Object Detection in Retail

Aniket Kumar (T23189) Akash Pal (T23197) Abhishek Singh Rawat (T23191)

***Abstract*—In retails, most of the time we find scenarios consist of densely packed objects lying on the shelf. It is enormous task to keeping track of the objects on the shelf. Similar kind of scenarios tends to happen in a crowd, traffic etc. Our task is to come up with an proposition to detect objects in such scenes. In this project, we have implemented two novels (1) the Soft-IoU score as an improvement of detections and (2) EM merger unit with ResNet50 as a backbone. We used SKU110K dataset to test this model which is highly densed, thus making object detection a challenging task. Result shows that average precision lies around 50% , which gives very much room for improvement in the model. We also test this dataset on YOLOv2 with modified parameters which results faster detection but low average precision.**

I. Introduction

We have reviewed some recent deep-learning papers [1], [2], [3], [4], [5], [6] related to object detection in an image. In those papers we found that they detect the object in an image which are not that much dense scenes such as occluded objects, objects closed to each other, or multiple objects jointly perceived as a single objects by human eye seen from distance.

In such densely packed scenes (Fig :1), most difficult part is to come up with an idea to remove near to coincide bounding boxes or whose IoU score is very much high among the predicted bounding boxes but their location is somewhat very much close to annotated boxes. Another difficult part is in these type of scenes is when two or more than two objects together looks like a single object to human eye also when seen from distance.

To resolve these problems, only relying on IoU score may not be sufficient, so in this SOTA method, IoU model is attached on Region Proposal Network (RPN) to come up with a new score i.e. Soft-IoU score. Expectation Maximization (EM) merger is used to represent detected bounding boxes as a mixture of Gaussian set and with by finding out their similarity/dissimilarity using KL divergent, they are merged/filtered.

A. Our Contributions

We went through some of the YOLO model that is YOLO9000 that can detect among 9000 different classes, batch normalization, anchor boxes are used in YOLO9000. YOLOv3 uses Darknet-53 as a backbone but requires higher execution time. In 2020, YOLOv5 [**?**] is introduced by Ultralytics. It is more user friendly and it uses PyTorch.
We used YOLOv5 to compare the previous model on SKU110K dataset.

Fig. 1. One of the image in SKU110K Dataset showing annotated bounding boxes

II. Related Work

Some of the object detectors use region proposals like R-CNN [1], Faster R-CNN [6], Fast R-CNN [2]. R-CNN is a class-agnostic object detector which uses selective search for proposing regions upto 2000. Finally it uses SVM as a classifier.

In Fast R-CNN, we do not need to recompute features on every region proposal as compared to region based CNN (R-CNN). It uses a single convolutional neural network on whole image and then propose Region of Interest (ROI) on this feature map. Thus making the detection process faster and more efficient. It also uses ROI pooling layer which is a maxpool layer on Region of Interest and then it's output is connected to a fully connected neural network which has softmax classifier and bounding box regressor.

In Faster R-CNN, it directly uses proposed regions of convolution network. Thats why it is faster than its predecessor. This eliminates the need for separate region proposal generation. But these methods require storing feature representations of all region proposals in memory, leading to high memory usage.

Other object detectors which do not uses region proposals are YOLO [4], SSD [3]. In YOLO, SxS grid made on an input image, when its center contains an object, it gives various bounding boxes and confidence score with respect to each bounding boxes. In SSD, using a single feature map it predicts bounding boxes and confidence score for different scales and its aspect ratios.

Another object detector that uses Feature Pyramid Network (FPN) model called RetinaNet [7] which introduced "focal

loss", which solves the "Class Imbalance" issue. i.e. When we have very less number of objects or overlapping objects in the image, it tries to give more priority to those object rather background which are abundance in nature.

Non-Maximum Suppression (NMS) [8] is a overlapping bounding box remover. By assuming some threshold, we sort all predicted boxes with object confidance score more than defined threshold. We take the box with maximum probability and repeat this process until no box is discarded. While NMS is widely used and effective, it does have drawbacks. One major limitation is that NMS is a deterministic approach, meaning it strictly enforces a threshold to decide whether to keep or discard a bounding box. As a result, it can be sensitive to the choice of the IoU threshold. Setting the threshold too high may result in missing some objects, while setting it too low may lead to keeping redundant detections.

