Active Learning for Deep Object Detection via Probabilistic Modeling

Jiwoong Choi\textsuperscript{1,3}, Ismail Elezi\textsuperscript{2,3}, Hyuk-Jae Lee\textsuperscript{1}, Clement Farabet\textsuperscript{3}, and Jose M. Alvarez\textsuperscript{3}
\textsuperscript{1}Seoul National University, \textsuperscript{2}Technical University of Munich, \textsuperscript{3}NVIDIA

\{jwchoi, hjlee\}@capp.snu.ac.kr, ismail.elezi@tum.de, \{cfarabet, josea\}@nvidia.com

Abstract

Active learning aims to reduce labeling costs by selecting only the most informative samples on a dataset. Few existing works have addressed active learning for object detection. Most of these methods are based on multiple models or are straightforward extensions of classification methods, hence estimate an image’s informativeness using only the classification head. In this paper, we propose a novel deep active learning approach for object detection. Our approach relies on mixture density networks that estimate a probabilistic distribution for each localization and classification head’s output. We explicitly estimate the aleatoric and epistemic uncertainty in a single forward pass of a single model. Our method uses a scoring function that aggregates these two types of uncertainties for both heads to obtain every image’s informativeness score. We demonstrate the efficacy of our approach in PASCAL VOC and MS-COCO datasets. Our approach outperforms single-model based methods and performs on par with multi-model based methods at a fraction of the computing cost. Code is available at https://github.com/NVlabs/AL-MDN.

1. Introduction

The performance of deep detection networks is dependent on the size of the labeled data [31, 32]. Motivated by this, researchers have explored smart strategies to select the most informative samples in the dataset for labeling, known as active learning [35]. Typically, this is done by devising a scoring function that computes the network’s uncertainty, selecting to label the samples for which the network is least confident with regard to its predictions [2, 4, 40].

In general, the predictive uncertainty is decomposed into aleatoric and epistemic uncertainty [15, 20]. The former refers to the inherent noise in the data, such as sensor noise, and can be attributed to occlusions or lack of visual features [10, 24]. The latter refers to the uncertainty caused by the lack of knowledge of the model and is inversely proportional to the density of training data [38]. Modeling and distinguishing these two types of uncertainty is very important in active learning, as it allows the deep learning models to know about their limitations [21, 38], i.e., recognize suspicious predictions in the sample (aleatoric uncertainty) and recognize samples that do not resemble the training set (epistemic uncertainty). To compute these types of uncertainty, researchers use multi-model based approaches, such as ensembles [2] or Monte Carlo (MC) dropout [13]. These methods reach good results but come with several limitations [11, 16]. In particular, by being multi-models, they require a much higher computing cost, and in the case of ensembles, they also increase the number of the network’s parameters [2]. Additionally, they rely only on classification uncertainty, totally ignoring the localization uncertainty.

In this paper, we propose a novel active learning approach for deep object detection. Our approach uses a single model with a single forward pass, significantly reducing the computing cost compared to multiple model-based methods. Despite this, our method reaches high accuracy. To manage so, our method utilizes both localization and classification-based aleatoric and epistemic uncertainties. As shown in Fig. 1, we base our method on a mixture density networks [3] that learns a Gaussian mixture model (GMM) for each of the network’s outputs, i.e., localization and classification, to compute both aleatoric and
epistemic uncertainties. To efficiently train the network, we propose a novel deep active learning method for object detection that leverages the aleatoric and epistemic uncertainties, by considering both the localization and classification information. Our method is efficient and uses a single forward pass in a single model.

We propose a novel loss to train the GMM-based object detection network that leads to overall performance improvements in the network.

We demonstrate the effectiveness of our approach using different models on two different datasets.

