|s_i| = T\ \forall\ s_i \in S$

To use this matching method with uncertainty metrics based on the softmax vector, we used a uniform distribution for the fabricated matches. When taking the entropy of the average softmax vector across matched detections, each fabricated uniform softmax vector increases the uncertainty score of the scene, reflecting the disagreement between the outputs. Mutual information is even better suited to this matching method because it penalizes outliers in the committee rather than unconfident predictions. If a prediction's matches are equally falsified and real, it will have high mutual information and reflect that inconsistency.

### 3.3 Localization-Based Uncertainty Metric

One key aspect of the uncertainty of an object detector is the spatial variation in its bounding box predictions. Though we use these values for object matching across inference runs, the existing works on active learning in object detection compare only the softmax values or IoU of the matches. We aim to incorporate the spatial uncertainty of the detected boxes themselves when assessing the models' understanding of the point cloud. Drawing from Miller et al.[41,42] and Feng et al.,[40] we adopt the total variance of the matched bounding boxes to account for spatial uncertainty. However, while they treated it as a separate assessment of uncertainty, we combine it with mutual information or entropy to form a holistic metric capturing both spatial and classification uncertainty.

For each object, we obtained the total variance across the bounding box dimensions of the matched detections as well as the mutual information or entropy of their softmax vectors. The total variance and the classification uncertainty metric of choice are each summed across the boxes in the tile, then normalized to [0, 1] and combined. We define the full acquisition function in Equations (4) and (5).

Let $X$ be the set of all unlabeled samples and $S_i$ be the set of objects found via a matching scheme for sample $x_i \in X$. Let $Z_s \in \mathbb{R}^{NxM}$ be the regressed bounding box values of the detections for object $s$ from each forward pass, where $N$ is the number detections for the object and $M$ is the number of output values characterizing the bounding boxes. If using our matching scheme from Section 3.2, $N$ will be exactly the number of forward passes performed.

$$\text{Total Variance for Sample } x_i = \sum_{s=0}^{S_i} \text{Trace}(\text{Cov}(Z_s)), \tag{4}$$

$$\begin{aligned}\text{LOCAL Acquisition Function} = &\frac{\text{Total Variance for } x_i}{\max\limits_{x_j \in X} \text{Total Variance for } x_j} + \frac{\text{Classification Uncertainty of } x_i}{\max\limits_{x_j \in X} \text{Classification Uncertainty of } x_j}, \\ &\text{where Classification Uncertainty} = \mathcal{H}(\mathbf{y}|x_i, D_{train}) \text{ or } \mathcal{MI}(\mathbf{y}, \mathbf{M}|x_i, D_{train}).\end{aligned} \tag{5}$$

# 4. EXPERIMENTS

## 4.1 Dataset

We trained our object detector and assessed various active learning methods on a government-owned dataset. The dataset consists of 41 LiDAR point clouds split into train, validation, and test sets with 25 point clouds in train, 9 in validation, and the remaining 7 in test. We further divided the train set for active learning as described in Section 4.2. The dataset contains 7 different classes of objects, totaling approximately 7900 individual objects. The distribution of classes is in Figure 4. Before inputting the point clouds into the object detector, we cut them into approximately 11,000 overlapping tiles in the xy-plane and used a single tile for each batch. Not only did this improve the efficiency of our object detector, but it also allowed us to choose specific regions in the larger point clouds to label during active learning.