Some NMS alternatives are there such as Mean-Shift [9], [10] and Agglomerative clustering [11]. In closely packed scenes, due to presence of overlapping bounding boxes, detection is pretty much difficult using above methods. Further, in paper, an unsupervised method is proposed for removing overlapping detections.

III. Proposed Methodology

Paper [12] proposed a model with baseline detector which contains ResNet50 as a backbone and Region Proposal Network (RPN) with Feature Pyramid Network (FPN). It gives three heads: (1) bounding box, (2) objectness score, (3) their novel soft-IoU score. It also includes EM Merger as their another novelty to split / merge the overlapping bounding boxes.

This model uses three types of losses: (1) Soft-IoU Loss i.e. binary cross-entropy loss defined in eq: . (2) Regression Loss, (3) Classification Loss.

EM Merger uses KL divergent algorithm to distinguish between different probabilistic distribution. It come up with K' number of bounding boxes by using covariance as a center of ellipse and eigen values as their axes. Which basically, takes those bounding box whose center lies inside the ellipse. These all are discussed in details later at implementation details.

IV. Experimental Evaluation

A. Implementation details

ResNet50 takes an image of size 224 x 224 px and has 5 conv. blocks named as $C1$, $C2$, $C3$, $C4$ and $C5$. On this ResNet50, RetinaNet model is constructed such as output of $C3$, $C4$ and $C5$ are used to make 5 feature pyramids named as $P3$, $P4$, $P5$, $P6$ and $P7$ as mentioned in Fig :6. P5 feature map is calculated by $C5$ using two $2D$ convolution with filter size 1 , stride= 1 and filter size 3 and stride= 1. $P6$ is created using $C5$ with filter size 3 and stride= 2. $P7$ is created using $P6$ by applying first ReLU activation function than $2D$ convolution of filter size 3 and stride= 2. $P4$ is created using $C4$ by first applying $2D$ convolution with filter size 1 , stride= 1 and added with upsampled of convoluted $C5$ and then again applied $2D$ convolution with filter size 3, stride= 1. Also, $P3$ is created using $C3$ by first applying $2D$ convolution with filter size 1 , stride= 1 and added with already upsampled feature as shown in Fig :6 then again applied $2D$ convolution with filter size 3, stride= 1. Finally we have all feature maps we need for Region Proposal Network (RPN).

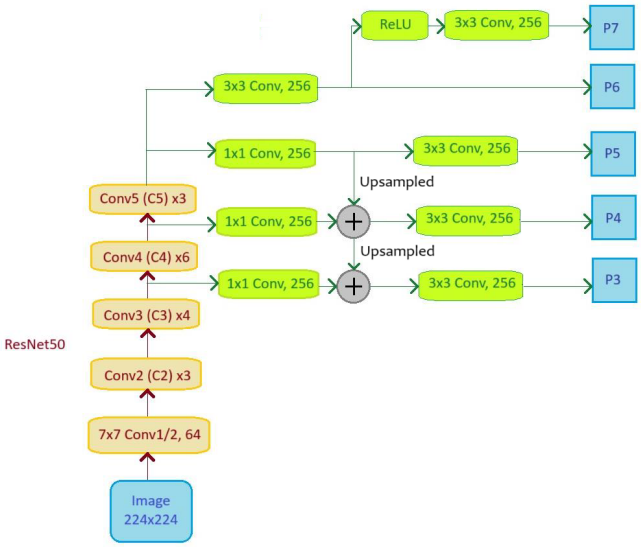

Fig. 2. Feature Pyramid Network in Baseline Detector

Further $P5$, $P6$, $P7$, $P4$ and $P3$ are used for anchor boxes of Region Proposal Network (RPN). On each of these features, nine anchor boxes are used with ratios 0.5, 1, 2 and scales 1, $2^{1/3}$, $2^{2/3}$. All these anchor boxes are then concatenated and further regressed and clipped and then filtered (Filter Detection function defined in the python code) using Non-Maximum Suppression (NMS) with predefined threshold of IOU 0.5. Further regression loss [4], [13] is used to train the model.

Also from each of these feature maps, IoU sub-modules are created and then concatenated to be used in earlier defined filtered detections. Classification sub-models are also created on these feature maps afterwards used in filter detection. A snippet of architecture of this model is given in fig: 7.