2. Related Work

Deep active learning for object detection has recently acquired interest. The work of [16] trains an ensemble [2] of neural networks and then selects the samples with the highest score defined by some acquisition function, i.e., entropy [36] or mutual information [5]. Concurrent work [11] explores similar directions, but by approximating the uncertainty via MC-dropout [12, 26]. The work of [1] presents a method of calculating pixel scores and using them for selecting informative samples. Another approach [33] proposes a query by committee paradigm to choose the set of images to be queried. The work of [34] uses the feature space to select representative samples in the dataset, reaching good performance in object detection [40]. A different solution was given by [23] where the authors define two different scores: localization tightness which is the overlapping ratio between the region proposal and the final prediction; and localization stability that is based on the variation of predicted object locations when input images are corrupted by noise. In all cases, images with the highest scores are chosen to be labeled. The state-of-the-art (SOTA) method of [40] offers a heuristic but elegant solution while outperforming the other single model-based methods.

3. Active Learning for Object Detection

The key novelty of our approach is designing the output layers of the neural network to predict a probability distribution, instead of predicting a single value for each output of the network (see Fig. 2a). To this end, we propose to make use of a mixture density network where the output of the network consists of the parameters of a GMM: the mean $\mu^k$, the variance $\Sigma^k$, and the mixture weight $\pi^k$ for the $k$-th component of the GMM. Given these parameters, we can estimate the aleatoric $u_{al}$ and epistemic $u_{ep}$ uncertainties [8]:

$$u_{al} = \sum_{k=1}^{K} \pi^k \Sigma^k, \quad u_{ep} = \sum_{k=1}^{K} \pi^k \| \mu^k - \sum_{i=1}^{K} \pi^i \mu^i \|^2, \quad (1)$$

where $K$ is the number of components in the GMM.
3.1. Object detection with probabilistic modeling

To introduce our approach, we first focus on the localization task and then extend it to the classification task. As we will show later in our experiments, our method is applicable to both single-stage and two-stage object detectors.

Localization: In object detection, a bounding box $b$ is defined by its coordinates for the center $(x, y)$, its width $(w)$, and its height $(h)$. In our work, instead of predicting a deterministic value, our mixture model predicts $3$ groups of parameters for each bounding box: the mean $(\mu_x, \mu_y, \mu_w, \mu_h)$, the variance $(\Sigma_x, \Sigma_y, \Sigma_w, \Sigma_h)$, and the weights of the mixture $(\pi_x, \pi_y, \pi_w, \pi_h)$.

Let $(\tilde{\pi}_b, \tilde{\mu}_b, \tilde{\Sigma}_b)^K_{k=1}$, $b \in \{x, y, w, h\}$ be the bounding box outputs obtained using our network. The parameters of a GMM with $K$ models for each coordinate of the bounding box are obtained as follows:

$$
\pi^k_b = \frac{e^{\tilde{\pi}^k_b}}{K \sum_{j=1}^{K} e^{\tilde{\pi}^k_j}}, \quad \mu^k_b = \tilde{\mu}^k_b, \quad \Sigma^k_b = \sigma(\tilde{\Sigma}^k_b),
$$

where $\pi$ is the mixture weight for each component, $\mu$ is the predicted value for each bounding box coordinate, and $\Sigma$ is the variance for each coordinate representing its aleatoric uncertainty. As suggested in [8], we use a softmax function to keep $\pi$ in probability space and use a sigmoid function to satisfy the positiveness constraint of the variance, $\Sigma_b > 0$.

Localization loss: The conventional bounding box regression loss, the smooth L1 loss [14], only considers the coordinates of the predicted bounding box and ground-truth (GT) box. Therefore, it cannot take into account the ambiguity (aleatoric uncertainty) of the bounding box. For training the mixture density network for localization, we propose a localization loss based on the negative log-likelihood loss.

Our loss regresses the parameters of the GMM to the offsets of the center $(x, y)$, width $(w)$, and height $(h)$ of the anchor (default) box $(d)$ for positive matches:

$$
L_{loc}(\lambda, l, g) = - \sum_{i \in P_{pos}} \sum_{b} \lambda_{ij}^G \log( \sum_{k=1}^{K} \pi^k_b N(\tilde{\mu}^k_b, \tilde{\Sigma}^k_b) + \epsilon),
$$

where $\lambda^G_i = \left\{ \begin{array}{ll} 1, & \text{if } IoU > 0.5; \\ 0, & \text{otherwise}. \end{array} \right.$, $\tilde{g}_{ix} = \frac{(g_{ix} - d_{ix})}{d_{ix}}$, $\tilde{g}_{iy} = \frac{(g_{iy} - d_{iy})}{d_{iy}}$, $\tilde{g}_{iw} = \log(g_{iw}^j / d_w^j)$, $\tilde{g}_{ih} = \log(g_{ih}^j / d_h^j)$.

