Learning Gaussian Maps for Dense Object Detection

Sonaal Kant
sonaal@paralleldots.com

Paralleldots Inc.
https://www.paralleldots.com/

Abstract

Object detection is a popular branch of research in computer vision, many state of the art object detection algorithms have been introduced in the recent past, but how good are those object detectors when it comes to dense object detection? In this paper we review common and highly accurate object detection methods on the scenes where numerous similar looking objects are placed in close proximity with each other. We also show that, multi-task learning of gaussian maps along with classification and bounding box regression gives us a significant boost in accuracy over the baseline. We introduce Gaussian Layer and Gaussian Decoder in the existing RetinaNet network for better accuracy in dense scenes, with the same computational cost as the RetinaNet. We show the gain of 6% and 5% in mAP with respect to baseline RetinaNet. Our method also achieves the state of the art accuracy on the SKU110K [10] dataset.

1 Introduction

Computer vision as a field has grown from research to more of an applied field. Many industries are using computer vision either to enhance their existing technology or to create an altogether new product around it. Either way, object detection algorithms play a crucial role in almost every aspect. It has attracted much attention in the computer vision field because of its numerous real world applications, from Self driving cars to Surveillance many applications require an object detection algorithm. Similar to these, companies are also using object detection in retail stores to maximize sales and store inventory management. Recent work from Fuchs et al. [6] show the computer vision challenges in supermarket or retail stores environment. They have also shown results from transfer learning for image-based product classification and multi-product object detection, using multiple CNN architectures on the images of vending machines.

Unlike popular object detection datasets such as ILSVRC [2], PASCAL VOC [13] detection challenges, MS COCO [15], and the very recent Open Images v4 [14] the retail stores based datasets such as [10] [29] are more densely packed. The annotations for the WebMarket [29], CAPG-GP [6] and Grocery Products [2] have been released by [27] for more robust comparison of object detection algorithms on retail stores datasets. The problem while working on the densely packed datasets is that the very similar looking objects are placed in close proximity with each other which makes it difficult for the object detection algorithms to find the boundaries and hence result in many overlapping bounding boxes with high objectness score.

Object detection algorithms have evolved in many years, starting off with the two-stage detection method RCNN [9] and its faster successors such as FastRCNN [8] and Faster-RCNN [24] which introduced a region proposal network (RPN). This later was improved by Mask-RCNN [12] by adding a segmentation output as a multi-task learning approach. Evolving from two-stage to single-stage detection for better and faster results, YOLO [23], SSD [18], and YOLO9000 [22] were introduced which removed the need of proposals from the algorithms. Recent work from [10] shows how badly the standard object detection methods fail in the case of densely packed scenes. Research by Lin et. al [17] showed that the foreground-background class imbalance is the reason why these state of the art detectors performed poorly. They introduced feature pyramid network (FPN) [16] with focal loss to handle the class imbalance and scale variance.

Contributions We extend the work of Lin et al. [17] by adding an auxiliary loss to the existing RetinaNet architecture. We show that sharing representations between related tasks can enable our model to generalize better on our original task. We introduce a gaussian loss as an auxiliary branch for predicting a low resolution, per-pixel heat-map, describing the likelihood of a object centre in each spatial location in parallel with the existing branch for bounding box regression and object classification. We try to overcome the limitation of detecting objects in close proximity by enforcing the network to learn less likelihood for the pixels which are not the centers of object, hence making it easier for the anchors to learn the boundaries. We introduce two different network architectures to emphasize the importance of multi-task learning in object detection for densely packed scenes.

1. Gaussian Layer Network is a multi-task learning approach where we share the common backbone with the feature pyramid network to predict the set of 2d gaussian in the image. We show that by adding a simple gaussian layer to your model as an auxiliary task will give you an accuracy boost with no extra inference time.

2. Gaussian Decoder Network is introduced as an experiment to show that the extra auxiliary loss just on the encoder only can help us achieve similar accuracy reported by [10] on SKU-110k dataset.

Both architectures show the improvement in accuracy in SKU-110K [10] and other groceries datasets such as WebMarket [29], GroceryProducts [7], CAPG-GP [6] with the baseline.

