Group 9: Active Learning for Deep Object Detection

Nishant Kiran Valvi Abhinav Kumar
21111407, 16907018.
{nkiranv21, abhikmr}@iitk.ac.in
Indian Institute of Technology Kanpur (IIT Kanpur)

Abstract

The goal of active learning is to reduce labeling costs by selecting only the most informative samples on a dataset. Most of the existing methods use multiple models or are extensions of classification methods. Therefore, they only utilize image's informativeness using the classification head. The approaches in [1] use mixture density networks to estimate probabilistic distribution for localization head and classification head. The predictive uncertainty is decomposed into aleatoric (inherent noise in the data) and epistemic (lack of knowledge of the model; inversly proportional to the density of training data) uncertainty. These uncertainties help the model recognize suspicious predictions (aleatoric uncertainty) and recognize samples that do not resemble the training set (epistemic uncertainty). The scoring function uses aleatoric and epistemic uncertainty obtained by a single forward pass of a single model to obtain every image's informativeness score. We propose a method to calculate these uncertainties and thereafter the scoring function to measure informativeness of an image. We show that deep active learning can reach comparable performance of the corresponding model trained without active learning with much less data.

1 Introduction

We aim to detect an object in an image and draw bounding boxes around the object in object detection. Recent deep object detection models such as YOLO [2] and SSD [3] have dramatically improved object detection results. These models are trained on annotated images with bounding boxes ground truth. Creating such annotated images require a lot of time and effort. To train the deep learning models, we require a large amount of labelled data. Besides, not all the training images are equally useful since 'similar' images may not contribute much to learning. The solution to these problems is using active learning methods i.e., choose the most informative training samples to get the required performance from the model with fewer training examples.

We propose a new method to calculate the aleatoric and epistemic uncertainties which utilizes the bounding boxes given by the YOLO model. For a given class of objects, we select clusters of bounding boxes formed around each object of that class. We utilize these clusters to calculate the terms required for the scoring functions. We further show the performance of our approach with baseline YOLOv3 model and a basic scoring function.

The structure of the rest of the report is as follows:

- Related Work
- Proposed Idea
- Methodology
- Results
- Discussion and Future Work
- Conclusion
- Individual Contributions

2 Related Work

Active learning in computer vision has been explored extensively recently. The survey in [4] talks about approaches such as membership query synthesis, stream-based selective sampling and pool-based selective sampling. These query strategies then include methods like Uncertainty-based query strategy wherein the samples are ranked based on their uncertainty and then queried; Deep Bayesian active learning where active learning is applied with the aid of Bayesian convolutional neural networks (CNN); Density based methods where they find the subset of the dataset which represents the whole dataset.

The authors of [1] propose a single model that uses a single forward pass and utilizes aleatoric and epistemic uncertainties which will be explained in the later sections. They model the localization and classification heads as the output of a mixture density network that learns a Gaussian mixture model. The output of these heads are then used to calculate a scoring function using which the samples are ranked and queried. In [5] they use Bayesian CNNs to have a Bayesian control over every step of the neural network so that anchor-level and object-level priors can be used. Moreover, instead of using non-maximum suppression (NMS) they use fully Bayesian inference so that they can make use of all the predicted information for the bounding box and category both. In [6] they utilize acquisition functions that rely on model uncertainty, but deep learning methods rarely represent model uncertainty so they use Bayesian CNNs instead. The authors in [7] consider different metrics so as to measure the informativeness of an object hypothesis.

3 Proposed Idea

Training deep learning models requires a large amount of data; acquiring labeled data for object detection is a costly and time-consuming process. Active learning aims to reduce labeling costs by selecting only the most informative samples on a dataset. The goal is to select the most informative images from the dataset that provide more value, rather than selecting images randomly for training and reducing labeling costs. This project has applied two scoring functions to select the most informative sample image for the training object detection model using YOLOv3 [2].