Classification: We now focus on the classification head of the object detector. We model the output of every class as a GMM (see Fig. 2a). Our approach estimates the mean $\mu^k_p$ and variance $\Sigma^k_p$ for each class, and the weights of the mixture $\pi^k$ for each component of the GMM. We process the parameters of the GMM following Eq. 2, and obtain the class probability distribution for the $k$-th mixture by using the reparameterization trick [25] of applying Gaussian noise and variance $\Sigma^k_p$ to $\mu^k_p$ [24]:

$$
\hat{c}^k_p = \mu^k_p + \sqrt{\Sigma^p_p} \gamma, \quad \gamma \sim \mathcal{N}(0, 1),
$$

where $\gamma$ is the auxiliary noise variable and has the same size as $\mu^k_p$ and $\Sigma^k_p$.
Classification loss: For training the mixture density network for classification, we propose a loss function that takes into account the IoU of the anchor box compared to the GT box and considers the hard negative mining. More precisely, we formulate the classification loss as a combination of two terms $L_{cl}^{Pos}$ and $L_{cl}^{Neg}$ representing the contribution of positive and negative matches:

$$
L_{cl}^{Pos}(\lambda, c) = -\sum_{i\in Pos} N \sum_{k=1}^{K} \prod_{p=0}^{C} \pi^{ik} (\hat{c}_{ij}^{p} - \log \sum_{p=0}^{C} e^{\hat{c}_{ij}^{p}}),
$$

$$
L_{cl}^{Neg}(c) = -\sum_{i\in Neg} M \sum_{k=1}^{K} \prod_{p=0}^{C} \pi^{ik} (\hat{c}_{ij}^{0} - \log \sum_{p=0}^{C} e^{\hat{c}_{ij}^{p}}),
$$

where $N$ is the number of positive matches, $K$ is the number of mixtures, $C$ is the number of classes, with 0 representing the background class $\hat{c}_{p0}^1$, $\hat{c}_{ij}^{p} = \pi_{ij}^{p}G_{ij}$ is the GT class for the $j$-th GT box, $\hat{c}_{ij}^{p}$ is the result calculated by Eq. 4, $\lambda^{ij}$ is the same as used in Eq. 3, and $M$ is the hard negative mining ratio. Instead of using all the negative matches, we sort them using the proposed mixture classification loss and choose top $M \times N$ as final negative matches for training. In experiments, we set $M$ to 3 as suggested in [31].

Final loss: We define the overall loss to train the object detector using mixture density network as:

$$
L = \begin{cases} 
\frac{1}{N} \left( L_{loc}(\lambda, l, b) / \eta + L_{cl}^{Pos}(\lambda, c) + L_{cl}^{Neg}(c) \right), & \text{if } N > 0, \\
0, & \text{otherwise},
\end{cases}
$$

where $N$ is the number of positive matches. In experiments, we set $\eta$ to 2 as suggested in [6].

At inference, we can compute the coordinates of the bounding box $R_b$ and the confidence score for each class $P_l$ by summing the components of the mixture model as follows:

$$
\text{Localization: } R_b = \sum_{k=1}^{K} \pi_{ik} \hat{c}_{ik}, \quad \text{Classification: } P_l = \sum_{k=1}^{K} \pi_{ik} \hat{c}_{ik} / \sum_{j=0}^{C} e^{\hat{c}_{ij}^{k}}.
$$

3.2. Improving parameter efficiency

In order to predict a probability distribution of the output values, our approach involves modifying the last layer of the network and therefore incurs an increase in the number of parameters, especially in the classification head. More precisely, for an output feature map of size $F \times F$, with $C$ classes, $D$ anchor boxes, and each bounding box defined using 4 coordinates, the number of parameters in the new layer added to estimate a $K$-component GMM with 3 parameters is $F \times F \times D \times (4 \times 3 \times K)$ for the localization and $F \times F \times D \times (C \times 2 \times K + K)$ for classification. We see that the number of parameters in the classification head is proportional to the number of classes.