2 Related Work

Goldman et. al [10] have recently released their SKU-110K dataset. The dataset represents various possible dense object detection examples at different scales, angles and containing different types of noise. Different brands and products which are often distinguishable only by fine-grained differences are kept in close proximity with each other. It contains 8233 training images, 588 validation images and 2941 testing images containing objects of different aspect ratios, quality and different lighting conditions. This variety in the dataset makes it a good benchmark to evaluate the performance of the object detection algorithm in densely packed scenes. They have also shown the performance of state of the art object detection algorithms such as Faster-RCNN [24], YOLO9000 [22] on their dataset to compare it with their approach. They have extended their work on RetinaNet [17] by introducing another head along with a classifier and the bounding box regressor and they call it a Soft-IoU layer. They proposed that the classifier predicts the objectness score which is not sufficient in

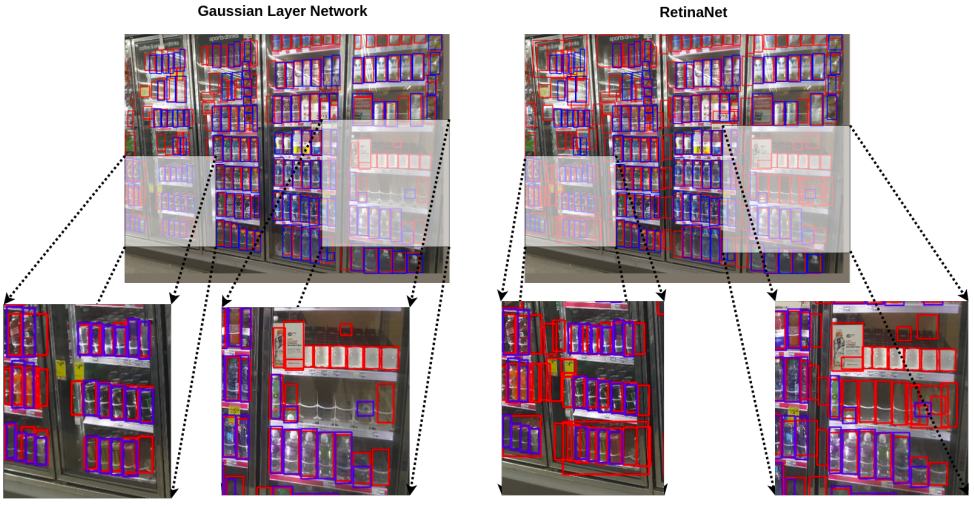


Figure 1: **Example Prediction In SKU-110K Dataset**. (Left) Detection results of Our Method. (Right) Detection Results of baseline RetinaNet. The red bounding box is the prediction and the blue bounding box is the groundtruth.

dense images because of the multiple overlapping bounding boxes which often reflect multiple tightly packed objects. So to handle these cluttered predictions they introduce IoU score as an additional value for every predicted bounding box with the object. In order to handle the multiple predicted bounding box they introduce an approach that replaces Non Maximum Suppression (NMS) with EM-Merger. This takes the predictions as a set of 2D Gaussians and performs Gaussian Mixture Modeling whereas we try two merge this two step process by training our model to predict the gaussian of every object as an auxillary task. This will ultimately help to learn to give better anchors instead of relying on the naive post processing step.

Instance Segmentation. The research community has started shifting their attention to the more complex task of instance segmentation, while the object detection methods give the bounding box for each object, the segmentation models give the pixel-level mask for that object. K He et al. [12] came up with the multi-task approach of training an object detection method along with the instance segmentation. They called their multi-task architecture as MaskRCNN which has an additional branch for predicting segmentation masks on each Region of Interest (RoI) in a pixel-to pixel manner. This multi-task training approach proves to be better than the normal FasterRCNN [24], hence giving the accuracy boost on both the object detection and Instance segmentation tasks. Path Aggregation Network extends the idea of MaskRCNN by introducing the bilinear Interpolation in their ROI Align module. Following the idea of training object detection and segmentation, Fu et al. [4] introduce RetinaMask which is the extended version of RetinaNet. In our approach where the objects are densely placed together, predicting the class at every pixel makes it difficult for the network to learn the objectness. So instead we only predict the gaussian with maxima at the centre of the object.