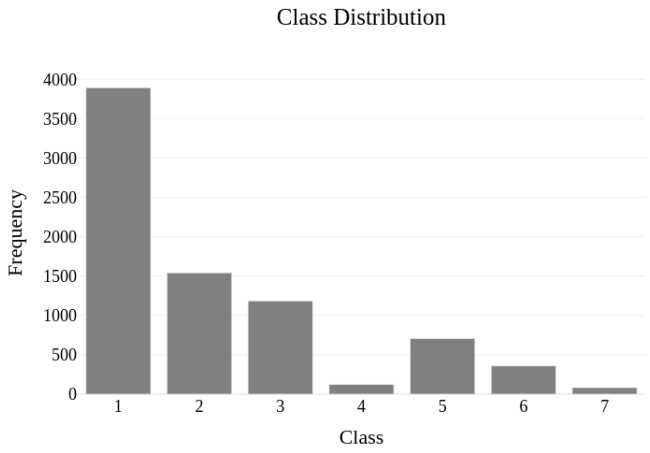


Figure 4. Histogram of object classes demonstrating significant class imbalance

## 4.2 Experimental Design

Our object detector was trained from scratch for a constant number of epochs after every iteration of active learning. We chose 100 epochs based on where performance began to plateau in training runs on the full labeled dataset. Prior knowledge of our dataset indicated that the majority of the point cloud tiles would not contain any objects of interest. In order to avoid overfitting the model to background-only tiles, we capped the number of point cloud scenes to train on at about a quarter of the total train set. As a result, the initial model was trained on approximately 5% of the total train set, and an additional 5% was added for 4 iterations of the active learning loop.

With this structure, we used the following uncertainty metrics as well as a random control: Least Confidence, Entropy, Entropy of Matched Objects, Mutual Information, Total Variance + Entropy of Matched Objects, Total Variance + Mutual Information. For all conditions other than Least Confidence and Entropy, we performed three

forward passes with dropout at scoring time and used our object matching method outlined in Section 3.2. In order to evaluate the contribution of the fabricated detections, we ran additional trials of each of the total variance metrics without adding fabricated detections.

For Least Confidence, we followed the existing literature and averaged the confidence scores of the detections in each tile. In the remaining conditions, we aggregated the scores across objects via summation. Scenes with no detections in any of the forward passes were marked as least uncertain with scores of zero. As discussed in Brust et al.,[36] this approach considers scenes with more objects to be more informative. Though this increases the number of objects manually labeled per iteration of the active learning loop, it minimizes the number of scenes that ultimately need to be annotated and the number of iterations necessary to achieve sufficient performance. Since we expected a large number of background-only tiles in our dataset, we decided to prioritize obtaining the smallest subset that could be used to train a satisfactory object detector.

### 4.3 Results

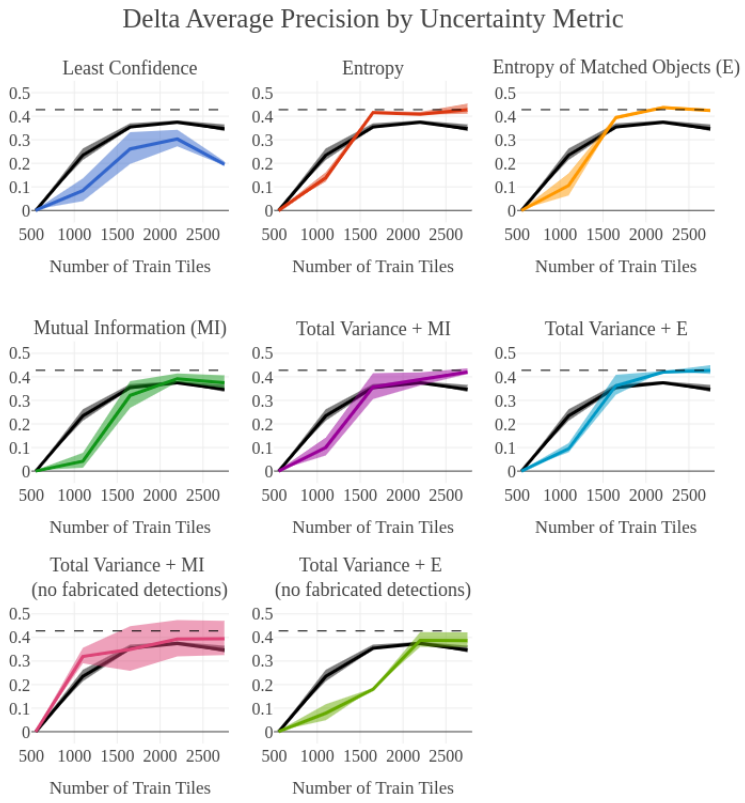


Figure 5. Improvement in object detector performance from the initial trained model over the course of active learning using different uncertainty metrics. All are compared to the random control (black shaded lines) and the best performance across the metrics after the final training iteration (black dotted line).