In this section, we focus on improving the efficiency of the algorithm by reducing the number of parameters in the classification head. To this end, as shown in Fig. 2b, we relax the problem of estimating the variance $\Sigma_p$, in order to reduce the number of parameters with $F \times F \times D \times (C \times K + K)$. Instead, we obtain class probabilities as $\pi_{ik} = \text{Softmax}(\mu_{ik}^p)$, and use them to estimate the aleatoric uncertainty as follows:

$$
u_{al} = \sum_{k=1}^{K} \pi^{ik} (\text{diag}(\mu_{ik}^p) - (\hat{c}_{ik}^{p})^2),
$$

where $\text{diag}(q)$ is a diagonal matrix with the elements of the vector $q$ and $q^\otimes^2 = qq^T$. In this case, $u_{al}$ is $C \times C$ matrix where the value of each diagonal element can be interpret as a class-specific aleatoric uncertainty [27].

Finally, we modify the classification loss function by reducing the number of parameters in the network and therefore incurs an increase in the number of parameters as follows:

$$
L_{cl}^{Pos}(\lambda, c) = -\sum_{i\in Pos} N \sum_{k=1}^{K} \prod_{p=0}^{C} \pi^{ik} (\hat{c}_{ij}^{p} - \log \sum_{p=0}^{C} e^{\hat{c}_{ij}^{p}}),
$$

$$
L_{cl}^{Neg}(c) = -\sum_{i\in Neg} M \sum_{k=1}^{K} \prod_{p=0}^{C} \pi^{ik} (\hat{c}_{ij}^{0} - \log \sum_{p=0}^{C} e^{\hat{c}_{ij}^{p}}),
$$

where all parameters are same as Eq. 5, except for class probability $\hat{c}_{ij}^{p}$.

3.3. Scoring function

The scoring function in active learning provides a single value per image indicating its informativeness. Our scoring function estimates the informativeness of an image by aggregating all the aleatoric and epistemic uncertainty values for each detected object in the image.

Specifically, let $U = \{u_{ij}\}$ be the set of uncertainty values (aleatoric or epistemic) of a group of images where $u_{ij}$ the uncertainty for the $j$-th object in the $i$-th image. For localization, $u_{ij}$ is the maximum value over the 4 bounding box outputs. We first normalize these values using z-score normalization ($\tilde{u}_{ij} = (u_{ij} - \mu_U) / \sigma_U$) to compensate for the fact that the values for the coordinates of the bounding box are unbounded and each uncertainty of an image might have a different range of values. We then assign to each image the maximum uncertainty over the detected objects $u^I = \max_i \tilde{u}_{ij}$. We empirically find that taking the maximum over the coordinates and the objects performs better than by taking the average.

Using the algorithm described above, we obtain four different normalized uncertainty values for each image: epistemic and aleatoric for classification and localization, $u = \{u_{ep_c}, u_{al_c}, u_{ep_l}, u_{al_l}\}$, respectively. The remaining part is to aggregate these scores into a single one. We explore different combinations of scoring functions that aggregate these uncertainties, including sum or taking the maximum,
like other active learning studies [16, 33]. As we will show in our experiments, taking the maximum over them achieves the highest results.

### 4. Experiments

In this section, we demonstrate the benefits of our approach. We first study the impact of using probabilistic modeling for the object detector and then analyze the proposed scoring function and relevant SOTA approaches in the context of active learning.

**Datasets:** We use PASCAL VOC [9] and MS-COCO [30] datasets. For PASCAL VOC, that contains 20 object categories, we use VOC07 (VOC2007) trainval and VOC07+12 trainval (union of VOC2007 and VOC2012) for training and evaluate our results on VOC07 test. For MS-COCO, that contains 80 object categories, we use MS-COCO train2014 for training and evaluate our results on val2017.