3.1 Score-B (basic scoring function)

The first method for selecting images is based on object class probability p_i and bounding box confidence c_i for ith object in an image. We calculate the score for every object in the images by multiplying class probability and bounding box confidence for each object and taking the average for every object in the image. It is given as:

$$S_i^B = \frac{1}{N_i} \sum_{j=1}^{N_i} p_j c_j \tag{1}$$

where S_i^B represents Score-B for ith image. N_i is the number of objects in ith image. We calculate this score for all the images $i \in \mathcal{D}_U$ where $\mathcal{D}_U$ is the unlabelled pool of data. We then select the top K images with lowest score and move them to the training dataset $\mathcal{D}_T$.

3.2 Score-U (uncertainty scoring function)

Aleatoric uncertainty: It is the inherent noise in the data, such as sensor noise and can be attributed to occlusions or lack of visual features.
Epistemic uncertainty: It refers to the uncertainty caused by the lack of knowledge of the model and is inversely proportional to the density of training data.

In this method, we calculate the score of the image using aleatoric u_{al} and epistemic u_{ep} uncertainties for the bounding box of the object in the image. Let $\mathcal{C}$ be the set of all image classes. The YOLOv3 algorithm outputs all the bounding boxes before non-maximum suppression is applied. Each bounding box b has the following tuple of coordinates (x_1, y_1, x_2, y_2). For every class $l \in \mathcal{C}$, we will have a cluster of bounding boxes around every detected object. Let the number of clusters be M_l, then for every cluster $m \in (1, M_l)$, with N_m bounding boxes, we will have a bounding box with maximum bounding box confidence, let it be b_m^{max}. We then remove all the bounding boxes b_{mj} $(j \in (1, N_m))$

with their intersection-over-union (IoU) greater than 0.5. Let the set of remaining bounding boxes of class l and cluster m after IoU process be $\mathcal{B}_m^l$. Now, for every cluster m we calculate the mean $\boldsymbol{\mu}_m^l = (\mu_{x_1}, \mu_{y_1}, \mu_{x_2}, \mu_{y_2})$ and variance $\boldsymbol{\Sigma}_m^l = (\Sigma_{x_1}, \Sigma_{y_1}, \Sigma_{x_2}, \Sigma_{y_2})$ of all the remaining bounding boxes in that cluster.

With this formulation, we can calculate the uncertainties for ith image as follows:

$$u_{al}^i = \sum_{l \in \mathcal{C}} \Big\{ \sum_{m=1}^{M_l} \sum_{b \in \mathcal{B}_m^l} \pi_b \boldsymbol{\Sigma}_b \Big\} \tag{2}$$

$$u_{ep}^i = \sum_{l \in \mathcal{C}} \Big\{ \sum_{m=1}^{M_l} \sum_{b \in \mathcal{B}_m^l} \pi_b \| \boldsymbol{\mu}_b - \sum_{f \in \mathcal{B}_m^l} \pi_f \boldsymbol{\mu}_f \|^2 \Big\} \tag{3}$$

where π_b is the bth bounding box confidence. We calculate aleatoric and epistemic uncertainties, add them up, and then select K images with top K scores from the unlabelled pool of data.

4 Methodology

4.1 Dataset:

We have used the COCO dataset [8] for training the model, which has objects from 80 different classes with various sizes in the image. We have divided the dataset into testing and unlabeled pool. This unlabeled pool of images is selected for labeling and then moved to the training set.

4.2 Algorithm:

We have used the YOLOv3 model, a real-time object detection algorithm that identifies objects in video and images. To detect an object, YOLO uses features learned by DarkNet architecture [2]. YOLO takes images in batches of size n with input shape $(n, 416, 416, 3)$ where n is the batch size, 416×416 is the resolution of input image and 3 is the channel size. YOLO gives output in three different scales $(n, 13, 13, 255)$ for a large object, $(n, 26, 26, 255)$ for a medium object, and $(n, 52, 52, 255)$ for a small object. For an image i, each grid return 255 numbers corresponding to 3 anchor boxes, each of which corresponds to 80 classes and a tuple of size 5 i.e., $(b_x, b_y, b_h, b_w, c_i)$ where (b_x, b_y) is the center of the bounding box and (b_w, b_h) are the width and height and c_i is the bounding box confidence. Hence the number $255 = (3 * (80 + 5))$. Finally, NMS is used to bring bounding boxes to one bounding box for every detected object.