The choice of uncertainty metric impacts the effectiveness and efficiency of the active learning process in improving model performance. Figure 5 illustrates how various uncertainty metrics impact the progression of model performance during active learning. Each of the tested uncertainty metrics are compared to a baseline where samples are randomly selected to be added to the labeled training dataset. These uncertainty metrics rely on a performant model to give meaningful information and, as a result, the random selection of samples outperforms most of the metrics for the first one or two iterations of active learning. As the process continues and the model improves on its performance, the metrics, apart from least confidence, become more informative

and ultimately outperform the naïve random selection. We use average precision at IoU 0.5 (AP50) to measure model performance, and our figures and tables report the increase in AP50 from the baseline value.

Table 1. Change in object detector performance from the initial trained model to after final iteration of active learning with each uncertainty metric, averaged over two independent trials. Final models were trained on 2750 tiles, or approximately 25% of the total train set.

| | ΔAP50 | # Non-Empty Tiles Added | # Annotations Added |
|---|---|---|---|
| Random | 0.347 | 353 | 1008 |
| Least Confidence | 0.196 | 300.5 | 659.5 |
| Entropy | 0.428 | 1068 | 3785 |
| Entropy of Matched Objects/E | 0.424 | 783 | 2998.5 |
| Mutual Information/MI | 0.375 | 668 | 2427 |
| Total Variance + E (LOCAL) | 0.427 | 787.5 | 3048.5 |
| Total Variance + MI (LOCAL) | 0.420 | 560.5 | 2096 |
| Total Variance + E, no fabricated detections | 0.386 | 810 | 2904.5 |
| Total Variance + MI, no fabricated detections | 0.394 | 961 | 3340 |

Table 2. Object detector performance gain per non-empty tile and annotation added.

| | ΔAP50 per Non-Empty Tile | ΔAP50 per Annotation |
|---|---|---|
| Random | 0.000983 | 0.000344 |
| Least Confidence | 0.000651 | 0.000297 |
| Entropy | 0.000401 | 0.000113 |
| Entropy of Matched Objects/E | 0.000542 | 0.000142 |
| Mutual Information/MI | 0.000562 | 0.000155 |
| Total Variance + E (LOCAL) | 0.000542 | 0.000140 |
| Total Variance + MI (LOCAL) | 0.000750 | 0.000200 |
| Total Variance + E, no fabricated detections | 0.000477 | 0.000133 |
| Total Variance + MI, no fabricated detections | 0.000410 | 0.000118 |

As shown in Table 1, using entropy in the uncertainty metric to inform active learning resulted in the highest final model average precision. Entropy of detections from a single forward pass had the best performance and drove the active learning process to identify a set of samples for manual labeling containing 3785 additional object annotations. In comparison, entropy of matched objects and total variance + entropy of matched objects, which had the next best average precision gains, resulted in manually labeling sample sets on average of 2998.5 and 3048.5 annotations respectively. Naturally, it stands to reason that adding more object labels to the training set will improve model performance, but the goal of active learning is to prioritize labeling the most informative samples. Table 2 shows efficiency metrics measuring average precision gain per added non-empty sample and per

added object label. Of the models that performed better than the baseline, those trained using total variance + mutual information to drive active learning resulted in the most efficient process on both a per sample and per object basis.

The addition of total variance to the entropy of matched objects had minimal effect on performance and the characteristics of the samples chosen for labeling. However, for mutual information adding total variance improved the metrics across the board. The results in Table 1 also show that adding fabricated detections for inconsistent objects across forward passes contributed to the performance gained through active learning. While there were comparable performance gains per object with and without fabricated detections, there were more non-empty samples selected and decreased performance per sample when fabricated detections were not used.

## 5. DISCUSSION

Comparison of the tested metrics comes with certain caveats. Entropy and least confidence are calculated using the results of a single inference run while the remaining metrics combine the results of multiple inference runs via object matching. This means that when comparing entropy with total variance + mutual information for example, there are changes resulting from two variables: the calculation of the uncertainty metric and the matching scheme used. We begin to control for matching scheme by testing the total variance metrics with traditional BSAS clustering in addition to the extended exclusive BSAS clustering approach detailed in Section 3.2.