Gaussian Based detection. Many problems such as Human Pose Estimation [28] [21], Face Keypoint detection [19] etc. uses gaussian maps in their approach. Similar to ours, Baek

et. al [1] used gaussian map prediction to localize the characters in scene text localization tasks. The performance of their algorithm shows the effectiveness of this method. They train a weakly supervised method to detect each individual character using a gaussian map which they call Region score. Despite being a completely different problem it relates to our retail based densely packed scenes. The numerous similar looking characters which are stacked together in a word correlates with the products which are placed together on the shelf. Inspired by his idea we add this gaussian map to our model but instead of localizing directly through the gausian map we train a seperate branch for bounding box regression with shared features.

3 Motivation

Many Object Detection algorithms and their variants have been proposed for detection tasks like PASCAL [13], COCO [15], but the object detection in dense scenes is still an area which is not much explored. Retail stores and supermarkets are the perfect case in point for densely packed scenes. They contain similar looking products which are very large in number and placed in close proximity with each other. Recent study by [10] has shown that state of the art object detectors like YOLO [23], Faster-RCNN [24], fail to perform well when it comes to dense object scenes. The performance is drastically improved by the RetinaNet [17] architecture with focal loss because of its ability to handle positive-negative class imbalance while training. Goldman et al. [10] introduced an EM-Merger module which is a gaussian mixture model to merge all the predictions in a post processing step instead of using standard Non Maximum Suppression(NMS). Our work aims to remove this two-step process and convert it to an end to end model which learns to give precise boxes through the supervision of gaussian. We hypothesize that instead of using a post processing method, adding an auxiliary loss of gaussian map to the RetinaNet architecture and performing a multi-task learning approach will directly help the anchors to learn the better boundaries of the object and will also help the network to generalize better.

4 Baseline

We use RetinaNet [17] as our baseline as it has been proven to work better than Faster-RCNN [24]. The reason for this is, Faster-RCNN [24] uses Region Proposal Network for bounding box regression and classification on top of high level feature map which losses lots of semantic information thus unable to detect small objects. RetinaNet uses Feature Pyramid Network (FPN) that naturally leverages the pyramidal shape of a Convnet feature hierarchy while creating a feature pyramid that has strong semantics at all scales, hence solving the problem of detecting small objects. The class imbalance is another reason why we use RetinaNet as our baseline. Many object detection algorithms face the problem of huge class imbalance because of less positive anchors and very large number of negative anchors. Similar imbalance problem was addressed by Abhinav et al.[26] while training object detection algorithms. To overcome this problem RetinaNet [17] used FocalLoss for classification. Focal loss is an extension of cross entropy loss that down-weights the loss assigned to easy negatives hence preventing the easy negatives to harm the detector during training. We determine the positive and negative anchors during training using the overlap with ground truth boxes. The classification subnet returns the objectness probability p for

every anchor whereas the regression subnet returns the offset x for every positive anchor.

$$pt = \begin{cases} p & \text{if } y = 1, \\ 1-p & \text{otherwise} \end{cases}$$

$$L_{cls}(pt) = \alpha(1-pt)^{\gamma}\log(pt) \tag{1}$$

In FocalLoss (1), we use γ and α as mentioned in the original paper that is 2, 0.25 respectively. Unlike classification subnet that uses focal loss instead of cross entropy loss, the bounding box subnet uses the standard Smooth L1 loss (2) that is applied on all positive anchors.

$$L_{reg} = \begin{cases} 0.5x^2 & \text{if } |x| < 1, \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{2}$$