Classification model is created by four times $2D$ convolution with ReLU activation function and some parameters such as kernel initializer and bias initalizer as 0, further again convoluted with bias initializer as prior probability as 0.01. This output is further given to sigmoid activation function and this gives the objectness score (hard score) corresponding to each bounding box. Similarly, Intersection over Union (IoU) value is also calculated for

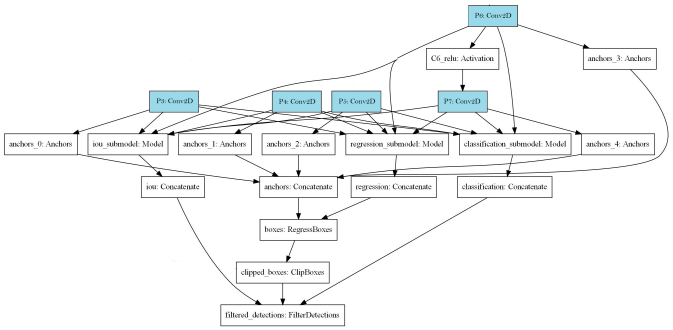

Fig. 3. A Snippet of Architecture of Model

each bounding box using IoU model convoluted as similar to above classification model except that no prior probability is used. Both these objectness score and IoU value are used to calculate Soft-IoU value. Soft-IoU score is the weighted sum of objectness score and the IoU value. A new loss i.e. Soft IoU loss in introduced which is mentioned as binary cross entropy loss given in $Eq.1$. This weight is experimentally evaluated which is 0.5 as hard score rate in this paper. This is the confidence score which is the sureness of containing the object in predicted bounding box.

Further, confidence score is used in Expectation Maximization (EM) merger (discussed later) to merge / split the overlapping bounding boxes by shrinking the boxes when needed.

All three detection heads such as bounding box, classification and Soft-IoU give bounding boxes coordinates and corresponding confidence score and hard score .

$$Soft-IoULoss = -\frac{1}{n}\sum_{i=1}^{n}[IoU_i log(C_i^{iou}) + (1-IoU_i) log(1-C_i^{iou})] \quad (1)$$

where number of images in each batch is n. Total loss is calculated as

$$TotalLoss = ClassificationLoss + RegressionLoss + Soft-IouLoss \quad (2)$$

Expectation Maximization merger:

Total M bounding boxes are constructed by model as two dimentional Gaussians set:

$$F = \{f_i\}_{i=1}^{M} = \{\mathcal{N}(q; \mu_i, \Sigma_i)\}_{i=1}^{M}, \quad (3)$$

where q is a two dimensional coordinate and for i^{th} predicted bounding box's center point is denoted by μ_i and covariance: $\Sigma_i = \begin{pmatrix} (height_i/4)^2 & 0 \\ 0 & (width_i/4)^2 \end{pmatrix}$

then all of these Gaussians are added with their mixture coefficient α_i which is defined as ratio of i^{th} box Soft-IoU score and sum of all M bounding boxes Soft-IoU score. It shows how much predicted box and annotated box are closed to one another. ($Eq.4$).

$$f(q) = \sum_{i=1}^{M} \alpha_i f_i(q) \quad (4)$$

Further, by using a mixture of Gaussian clustering method [14], [15], [16], [17] , accurate bounding boxes are detected which do not overlap with each other.

Now, we find K Gaussians which is much less than M Gaussians when accumulated, these Gaussians relatively represents original mixture of Gaussian distribution f ($Eq.4$)

$$G = \{g_{j=1}^{K}\} = set(\mathcal{N}(q; \mu_j^{'}, \Sigma_j))_{j=1}^{K} \quad (5)$$

We define g as,

$$g(q) = \sum_{j=1}^{K} \beta_j \}_j(q), \quad (6)$$

and for mixture of K Gaussians, weighted sum of minimum difference between two distributions f and g is,

$$d(f,g) = \sum_{i=1}^{M} \alpha_i \min_{j=1}^{K} KL(f_i || \}_j), \quad (7)$$

where KL is Kullback-Leibler-divergence, and here, it denotes how much two predicted boxes are distant from each other non-symmetrically.