**Experimental settings:** We employ Single Shot MultiBox Detector (SSD) [31], which is widely used in active learning studies [33, 40], with a VGG-16 backbone [37]. We train our models for 120,000 iterations using SGD with a batch size of 32 and a maximum learning rate of 0.001. We use a learning rate warm-up strategy for the first 1,000 iterations and divide the learning rate by 10 after 80,000 and 100,000 iterations. We set the number of Gaussian mixtures to 4, and in the supplementary materials, we provide an ablation study on the number of mixtures. Unless specified otherwise, we report the performance using the average and standard deviation of mAP for three independent trials.

### 4.1. Object detection with probabilistic modeling

We first analyze the impact of using our proposed probabilistic modeling for object detection on PASCAL VOC and MS-COCO. For MS-COCO, we use a random subset of 5,000 training images from train2014. We compare the accuracy of our GMM Our\textsubscript{gmm} and the model with improved parameter efficiency Our\textsubscript{eff} to the SSD [31] and several network configurations either using single or multiple Gaussians for the classification or localization heads.

### Table 1: mAP (in %) of different instances of our approach compared to the original SSD network. SGM and MDN refer to single and multiple Gaussian models, and we apply those to localization (Loc), classification (Cl), and their combination (Loc+Cl).

<table>
<thead>
<tr>
<th>Method</th>
<th>Head</th>
<th>(a) VOC07</th>
<th>(b) MS-COCO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IoU&gt;0.5</td>
<td>IoU&gt;0.75</td>
<td>IoU&gt;0.5</td>
</tr>
<tr>
<td>SSD</td>
<td>69.29±0.51</td>
<td>43.36±1.24</td>
<td>25.63±0.40</td>
</tr>
<tr>
<td>SGM</td>
<td>Loc</td>
<td>70.20±0.27</td>
<td>45.39±0.23</td>
</tr>
<tr>
<td>MDN</td>
<td>Loc</td>
<td>70.09±0.22</td>
<td>46.01±0.27</td>
</tr>
<tr>
<td>SGM</td>
<td>Cl</td>
<td>69.95±0.41</td>
<td>44.25±0.26</td>
</tr>
<tr>
<td>MDN</td>
<td>Cl</td>
<td>70.47±0.17</td>
<td>44.47±0.06</td>
</tr>
<tr>
<td>Ours\textsubscript{gmm}</td>
<td>Loc+Cl</td>
<td>70.19±0.36</td>
<td>46.11±0.38</td>
</tr>
<tr>
<td>Ours\textsubscript{eff}</td>
<td>Loc+Cl</td>
<td>70.45±0.06</td>
<td>46.18±0.26</td>
</tr>
</tbody>
</table>

Figure 3: Examples of aleatoric and epistemic uncertainties for inaccurate detections, see more examples in the supplementary material. Starting from top-left image and going in clockwise direction: Person is a false positive; Person bounding box is not correct; A sheep is misclassified as a bird; A sheep is misclassified as a cow.

In Tab. 1a and Tab. 1b, we summarize the results of this experiment performed on VOC07 and MS-COCO, respectively. As shown, all networks that include probabilistic modeling outperform the SSD on both datasets. This is because of the regularization effect of the proposed loss function, which has a loss attenuation due to aleatoric uncertainty [6]. As a result, we obtain models that are robust to noisy data. Considering both the normal (IoU>0.5) and the strict metric (IoU>0.75), Ours\textsubscript{gmm} and Ours\textsubscript{eff} outperform all other variations on VOC07. On MS-COCO, Ours\textsubscript{gmm} outperforms all other instances and the baseline, while Ours\textsubscript{eff} reaches competitive results. We expect that the amount of noisy data in MS-COCO is larger than that of PASCAL VOC because MS-COCO has more diverse data. As shown in Eq. 9, there is no aleatoric uncertainty in Ours\textsubscript{eff}’s classification loss and therefore, we argue that the regularization by aleatoric uncertainty has a greater effect in MS-COCO.