5 Our Approach

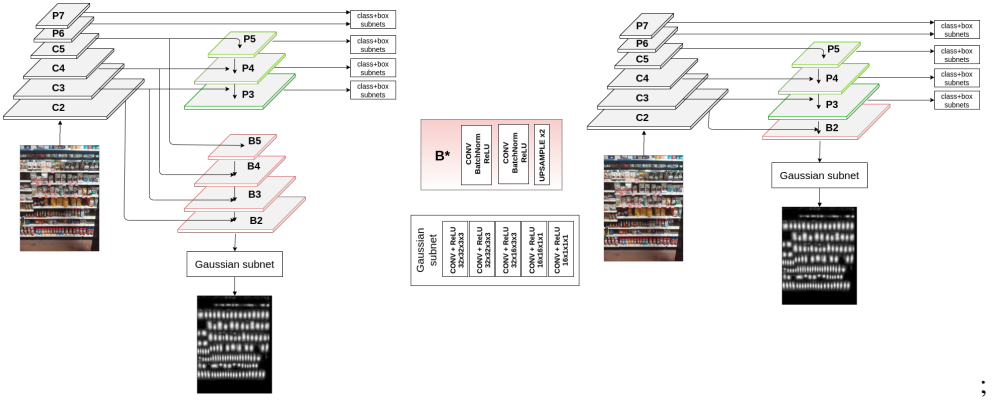
Baek et al [1] used gaussian heatmap for predicting the character level bounding boxes for scene text detection. The scene text detection datasets have numerous number of words and in a word the characters are close together and almost similar looking. This trend in the scene text dataset can be seen in our densely packed scenes. Similar to that, we can see our objects as a 2D gaussian with its peak at the center of the object and the σ_x and σ_y of that gaussian is defined by the width and the height of the object. Scene text datasets don't have character level bounding box annotation that is why they do weakly supervised character detection whereas we have the bounding box annotation for every object so we perform a fully supervised training by generating the gaussian heatmaps using the ground truth bounding box. For each training image, we generate the ground truth gaussian map using the object ground truth bounding box. The gaussian map is a set of 2d gaussians for every object in the training image. Every 2d gaussian represents the object with the highest probability at the center of the object. To generate the gaussian map, we first make a square gaussian of size 120 and sigma 40. For every bounding box in the training image we find the homography $\mathbf{H}$ using four point transform $\mathbf{P}$, which is then applied to the gaussian $\mathbf{G}$ to wrap it to the box area. We consider N ground truth bounding boxes $B_i \in R^2$ and convert them to $2D$ gaussians, we start with an empty image $\mathbf{I}$

$$\mathbf{G} = \exp^{-4\log 2((x-x_c)^2+(y-y_c)^2)/\sigma^2}$$

$$\mathbf{H} = \{h_i\}_{i=1}^{N} = \{P(B_i)\}_{i=1}^{N}$$

$$\mathbf{I} = \mathbf{I} + \mathbf{H}(\mathbf{G})$$

We train RetinaNet from scratch with Resnet50 [11] as the backbone which is pretrained on ImageNet [3]. We would like to emphasize on the point that we do multi-task learning training with additional Mean Squared Loss(MSE) with hard example mining on the output gaussian map $\mathbf{I}^*$ added to the existing RetinaNet architecture and we call this a gaussian loss. We create two empty masks, δ_n for negative sampling and δ_p for positive sampling,



;

Figure 2: **Left :** Gaussian Decoder Network. A standard UNet Achitectecture is used with B2, B3, B4, B5 layers as a decoder. **Right :** Gaussian Layer Network. Instead of adding an additional decoder for gaussian map we add an extra layer B2 and call it as a gaussian layer. Intrinsic details of the gaussain layer and decoder can be seen in B*.

of dimension $\mathbf{I}$. δ_n is activated when the target is less than equal negative thresh and δ_p is activated when it is greater than equal positive thresh.

$$\mathcal{L}_{gl} = \frac{1}{n}\sum_{i=1}^{n}\sum_{xy}(\delta_n||\mathbf{I}_{xy} - \mathbf{I}^*_{xy}||^2 + \delta_p||\mathbf{I}_{xy} - \mathbf{I}^*_{xy}||^2) \tag{3}$$

5.1 Gaussian Layer Network (GLN)

We propose a Gaussian Layer and Gaussian subnet in the RetinaNet architecture for gaussian map prediction. The main idea behind the gaussian layer is to help the network correctly predict centres of the objects which are very similar looking and are placed close to each other. This additional task of predicting centres helps the anchors to learn better boundaries using the combination of low resolution semantically strong features with the high resolution semantically weak features using skip connections. Similar to the bbox subnet and class subnet introduced in RetinaNet, we introduce gaussian subnet on top of the gaussian layer which has sequence of convolution, batchnorm and relu blocks as shown in Figure 2.