Expectation Maximization Method:

Using this method ($Eq.7$) is minimized. Expectation step is defined by:

$$\pi(i) = arg \min_{j=1}^{K} KL(f_i || \}_j), \quad (8)$$

Model hyper parameters are estimated by Maximization step as:

$$\beta_j = \sum_{i \in \pi^{-1}(j)} \alpha_i \quad (9)$$

$$\mu_j^{'} = \frac{1}{\beta_j} \sum_{i \in \pi^{-1}(j)} \alpha_i \mu_i \quad (10)$$

$$\Sigma_i^{'} = \frac{1}{\beta_i} \sum_{j \in \pi^{-1}(i)} \alpha_j [\Sigma_j + [\mu_j - \mu_i^{'}][\mu_j - \mu_i^{'}]^T] \quad (11)$$

These hyper parameters are initialized by agglomerative, hierarchical clustering [18] . Clustering methods which maps input from higher dimension to lower dimension are not used here because data is in two dimension.

Finally, in post processing, by using some threshold, some Gaussians are discarded and now predictions reduced to $K^{'}$ which goes up to K. For final estimation of bounding boxes, an ellipse is constucted on the center of above $K^{'}$ predictions and only those M predictions are selected whose center lies inside the ellipse. Results are shown in $Table$ I along with SOTA implementation.

B. Details of the Dataset

SKU110K contains 11762 images of various retail shelves of different super markets clicked by ordinary phones not less than 5 MP resolution. And then compressed to JPG format mostly around 1 MP resolution. Phone models / cameras were not fixed in their specification.

The dataset contains 8233 images in training set i.e. 70% of total images. In total, training set has 1210431 objects. In test set, dataset contains 2941 images having 432312 objects. In validation set, dataset contains 588 images having 90968 objects. All objects were annotated on images by skilled workers. It was ensured that no two images are of same shelves in any one of the subsets of datasets.

File and Folder Structure of SKU110K

SKU110K dataset folder contains two sub-folder named as 'images' and 'annotations'. Images folder contains all train, test and validation images with their identifiable names. Their count are mentioned in previous paragraph. Annotation folder contains three CSV files for train, test and validation with their identifiable names. All three annotation file follows similar format. They contain all annotated boxes coordinates with their image file name, image height and width. (Filename, top left corner coordinate($X1$, $Y1$), bottom right corner coordinate($X2$, $Y2$), image width, image height)

V. Novelty

We used YOLOv2 [5] model to test SKU110K dataset in which we changed some parameters such as image input size and grid cells.

A. Implementation Details

YOLOv2 uses DarkNet-19 as a backbone which has 19 layers and output feature map size is 13×13 by default. In original paper, it uses a pass through layer from feature map size 26×26 to output feature map 13×13 such that higher dimensional feature information cal also be used for detections.

We modified grid cells from 13×13 to 15×15 assuming we can capture densely packed objects. Only odd number of grid cells can be used so that we can find a single center point.

We also modify input image size to 480×480 . As required by YOLOv2 architecture, it will be divided by 32 to get the final output feature maps. Thus final output feature map size is 15×15.

Results are shown in $Table$ I along with SOTA implementation.

VI. Results

Paper followed evaluation assessment as given in COCO [19] such as average precision (AP) is calculated on IoU = 0.5:0.05:0.95. $AP^{.75}$ is calculated on only IoU = 0.75. AP is the area-under-curve(AUC) of Precision vs. Recall curve. AR^{300} is calculated as average recall with maximum 300 bounding box (if number of bounding box exceeds 300, then top 300 bounding box will be taken based on confidence score). $P^{R=.5}$ is the precision at recall = 0.5. MAE is the Mean Absolute Error, calculated as $\frac{1}{n}\sum_i^n |K_i^{'} - t_i|$ and RMSE is the Root Mean Squared Error, calculated as $\sqrt{\frac{1}{n}\sum_i^n (K_i^{'} - t_i)^2}$, where $\{K_i^{'}\}_{i=1}^n$ is the number of detected bounding boxes in i^{th} test image and $\{t_i\}_{i=1}^n$ is the number of annotated bounding box for each test image. Also object detection result with bounding box is shown in $Fig.4$ and $Fig.5$.

Fig. 4. SOTA Result with IoU=0.75 (Our SOTA Implementation)

The $Table$I shows the comparison between the SOTA method and Novelty results and past work results.