In traditional BSAS clustering without fabricated detections, the number of detections per identified object is not guaranteed to be constant. In particular, when a detection from one forward pass does not have sufficient IoU with any detections from the other forward passes, we will have an object defined by only one detection. In this case, both the total variance and the mutual information of the object are zero, despite the apparent inconsistency in the inference outputs. Thus, we expect the total variance metrics to be more informative with the inclusion of fabricated detections, especially for total variance + mutual information. As noted in Section 4.3, this is just what we observe in the performance of the models from the final iterations.

Curiously, Figure 5 shows that using total variance + mutual information without fabricated detections was the only condition in which the performance of the model after the first iteration of active learning was higher than the random baseline. One possible explanation is this condition's tendency to designate samples as less informative when they lacked overlap between detections from different forward passes. This likely resulted in the algorithm giving preference to the selection of samples containing well-defined objects. Those samples may be more valuable at the early stages than those with the hard-to-detect objects laying close to classification decision boundaries that we generally prioritize.

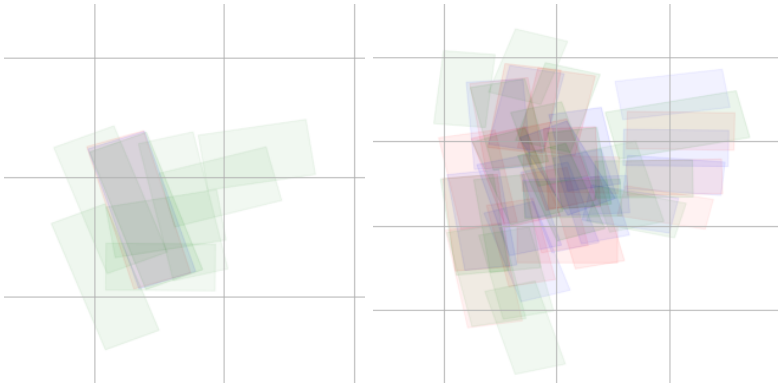

Figure 6. Birds-eye-view of detected bounding boxes from 3 forward passes (red, blue, and green); classification confidence is reflected in the opacity of the boxes. All metrics aside from least confidence score the tile on the right as more uncertain, while least confidence scores the tile on the left as more uncertain.

The least confidence uncertainty metric notably performed worse than simply selecting random samples for manual labeling. This metric resulted in the selection of the fewest tiles containing annotations. The model's

poor performance follows from this since it does not have enough examples to reason over. It is likely that the aggregation scheme used for this metric is causing an over-selection of empty tiles. Unlike the other metrics, uncertainty was averaged across detections as opposed to summed. As a result, a single confident false positive detection would result in the selection of an empty tile. Since the model is not performant to begin with, these poor detections are more likely which, in turn, results in poor selection criteria for the next loop. The model then repeats this flawed process at every iteration resulting in performance worse than the random baseline. Additionally, least confidence does not make use of the information available in the softmax vector nor the differences between multiple inference runs. Figure 6 shows an example where the least confidence metric disagrees with the other metrics when flagging which of the two tiles is more uncertain. The other metrics all select the tile containing more detections, further supporting the theory that the discrepancy in aggregation techniques is contributing to this poor performance.

Both the calculation and interpretation of uncertainty metrics need to be informed by the annotation workflow. Providing the model with more object annotations to train on will generally result in better performance, but how those annotations are distributed can have a large impact on the time it takes for both training and labeling. If the only consideration is maximizing model performance, the results from Table 1 suggest the use of entropy as an uncertainty metric. There is a cost associated with having to manually label more objects and samples, however. When we sum uncertainty of predictions within a tile, a premium is put on tiles containing more objects. This calculation decision impacts training time as well as the manual labeling workflow. Since tiles with more objects are inherently prioritized, the model will train on fewer tiles and the annotator will label fewer tiles at each iteration. Instead, one might choose to use a different aggregation of uncertainty within tiles such as maximum or average, which would likely increase the number of tiles for training and labeling. Whether that trade-off is worthwhile is dependent on how effortful it is for an annotator to switch between tiles to annotate relative to simply labeling more objects within a tile.