YOLOv3 uses Darknet-53, a 53 layer network trained on Imagenet as the base. For detection, 53 more layers are added on the Darknet giving 106 layers of CNN architecture for YOLOv3. It uses binary cross-entropy loss for the class predictions and the sum of squared error loss for bounding box prediction. We have used score threshold as 0.6 and IOU threshold as 0.5. We have used the Adam optimizer with a learning rate of 1e-4.

Now, we explain our proposed active learning idea. We use YOLOv3 as our deep object detector. On top of it we perform active learning by utilizing the scoring functions Score-B and Score-U. We start by training the YOLOv3 algorithm on initial data of size s_0. We then start the active learning loop wherein, at each iteration we select K images by using Score-B/Score-U and add them to the training data and then retrain the model. In this way, the model is able to learn from better samples and hence it achieves good performance with less data.

4.3 Implementation details

4.3.1 Active Learning Loop

Algorithm 1 Active Learning Loop

Require: Label images L, Unlabelled images U, initial model Y, Scoring function S

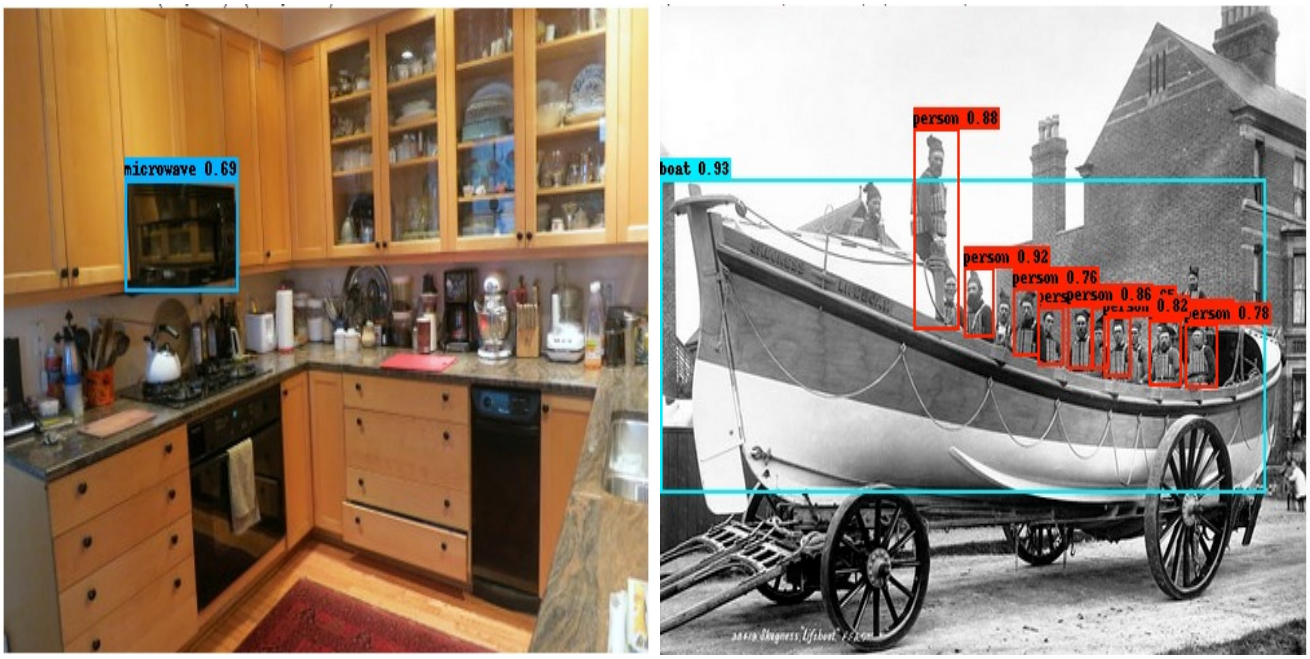
while In budget Or got required performance **do**
  for image in U **do**
    Calculate score by S
  end for
  $L \leftarrow K$ most informative images according to S
  Train Y with L
  evaluate model Y
end while