Our Gaussian Layer Network is a multitask learning architecture with shared encoder and decoder. We take the concatenation of low level features C2 and P3 as the input for the gaussian layer ($B2$). The output features from the gaussian layer are then passed to the gaussian subnet. The anchors from layers P3, P4, P5, P6, P7 are trained using the standard regression and classification loss. In addition to this, a gaussian loss is applied on the output of the gaussian subnet. We hypothesize that, gaussian loss will not only refine the anchors from P3, P4, P5 but will also enhance the low level features C2. The final output from the gaussian subnet is a single channel map of size (H/2, W/2) where H and W is the height and width of the original image. The final loss is calculated as the weighted sum of classification (1), regression (2) and gaussian loss (3).

$$\mathcal{L}_{total} = \lambda_1\mathcal{L}_{cls} + \lambda_2\mathcal{L}_{reg} + \lambda_3\mathcal{L}_{gl} \tag{4}$$

5.2 Gaussian Decoder Network (GDN)

Gaussian Decoder Network is an extended version of RetinaNet similar to Gaussian Layer Network but with the different decoder. The idea behind introducing the Gaussian Decoder Network is to show the importance of predicting gaussian centres with just the shared encoder. This simple idea of training with an auxiallry task only on the encoder gives you the similar accuracy to previous state of the art. Similar to RetinaNet we use Resnet50 [11] as the encoder but instead of using the feature pyramid network as the decoder we propose a separate decoder which predicts the sets of 2d gaussians of every objects in the image. The feature pyramid network predicts the bounding box and the classes at every level. In order to predict the gaussian center of each product in the image, the network should have an idea of "what" and "where", which means what are the objects which the network has to predict and where are the center of those objects. Ronneberger et. al. [25] showed that the U shape architecture which has a contracting path mainly consists of convolutional and downsampling layers and the expansive path which consists of transpose-2d convolutional layers for upsampling along with the skip connections to concatenate the features from contracting path is a good architecture for this. The same idea is used to design the decoder of the GDN. As shown in Figure 2, layers C2, C3, C4, C5 of the encoder Resnet50 [11] are used as skip connections to the decoder. The layers B2, B3, B4, B5 in decoder consists of convolution, batchnorm and relu followed by an interpolation of 2x.The interpolated output from B2 of size H/2, W/2 is then pass to gaussain subnet for final gaussian map prediction.

6 Experiments & Results

We train one model of each proposed approach on the SKU-110K training set which has 8233 images and use the checkpoint with best performance on the validation set that has 556 image. These models trained on SKU-110K [10] dataset are tested not just on the test set of SKU-110K but also on WebMarket [29], GroceryProducts [7], CAPG-GP [6] and Holoselecta [5]. All the implementation is done in Pytorch [20]. We compare our method on SKU-110K datasets with the baselines and the full approach given by Goldman et. al [10] and we also add another baseline of MaskRCNN to compare our method with the multi-task segmentation approach. We also compare the full approach given by Goldman et. al [10] on other retail based datasets. We want to make clear that we used the improved weights given by the authors which are better than the one they reported in the paper [10]. The link of those weights can be found here.

6.1 Training

We train all our experiments by following the settings given in the original RetinaNet [17] paper. Input images are resized by keeping minimum dimension as 800 and maximum dimension as 1333. All our models are trained on a single 1080ti GPU, as some of the models take larger GPU RAM we keep the batch size as 1 for all training. To compare our models well with the previously trained methods we keep all the hyperparameters constant as mentioned in the original paper. We take the anchor boxes on feature pyramid levels P3 to P7. Every anchor box is matched with a single ground truth bounding box and all the anchors that have intersection over union overlap greater than 0.5 are taken as positive anchors and those with less than 0.4 are taken as negative, rest all the anchors are ignored from training.