In Fig. 3, we present representative examples of uncertainty scores for several images where the detector fails to detect the object. As shown, each uncertainty value (bold numbers in Fig. 3) provides a different insight into some particular failure. Localization uncertainties are related to the accuracy of the bounding box prediction, whereas classification uncertainties are related to the accuracy of the category prediction. Interestingly, in these examples, even if the predictions are wrong, uncertainty values seem to be uncorrelated suggesting each uncertainty could predict inaccurate results independently. From these results, we can...
conclude that the proposed approach not only computes uncertainty in a single forward pass of a single model but also boosts the performance of the detection network. As shown in the next experiment, combining these values will improve the data selection process during active learning.

### 4.2. Active learning evaluation

We now focus on evaluating the performance of our active learning on PASCAL VOC and MS-COCO datasets. We use an initial set of 2,000 for VOC07, 1,000 for VOC07+12 as suggested by [40], and 5,000 samples in MS-COCO as suggested by [23]. Then, during the active learning stage, for each unlabeled image, we apply non-maximum suppression and we compute the uncertainties for each of the surviving objects. The scoring function aggregates these uncertainties using the maximum or sum to provide the final informativeness score for the image. We score the set of unlabeled images and select the 1,000 images [40] with the highest score. Then, we add them to the labeled training set and repeat this process for several active learning cycles. For every active learning iteration, we train the model from scratch, using ImageNet pretrained weight.

**Scoring aggregation function:** We compare the active learning performance obtained using different functions to aggregate the aleatoric and epistemic uncertainties of both classification and localization heads. In particular, we compare seven different instances of our approach with random sampling: 1) Only the aleatoric or epistemic uncertainty on each task; 2) The sum of aleatoric and epistemic uncertainty on the localization or classification head; 3) The sum of aleatoric or epistemic uncertainty on the localization and classification; 4) The sum of aleatoric and epistemic uncertainties for both localization and classification; 5) The maximum of aleatoric and epistemic uncertainty on the localization or classification head; 6) The maximum of aleatoric or epistemic uncertainty on the localization and classification, and 7) The maximum value of these four uncertainties. The results for this comparison are shown in Tab. 2. Our approach using the maximum value of aleatoric and epistemic uncertainties of both localization and classification tasks consistently outperforms all the other aggregation functions on each active learning iterations. More concretely, the maximum value of all uncertainties for both tasks shows better data selection performance in active learning than others. Based on these results, we use the maximum value of all uncertainties as a scoring function during active learning to compare with other active learning studies.

In Tab. 3, we summarize the overlap in the selection as a function of the uncertainty measure. The overlapping ratio using both uncertainties is 48% for localization and 33% for classification. More importantly, when we consider both uncertainties on localization and classification together, the overlapping ratio decreases to barely 14%. This suggests that uncertainty measures obtained for localization and classification are diversified and their combination improves the image selection process.

**Comparison to SOTA on VOC07:** In Tab. 4, we summarize the active learning results and computing cost of our method compared to relevant active learning approaches in the literature. In order to compare the computing cost, we provide the number of parameters and the forward time of each method. In general, a fast forward (backward) step and a small model size leads to a lower training cost and data sampling time during active learning [19, 22].

To focus on the active learning, we reproduce all numbers by applying each sampling method to the proposed GMM architecture where the output is a mixture distribu-