Runtime Analysis: FPS is the frames per second i.e. per second the number of image processed. And DPS is the detections per second i.e. the number of bounding box predicted in a second from the test images.

We measure the runtime on GPU specification: 2x Nvidia V100 PCIE Accl. cards each with 5120 CUDA cores 16 GB

Method	AP	$AP^{.75}$	AR^{300}	$P^{R=.5}$	MAE	RMSE
Faster-RCNN [13]	0.045	0.01	0.066	0	107.46	113.42
YOLO9000 [5]	0.094	0.073	0.111	0	84.166	97.809
RetinaNet [20]	0.455	0.389	0.53	0.544	16.584	30.702
SOTA (Paper) [12]	0.492	0.556	0.554	0.834	14.522	23.992
Our SOTA	0.516	0.560	0.549	0.824	15.741	25.272
Our Novelty	0.056	0.029	0.102	0.003	94.256	106.546

TABLE I
COMPARISION OF OBJECT DETECTION METHODS

Fig. 5. Novelty Result (Our Novelty Implementation)

Method	FPS	DPS
Faster-RCNN [13]	2.37	93
YOLO9000 [5]	5	317
RetinaNet [20]	0.5	162
SOTA (Paper Claimed) [12]	0.23	73
Our SOTA Implementation	0.25	79
Our Novelty Implementation	6.03	328

TABLE II
PROCESSING SPEEDS FOR DIFFERENT METHODS

HBM2 and CPU specification: 2x Intel Xeon Skylake 6148, 20 cores of 2.4 GHz. ($Table$ II)

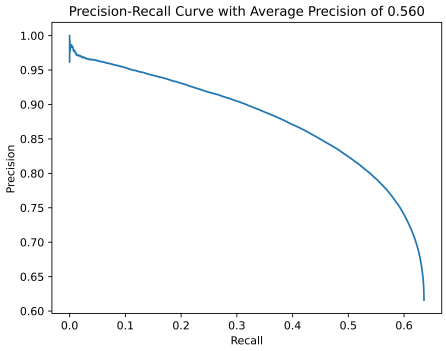

Fig. 6. Precision-Recall Curve with IoU=0.75 (Our SOTA Implementation)

Miss rate (MR) is the ratio of True Positive (TP) and the sum of True Positive and False Negative (FN). Sum of True Positive and False Negative results the number of ground truth (GT) box. FPPI is the number of False Positive per image. ($Fig.7$) $MissRate = \frac{TP}{TP+FN} = \frac{No.ofGTbox-TP}{No.ofGTbox}$

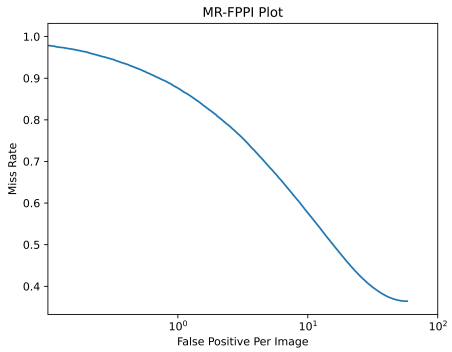

Fig. 7. Miss Rate vs False Positives Per Image with IoU=0.75 (Our SOTA Implementation)

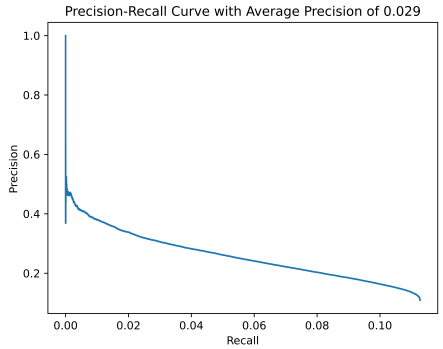

Fig. 8. Precision-Recall Curve with IoU=0.75 (Our Novelty Implementation)

VII. CONCLUSION

SOTA Method:

We tested this model on SKU110K dataset and found remarkable achievements in precision but still there is chance of improvement. We got improvement in average precision with respect to paper because the code is more optimized over time with fine tuning the parameters. There are two novelties in the paper: (1) Soft-IoU which is much better than standard IoU because it rely on objectness score as well as IoU score. Thus, it takes the account of confidence of presence of object inside the bounding box as well as overlapping of predicted bounding box with ground truth. (2) Expectation Maximization (EM) merger which is much more reliable than Non-maximum Suppression (NMS) because it uses mixture of Gaussions and does not only depend on IoU value.