Method	AP	$AP^{.50}$	$AP^{.75}$	AR_{300}	AR^{50}_{300}
Faster-RCNN [24]	0.045	-	0.010	0.066	-
YOLO9000 [22]	0.094	-	0.073	0.111	-
RetinaNet [17]	0.455	-	0.389	0.530	-
Goldman et. al [10]	0.492	-	0.556	0.554	-
Goldman et. al*	0.514	0.853	0.569	0.571	0.872
MaskRCNN [12]	0.403	0.742	0.396	0.465	0.778
Gaussian Decoder	0.512	0.878	0.552	0.582	0.917
Gaussian Layer	**0.521**	**0.891**	**0.562**	**0.596**	**0.931**

Table 1: Performance of *our* approach on SKU-110K dataset. We compare our model also with the baselines provided by [10]. * denotes results obtained using the improved model given by the authors at URL

Method	FPS	DPS
Faster-RCNN [24]	2.37	93
YOLO9000 [22]	5	317
RetinaNet [17]	0.5	162
Goldman et. al [10]	0.23	73
Gaussian Decoder	0.5	162
Gaussian Layer	0.5	162

Table 2: Detection run-time comparison on SKU-110k

We then train our network with Focal Loss for classification, regression for bounding box and L2 Norm with hard example mining for gaussian maps till the best validation loss is not achieved. We have also shown MaskRCNN [12] as our one of the baselines, we have used the implementation provided by Pytorch [20] with Resnet 50 as the backbone which is common for all the networks we have trained as well the current state of the art model on SKU-110K [10] dataset.

6.2 Evaluation

We evaluate all methods using the COCO [15] object detection metric. In Table 1 and Table 3 we report the average precision (AP) at IoU=0.50:.05:0.95, AP at IoU=0.5, AP at IoU=0.75, average recall (AR_{300}) at IoU=0.50:.05:0.95, AR_{300} at IoU=0.5. The average recall is calculated on top 300 predictions of the model. The evaluation on inference time can be seen in Table 2. We report frames per second (FPS) and detections per second (DPS). Both our methods, Gaussian decoder and Gaussian Layer network uses an auxiliary task for training purpose only. During test time the auxilary weights are removed and the gaussian map is not predicted. Hence, the inference time is same as Retinanet.

6.3 Comparison on SKU-110K

SKU-110K test set comprises of 2941 images with total 432,312 ground truth bounding boxes which makes it approximately 146 objects per image, similar statistics belong to the training set. We compare our models with baselines provided by [10]. We also add MaskR-CNN as another baseline to the list for future work comparisons. As shown in Table 1, Gaussian Decoder and Gaussian Layer Network outperforms the baseline RetinaNet with

Dataset	Method	AP	$AP^{.50}$	$AP^{.75}$	AR_{300}	AR^{50}_{300}
WebMarket[29]	Goldman et. al*	0.383	0.773	0.332	0.491	0.855
	Gaussian Decoder	0.397	0.798	0.340	0.547	0.946
	Gaussian Layer	**0.403**	**0.813**	**0.340**	**0.551**	**0.954**
Holoselecta[5]	Goldman et. al*	**0.454**	**0.835**	**0.447**	**0.581**	**0.955**
	Gaussian Decoder	0.368	0.717	0.316	0.497	0.842
	Gaussian Layer	0.384	0.705	0.368	0.524	0.843
GP[7]	Goldman et. al*	0.259	0.520	0.241	0.403	0.716
	Gaussian Decoder	0.494	0.846	0.539	0.623	0.967
	Gaussian Layer	**0.506**	**0.862**	**0.548**	**0.634**	**0.975**
CAPG-GP[6]	Goldman et. al*	0.431	0.684	0.519	0.481	0.721
	Gaussian Decoder	0.482	**0.782**	0.573	0.542	**0.819**
	Gaussian Layer	**0.510**	0.777	**0.616**	**0.572**	0.816

Table 3: Performance of *our* approach across different general product datasets. * denotes results obtained using the trained model given at URL as is.

approximately 5% and 6% respectively. This accuracy gain on the baseline validates our hypothesis of performing multitask learning with gaussian maps.We also show the improvement in accuracy in comparison with the previous state of the art method. We want to make it clear that our final accuracy is better than the numbers reported in the paper by 3% and also than the weights given by the author in his github repository by 0.8%.

6.4 Comparision on Other Datasets

We also compare our trained model on different datasets [29] [7] [6] [5], unlike SKU-110k these datasets are not that dense, the number of ground truth bounding box per image are 37, 13, 20 and 34 respectively. We want to clarify that we have not fine tuned our model on any of these datasets and while training there were no augmentations with respect to different scale and size. [27] has given a detailed analysis on these datasets with the general object annotations which we use to compare our model accuracy. As shown in Table 3, our Gaussian Decoder and Gaussian Layer Network outperforms the model given by [10] on WebMarket [29], Grocery Products [7] and CAPG-GP [6] dataset by a large margin, where as we see a drastic performance loss in the Holoselecta [5] dataset. The performance drop in the Holoselecta dataset is observed because of their varied image dimensions and the object scale variance in the datasets, these mistakes can be easily solved with multi-scale testing or training but we perform single scale testing on all the datasets for fair comparison.