<table>
<thead>
<tr>
<th>Aggregation function</th>
<th>mAP in % (# images)</th>
</tr>
</thead>
<tbody>
<tr>
<td>random sampling</td>
<td></td>
</tr>
<tr>
<td>$u_{al}$</td>
<td>62.43±0.10 66.36±0.13 68.47±0.09</td>
</tr>
<tr>
<td>$u_{ep}$</td>
<td>62.43±0.10 67.08±0.18 68.84±0.18</td>
</tr>
<tr>
<td>$u_{al}$</td>
<td>62.43±0.10 68.75±0.26 69.01±0.17</td>
</tr>
<tr>
<td>$u_{ep}$</td>
<td>62.43±0.10 67.09±0.09 68.75±0.08</td>
</tr>
<tr>
<td>$\sum_{j}(u_{al}(\cdot,p_j))$</td>
<td>62.43±0.10 67.01±0.10 68.58±0.29</td>
</tr>
<tr>
<td>$\sum_{j}(u_{ep}(\cdot,p_j))$</td>
<td>62.43±0.10 67.07±0.27 69.03±0.20</td>
</tr>
<tr>
<td>$\sum_{j}(u_{al}(\cdot,p_j)+u_{ep}(\cdot,p_j))$</td>
<td>62.43±0.10 66.96±0.08 68.92±0.23</td>
</tr>
<tr>
<td>$\sum_{j}(u_{al}(\cdot,p_j)\cdot u_{ep}(\cdot,p_j))$</td>
<td>62.43±0.10 66.49±0.14 68.62±0.24</td>
</tr>
</tbody>
</table>

Table 2: VOC07: Comparison of scoring aggregation functions for active learning based on the aleatoric uncertainty, epistemic uncertainty, and their combination of each task.

<table>
<thead>
<tr>
<th>Localization</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aleatoric</td>
<td>Epistemic</td>
</tr>
<tr>
<td>$u_{al}$</td>
<td>$u_{ep}$</td>
</tr>
<tr>
<td>1st (2k)</td>
<td>1st (2k)</td>
</tr>
<tr>
<td>62.43±0.10</td>
<td>52.35</td>
</tr>
<tr>
<td>66.72±0.21</td>
<td>52.35</td>
</tr>
<tr>
<td>68.47±0.09</td>
<td>52.35</td>
</tr>
<tr>
<td>2nd (3k)</td>
<td>62.43±0.10</td>
</tr>
<tr>
<td>67.02±0.10</td>
<td>52.35</td>
</tr>
<tr>
<td>68.84±0.18</td>
<td>52.35</td>
</tr>
<tr>
<td>3rd (4k)</td>
<td>62.43±0.10</td>
</tr>
<tr>
<td>67.07±0.27</td>
<td>52.35</td>
</tr>
<tr>
<td>69.01±0.17</td>
<td>52.35</td>
</tr>
</tbody>
</table>

Table 3: Overlapping ratio (in %) of selected images as a function of the type of uncertainty used.

<table>
<thead>
<tr>
<th>Method</th>
<th>mAP in % (# images)</th>
<th>Number of para. ($\times 10^6$)</th>
<th>Forward time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random [33]</td>
<td>52.35</td>
<td>8.011</td>
<td></td>
</tr>
<tr>
<td>Entropy [33]</td>
<td>62.43±0.10</td>
<td>66.85±0.12</td>
<td>68.70±0.18</td>
</tr>
<tr>
<td>Core-set [34]</td>
<td>62.43±0.10</td>
<td>66.57±0.20</td>
<td>68.57±0.26</td>
</tr>
<tr>
<td>LLAL [10]</td>
<td>62.47±0.16</td>
<td>67.02±0.11</td>
<td>68.90±0.15</td>
</tr>
<tr>
<td>MC-dropout [11]</td>
<td>62.43±0.19</td>
<td>67.10±0.07</td>
<td>69.30±0.09</td>
</tr>
<tr>
<td>Ensemble [16]</td>
<td>62.43±0.10</td>
<td>67.11±0.26</td>
<td>69.26±0.14</td>
</tr>
<tr>
<td>Ours$_{max}$</td>
<td>62.43±0.10</td>
<td>67.32±0.12</td>
<td>69.43±0.11</td>
</tr>
<tr>
<td>Ours$_{max}$</td>
<td>62.43±0.10</td>
<td>67.32±0.12</td>
<td>69.43±0.11</td>
</tr>
</tbody>
</table>

Table 4: VOC07: Comparison of mAP and computing cost of active learning with most relevant approaches. Para. and sec refer to parameters and seconds, respectively.
VOC07+12: a) Comparison to published works using a single model for scoring. Numbers are taken from [40]; b) Comparison to multiple model-based methods, ensemble and MC-dropout. Details of the numbers to reproduce the plot are in the supplementary material.