Another consideration with respect to the efficiency of the overall active learning loop is the number of forward passes required for the uncertainty calculation. This aspect would have had a more pronounced impact if we had used an ensemble of models instead of inference-time dropout, however it still contributed to the overall computation time. We observed that using the entropy of objects from a single forward pass and using the entropy of the Monte Carlo averaged softmax vectors of matched detections led to similar model performance. However, the condition with multiple forward passes selected far fewer non-empty tiles and total objects to annotate, leading to a higher performance gain to annotation effort ratio as seen in Table 2. Contrary to our expectations, the condition using matched objects had more variation between the two trials than the single forward pass.

Entropy outperformed the other metrics in terms of the average precision of the final training iteration. This implies that the model's classification confidence was a good approximation of its epistemic uncertainty, while the variation produced by dropout that is captured with mutual information and spatial variance was less informative. One contributing factor to this outcome could be the amount of stochasticity in the model. Blok et al.[44] found that dropout probability had a large impact on the performance of their active learning algorithm for instance segmentation, suggesting that our methods would benefit from increasing the number of dropout layers in the model and the probability of dropout. Additionally, a greater quantity of forward passes during the uncertainty calculations would better reflect the underlying probability distributions and may increase the information gained from the uncertainty metrics assessing variation within the distribution. These modifications may also add stability to those metrics and decrease the variation between trials.

Entropy captured enough of the model uncertainty, that it saw only a small performance improvement with the addition of bounding box total variance. It is possible that putting more weight on the entropy or spatial uncertainty may lead to more impact on performance. On the other hand, total variance + mutual information well outperformed mutual information alone. The total variance condition led to higher final average precision despite adding fewer non-empty tiles and total annotated objects. This suggests that incorporating spatial uncertainty via total variance of the bounding box outputs encouraged the algorithm to select more informative samples and objects. Looking closely at how the two metrics ranked the tiles at each iteration, we saw that they agreed most of the time. In theory, they disagreed only when the classification uncertainty captured by

mutual information did not coincide with the spatial uncertainty. Adding total variance enabled the algorithm to consider this element of model uncertainty that was otherwise missed.

## 6. CONCLUSION

The availability of labeled 3D point clouds is limited, and active learning is an invaluable tool for making the most out of the time and resources spent on manual annotation. In this paper, we aimed to develop LOCAL, an active learning scheme designed for object detection in any dimension. First, we reviewed the existing works on object detection in 3D, active learning, and both classification and spatial uncertainty estimation. Based on the prior research, we expanded a sequential matching scheme across multiple stochastic forward passes to identify objects and compare model outputs for those objects. We explored the performance of various common uncertainty measures for active learning with a VoxelNet-based 3D object detector on a new point cloud dataset. Of these metrics, Shannon entropy identified the point cloud tiles that resulted in the best object detector performance. We observed that uncertainty metrics that combined or compared outputs across multiple forward passes with dropout and used our extended object matching scheme found tiles with fewer, more informative objects. To asses our modified matching method, we demonstrated that the original method was less efficient in its selection of non-empty tiles, leading to worse performance of the final model. Additionally, we tested a new metric combining spatial and classification uncertainty and compared its efficacy with the existing metrics. We found that adding a measure of spatial uncertainty to the standard classification uncertainty metrics was beneficial both for maximizing model performance and for minimizing the manual labeling required, particularly when using mutual information to estimate the classification uncertainty.

We did not fully optimize the parameters of our conditions nor run as many trials of each as would be necessary to characterize experimental variance. We leave this as future work. Additionally, the small size of our dataset and the large proportion of background-only tiles presented a challenge for obtaining high performance by our object detector. Further experimentation is needed to fully explore the effects of both matching strategies and uncertainty metrics on active learning in object detection. Future work may investigate alternative approaches to approximating the underlying distribution of the outputs, such as model ensembles and test-time data augmentations. Other measures of dispersion may also be beneficial in place of total variance in the estimation of spatial uncertainty.

Given how aspects of LOCAL were inspired by research on other deep learning problems, one can extrapolate that our work may also be applied beyond active learning for object detection. Our approach of measuring the dispersion of the output distribution may be utilized to estimate the uncertainty of any deep learning model that outputs stochastic regressed values. Furthermore, incorporating both the spatial and classification uncertainty of an object detector in one metric has applications not only for active learning in open-set recognition, semi-supervised learning, reinforcement learning, and more.