We also tested this dataset on YOLOv2 with some modifications.

Novelty:

On YOLOv2 [5] we tested SKU110K dataset and found no improvement in average precision, it may be due to as YOLO is single stage detector and it more focus on bigger objects.

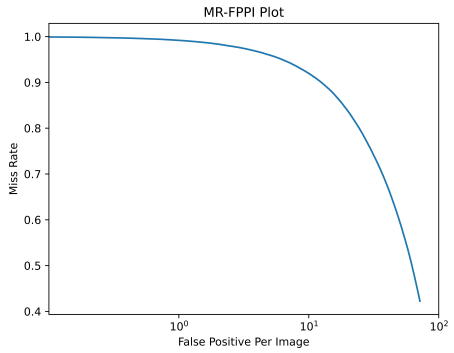

Fig. 9. Miss rate vs False Positives Per Image with IoU=0.75 (Our Novelty Implementation)

Because receptive field is high on output feature map and it is also mentioned in the paper that the large objects usually tries to occupy the center of image that's why YOLOv2 used odd number of grid cells to make single location at the center.

We got improvement in the detection speed and it reflects on FPS and DPS.

REFERENCES

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. Conf. Comput. Vision Pattern Recognition*, 2014.

[2] R. B. Girshick, "Fast r-cnn," in *Proc. Int. Conf. Comput. Vision*. IEEE Computer Society, 2015.

[3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European Conf. Comput. Vision*, 2016.

[4] J. Redmon, S. K. Divvala, and R. B. Girshick, "You only look once: Unified, real-time object detection," in *Proc. Conf. Comput. Vision Pattern Recognition*, A. Farhadi, Ed., 2016.

[5] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *Proc. Conf. Comput. Vision Pattern Recognition*, 2017.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Neural Inform. Process. Syst.*, 2015.

[7] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. D. a, "Focal loss for dense object detection," in *Trans. Pattern Anal. Mach. Intell.*, 2018.

[8] P. Viola and M. J. Jones, "Robust real-time face detection," in *Int. J. Comput. Vision*, 2004.

[9] N. Dalal and B. Triggs., "Histograms of oriented gradients for human detection," in *Proc. Conf. Comput. Vision Pattern Recognition. IEEE*, 2005.

[10] C. Wojek, G. D. o, A. e Schulz, and B. Schiele, "Sliding-windows for rapid object class localization: A parallel technique," in *Joint Pattern Recognition Symposium. Springer*, 2008.

[11] L. Bourdev, S. Maji, T. Brox, and J. Malik, "Detecting people using mutually consistent poselet activations," in *European Conf. Comput. Vision. Springer*, 2010.

[12] E. Goldman, R. Herzig, A. Eisenschtat, J. Goldberger, and T. Hassner, "Precise detection in densely packed scenes," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Neural Inform. Process. Syst.*, 2015.

[14] P. Bruneau, M. Gelgon, and F. Picarougne, "Parsimonious reduction of gaussian mixture models with a variational bayes approach," in *Pattern Recognition*, 10.

[15] J. Goldberger, H. K. Greenspan, and J. Dreyfuss, "Simplifying mixture models using the unscented transform," in *Trans. Pattern Anal. Mach. Intell.*, 2008.

[16] J. Goldberger and S. T. Roweis, "Hierarchical clustering of a mixture model," in *Neural Inform. Process. Syst.*, 2005.

[17] K. Zhang and J. T. Kwok, "Simplifying mixture models through function approximation," in *Neural Inform. Process. Syst.*, 2007.

[18] L. Rokach and O. Maimon, "Clustering methods," in *Data mining and knowledge discovery handbook*, 2005.

[19] T.-Y. Lin, S. B. Michael Maire, J. Hays, P. Perona, D. Ramanan, P. D. r, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conf. Comput. Vision*, 2014.

[20] Tsung-YiLin, PriyalGoyal, K. RossGirshick, and P. D. r, "Focal loss for dense object detection," in *Trans. Pattern Anal. Mach. Intell.*, 2018.