For the training model with active learning, we have an initial trained model on a small number s_0 of training images. Given the trained model, we calculate the scores of every image using the scoring functions defined in above sections and select the most informative sample according to the scoring function. These selected images are added to the training set to train the model in the next iteration. This active learning loop is run until we run out of images labeling budget or get satisfactory performance from the model.

4.3.2 Image Annotation

All the unlabeled images are ranked based on the score generated by the scoring function. In Fig. 1, the left side image with a score of 0.66 is more informative as the model cannot detect with confidence, and on the right side image with a score of 0.99 model can detect objects with a large confidence score. This image is not informative for further training. Based on the score, top N the most informative images are selected for labeling, and the model has trained again by adding newly selected images to the training set.

Figure 1: Image left: score = 0.66, more informative; Image right: score = 0.99, less informative



(a) (b)

Figure 2: a) Low score, more informative; b) High score, less informative

5 Results

Model /Scores	Pre-trained YOLOv3	Baseline YOLOv3 (10k data points)	Score-B (3k data points)	Score-U (3k data points)
mAP	30.35 %	23.62 %	18.98 %	16.62 %
mAP 50	48.61 %	38.73 %	32.73 %	29.98 %
mAP 75	33.54 %	25.73 %	20.22 %	16.73 %
mAP Large	47.34 %	36.53 %	30.08 %	25.94 %
mAP Medium	27.67 %	20.36 %	15.12 %	14.69 %
mAP Small	6.71 %	4.9 %	3.10 %	3.7 %

Table 1: mAP scores

We train a baseline YOLOv3 model with $10,000$ images and then $3,000$ images for active learning models using scoring functions. In Table 1 we have the mAP scores of different models. mAP 50 and mAP 75 represent mAP scores with IoU thresholds of 0.5 and 0.75. mAP large, medium, and small are mAP scores of the objects of different scales.

We observe that even though we train the active learning models with $3,000$ data points, they are able to reach comparable performance as compared with pre-trained YOLOv3 and baseline YOLOv3.

6 Discussion and Future Work

We presented two scoring functions for the training model with active learning. We improved the model's performance by training it iteratively by adding informative images in training set at every iteration. The model is reset and trained again with an updated training set at every iteration. If the model is run till convergence, training time is large for every iteration. Training time can be reduced by training the previous model with updated training images. As per the mAP score, the models are not able to detect small objects properly scoring function can be updated to select images considering

object size in the images.

7 Conclusion

This project has proposed active learning approaches to train object detection using a fraction of training images. We also proposed two different scoring functions to rank images as per their informativeness for the model. The model can be trained in less training images by selecting informative images. The active learning approach can be useful in scenarios where obtaining annotations for images is a costly affair for example, in the medical field.

8 Individual Contributions

Contribution/Name	Literature Survey	Code	Presentations	Report	Total
Abhinav	50%	50%	50%	50%	50%
Nishant	50%	50%	50%	50%	50%

References

[1] J. Choi, I. Elezi, H.-J. Lee, C. Farabet, and J. M. Alvarez, "Active learning for deep object detection via probabilistic modeling," 2021.

[2] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018.

[3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Computer Vision – ECCV 2016*, pp. 21–37, Springer International Publishing, 2016.

[4] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, "A survey of deep active learning," 2020.

[5] A. Harakeh, M. Smart, and S. L. Waslander, "Bayesod: A bayesian approach for uncertainty estimation in deep object detectors," 2019.

[6] Y. Gal, R. Islam, and Z. Ghahramani, "Deep bayesian active learning with image data," 2017.

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

[8] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2014.