## ACKNOWLEDGMENTS

This work was supported by the US Air Force Research Laboratory and the US National Geospatial-Intelligence Agency.

## REFERENCES

[1] Nguyen, H. T. and Smeulders, A., "Active learning using pre-clustering," in [*Proceedings of the twenty-first international conference on Machine learning*], 79 (2004).

[2] Sener, O. and Savarese, S., "Active learning for convolutional neural networks: A core-set approach," *arXiv preprint arXiv:1708.00489* (2017).

[3] Yang, L., Zhang, Y., Chen, J., Zhang, S., and Chen, D. Z., "Suggestive annotation: A deep active learning framework for biomedical image segmentation," in [*International conference on medical image computing and computer-assisted intervention*], 399–407, Springer (2017).

[4] Beluch, W. H., Genewein, T., Nürnberger, A., and Köhler, J. M., "The power of ensembles for active learning in image classification," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 9368–9377 (2018).

[5] Haussmann, E., Fenzi, M., Chitta, K., Ivanecky, J., Xu, H., Roy, D., Mittel, A., Koumchatzky, N., Farabet, C., and Alvarez, J. M., "Scalable active learning for object detection," in [*2020 IEEE intelligent vehicles symposium (iv)*], 1430–1435, IEEE (2020).
[6] Settles, B., "Active learning literature survey," (2009).
[7] Gal, Y. and Ghahramani, Z., "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in [*international conference on machine learning*], 1050–1059, PMLR (2016).
[8] Schmidt, S., Rao, Q., Tatsch, J., and Knoll, A., "Advanced active learning strategies for object detection," in [*2020 IEEE Intelligent Vehicles Symposium (IV)*], 871–876, IEEE (2020).
[9] Feng, D., Wei, X., Rosenbaum, L., Maki, A., and Dietmayer, K., "Deep active learning for efficient training of a lidar 3d object detector," in [*2019 IEEE Intelligent Vehicles Symposium (IV)*], 667–674, IEEE (2019).
[10] Zhou, Y. and Tuzel, O., "Voxelnet: End-to-end learning for point cloud based 3d object detection," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 4490–4499 (2018).
[11] Yan, Y., Mao, Y., and Li, B., "Second: Sparsely embedded convolutional detection," *Sensors* **18**(10), 3337 (2018).
[12] Yin, T., Zhou, X., and Krahenbuhl, P., "Center-based 3d object detection and tracking," in [*Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*], 11784–11793 (2021).
[13] Zhou, X., Wang, D., and Krähenbühl, P., "Objects as points," *arXiv preprint arXiv:1904.07850* (2019).
[14] Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., and Beijbom, O., "Pointpillars: Fast encoders for object detection from point clouds," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 12697–12705 (2019).
[15] Geiger, A., Lenz, P., and Urtasun, R., "Are we ready for autonomous driving? the kitti vision benchmark suite," in [*2012 IEEE conference on computer vision and pattern recognition*], 3354–3361, IEEE (2012).
[16] "Waymo open dataset: An autonomous driving dataset," (2019).
[17] Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., and Beijbom, O., "nuscenes: A multimodal dataset for autonomous driving," in [*Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*], 11621–11631 (2020).
[18] Ding, Z., Hu, Y., Ge, R., Huang, L., Chen, S., Wang, Y., and Liao, J., "1st place solution for waymo open dataset challenge–3d detection and domain adaptation," *arXiv preprint arXiv:2006.15505* (2020).
[19] Hu, Y., Ding, Z., Ge, R., Shao, W., Huang, L., Li, K., and Liu, Q., "Afdetv2: Rethinking the necessity of the second stage for object detection from point clouds," *arXiv preprint arXiv:2112.09205* (2021).
[20] Lewis, D. D. and Gale, W. A., "A sequential algorithm for training text classifiers," in [*SIGIR'94*], 3–12, Springer (1994).
[21] Wang, K., Zhang, D., Li, Y., Zhang, R., and Lin, L., "Cost-effective active learning for deep image classification," *IEEE Transactions on Circuits and Systems for Video Technology* **27**(12), 2591–2600 (2016).
[22] Kao, C.-C., Lee, T.-Y., Sen, P., and Liu, M.-Y., "Localization-aware active learning for object detection," in [*Asian Conference on Computer Vision*], 506–522, Springer (2018).
[23] Roth, D. and Small, K., "Margin-based active learning for structured output spaces," in [*European Conference on Machine Learning*], 413–424, Springer (2006).
[24] Joshi, A. J., Porikli, F., and Papanikolopoulos, N., "Multi-class active learning for image classification," in [*2009 ieee conference on computer vision and pattern recognition*], 2372–2379, IEEE (2009).
[25] Shannon, C. E., "A mathematical theory of communication," *The Bell system technical journal* **27**(3), 379–423 (1948).
[26] Gal, Y., Islam, R., and Ghahramani, Z., "Deep bayesian active learning with image data," in [*International Conference on Machine Learning*], 1183–1192, PMLR (2017).
[27] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q., "On calibration of modern neural networks," in [*International Conference on Machine Learning*], 1321–1330, PMLR (2017).
[28] Ayhan, M. S. and Berens, P., "Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks," (2018).
[29] Pop, R. and Fulop, P., "Deep ensemble bayesian active learning: Addressing the mode collapse issue in monte carlo dropout via ensembles," *arXiv preprint arXiv:1811.03897* (2018).