Figure 4: VOC07+12: a) Comparison to published works using a single model for scoring. Numbers are taken from [40]; b) Comparison to multiple model-based methods, ensemble and MC-dropout. Details of the numbers to reproduce the plot are in the supplementary material.

Figure 5: VOC07+12: Computing cost comparison to baseline and methods using multiple models; Model parameters in millions (M) and the forward time in seconds (sec). See the supplementary material for numerical details.

As shown in Tab. 4, both instances of our approach consistently outperform all the other single-model based methods [31, 33, 34, 40] in every active learning iteration. Compared to multi-model based methods [11, 16], our proposed methods show higher accuracy while requiring a significantly lower computing cost. These results demonstrate that despite having a lower computational cost, our proposed method improves the active learning sampling performance compared to previous works.

Comparison to SOTA on VOC07+12: We now compare our approach to existing single-model based approaches on VOC07+12. Here, we consider the SOTA results reported in [40] including LLAL [40] and core-set [34], in addition to entropy [33] and random sampling. We use the same open-source and setting used in [40] for a fair comparison. To solely focus on active learning, we compare the performance based on the same baseline with [40], i.e., SSD. To do this, we train the SSD using a dataset sampled by our proposed scoring function in the proposed GMM architecture and the architecture with improved parameter efficiency. As shown in Fig. 4a, our method outperforms all the other single-model based methods. In the last active learning iteration, our approach achieves 0.7598 mAP which is 2.6 percent points higher than the score achieved by LLAL [40] (0.7338 mAP), thus showing a high-performance improvement in active learning based on a single model.

Finally, we compare our approach with methods using multiple models, i.e., ensembles [16] and MC-dropout [11]. For ensembles and MC-dropout, we follow the same design mentioned in Tab. 4 and apply it to the SSD. In Fig. 4b and Fig. 5, we present the accuracy and computational cost comparison of these methods. As shown in Fig. 4b, in terms of the accuracy, our approach performs on par with MC-dropout and ensembles. However, our method uses a single forward pass of a single model to estimate the uncertainties, which is more efficient than ensembles and MC-dropout based methods. With respect to the number of parameters, MC-dropout has the same number of parameters as SSD since dropout layers do not add any new parameters, but it requires multiple forward passes. Our approach adds extra
Here, in a first experiment, we show how our method can be applied to a two-stage detector such as Faster-RCNN [32] with FPN [29]. For this experiment, we use the same PASCAL VOC dataset as in Tab. 1a. In Tab. 6, we show the summary of the accuracy and the computing cost of our mixture models based on Faster-RCNN and the original Faster-RCNN as a baseline. As shown, both versions of our approach outperform the original model with up to 1.13 mAP improvement. Importantly, in this case, our approach is applied to the output layer of the detection network after region proposal in Faster-RCNN, therefore there is a negligible increase in computing cost and latency because the computation does not include the number of anchor boxes.

Finally, we study the transferability of actively acquired datasets. We compare the performance of SSD using different backbones such as Resnet-34 and Resnet-50 [17], and Faster-RCNN [32] detector trained using our actively sampled dataset. We perform the experiments in the actively sampled dataset from the last active learning cycle in Tab. 4. For completeness, we also report the accuracy obtained using random sampling. We summarize the results of this experiment in Tab. 7. As shown, networks trained using the actively sampled dataset outperform those trained using random sampling with up to 2.52 mAP improvement. Conclusively, our method not only scales to other object detection networks but also datasets actively acquired using our approach can be used to train other architectures.

5. Conclusions

We have proposed a novel deep active-learning approach for object detection. Our approach relies on mixture density networks to estimate, in a single forward pass of a single model, two types of uncertainty for both localization and classification tasks, and leverages them in the scoring function. Our proposed probabilistic modeling and scoring function achieve outstanding performance gains in accuracy and computing cost. We present a wide range of experiments on two publicly available datasets, PASCAL VOC and MS-COCO. Besides, our results suggest that our approach scales to new models with different architectures.
References