[30] Atighehchian, P., Branchaud-Charron, F., and Lacoste, A., "Bayesian active learning for production, a systematic study and a reusable library," *arXiv preprint arXiv:2006.09916* (2020).

[31] Seung, H. S., Opper, M., and Sompolinsky, H., "Query by committee," in [*Proceedings of the fifth annual workshop on Computational learning theory*], 287–294 (1992).

[32] Melville, P. and Mooney, R. J., "Diverse ensembles for active learning," in [*Proceedings of the twenty-first international conference on Machine learning*], 74 (2004).

[33] Ducoffe, M. and Precioso, F., "Active learning strategy for cnn combining batchwise dropout and query-by-committee.," in [*ESANN*], (2017).

[34] Freeman, L. C., "Elementary applied statistics. new york: Johnwiley and sons," *Inc., 19t* **6** (1965).

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

[36] Brust, C.-A., Käding, C., and Denzler, J., "Active learning for deep object detection," *arXiv preprint arXiv:1809.09875* (2018).

[37] Desai, S. V., Chandra, A. L., Guo, W., Ninomiya, S., and Balasubramanian, V. N., "An adaptive supervision framework for active learning in object detection," *arXiv preprint arXiv:1908.02454* (2019).

[38] Roy, S., Unmesh, A., and Namboodiri, V. P., "Deep active learning for object detection.," in [*BMVC*], 91 (2018).

[39] Aghdam, H. H., Gonzalez-Garcia, A., Weijer, J. v. d., and López, A. M., "Active learning for deep detection neural networks," in [*Proceedings of the IEEE/CVF International Conference on Computer Vision*], 3672–3680 (2019).

[40] Feng, D., Rosenbaum, L., and Dietmayer, K., "Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection," in [*2018 21st International Conference on Intelligent Transportation Systems (ITSC)*], 3266–3273, IEEE (2018).

[41] Miller, D., Nicholson, L., Dayoub, F., and Sünderhauf, N., "Dropout sampling for robust object detection in open-set conditions," in [*2018 IEEE International Conference on Robotics and Automation (ICRA)*], 3243–3249, IEEE (2018).

[42] Miller, D., Dayoub, F., Milford, M., and Sünderhauf, N., "Evaluating merging strategies for sampling-based uncertainty techniques in object detection," in [*2019 International Conference on Robotics and Automation (ICRA)*], 2348–2354, IEEE (2019).

[43] Morrison, D., Milan, A., and Antonakos, E., "Uncertainty-aware instance segmentation using dropout sampling," in [*Proceedings of the Robotic Vision Probabilistic Object Detection Challenge (CVPR 2019 Workshop), Long Beach, CA, USA*], 16–20 (2019).

[44] Blok, P. M., Kootstra, G., Elghor, H. E., Diallo, B., van Evert, F. K., and van Henten, E. J., "Active learning with maskal reduces annotation effort for training mask r-cnn," *arXiv preprint arXiv:2112.06586* (2021).