7 Conclusion

In this work, we proposed an additional multi-task training on the existent RetinaNet architecture. As shown in Fig. 1 gaussian layer network does not confuses with the background as much as the simple RetinaNet because of the gaussian map training, the network now is more robust to background objects and can distinguish better between objects placed in close proximity. This gives the significant boost in accuracy in various datasets without any overhead. Our proposed gaussian decoder network shows the affect of multitask training with shared encoder whereas gaussian layer network shows the same with shared encoder

and decoder. The improvement in accuracy from gaussian decoder to gaussian layer network also proves our hypothesis of having shared representations for the anchors.

References

[1] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9365–9374, 2019.

[2] J Deng, A Berg, S Satheesh, H Su, A Khosla, and L Fei-Fei. Ilsvrc-2012, 2012. *URL http://www. image-net. org/challenges/LSVRC*, 3, 2012.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[4] Cheng-Yang Fu, Mykhailo Shvets, and Alexander C Berg. Retinamask: Learning to predict masks improves state-of-the-art single-shot detection for free. *arXiv preprint arXiv:1901.03353*, 2019.

[5] Klaus Fuchs, Tobias Grundmann, and Elgar Fleisch. Towards identification of packaged products via computer vision: Convolutional neural networks for object detection and image classification in retail environments. In *Proceedings of the 9th International Conference on the Internet of Things*, IoT 2019, pages 26:1–26:8, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-7207-7. doi: 10.1145/3365871.3365899. URL http://doi.acm.org/10.1145/3365871.3365899.

[6] Weidong Geng, Feilin Han, Jiangke Lin, Liuyi Zhu, Jieming Bai, Suzhen Wang, Lin He, Qiang Xiao, and Zhangjiong Lai. Fine-grained grocery product recognition by one-shot learning. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, pages 1706–1714, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5665-7. doi: 10.1145/3240508.3240522. URL http://doi.acm.org/10.1145/3240508.3240522.

[7] Marian George and Christian Floerkemeier. Recognizing products: A per-exemplar multi-label image classification approach. pages 440–455, 09 2014. doi: 10.1007/978-3-319-10605-2_29.

[8] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[10] Eran Goldman, Roei Herzig, Aviv Eisenschtat, Oria Ratzon, Itsik Levi, Jacob Goldberger, and Tal Hassner. Precise detection in densely packed scenes. *CoRR*, abs/1904.00853, 2019. URL http://arxiv.org/abs/1904.00853.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[13] Derek Hoiem, Santosh K Divvala, and James H Hays. Pascal voc 2008 challenge. In *PASCAL challenge workshop in ECCV*. Citeseer, 2009.

[14] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*, 2018.

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[16] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[18] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015. URL http://arxiv.org/abs/1512.02325.

[19] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Posefix: Model-agnostic general human pose refinement network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7773–7781, 2019.

[20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learn pdf.

[21] Xi Peng, Zhiqiang Tang, Fei Yang, Rogerio S Feris, and Dimitris Metaxas. Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2226–2234, 2018.

[22] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[23] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[26] Abhinav Shrivastava, Abhinav Gupta, and Ross B. Girshick. Training region-based object detectors with online hard example mining. *CoRR*, abs/1604.03540, 2016. URL http://arxiv.org/abs/1604.03540.

[27] Srikrishna Varadarajan, Sonaal Kant, and Muktabh Mayank Srivastava. Benchmark for generic product detection: A low data baseline for dense object detection. *arXiv*, pages arXiv–1912, 2019.

[28] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. *arXiv preprint arXiv:1910.06278*, 2019.

[29] Yuhang Zhang, Lei Wang, Richard I. Hartley, and Hongdong Li. Where's the weet-bix? In Yasushi Yagi, Sing Bing Kang, In-So Kweon, and Hongbin Zha, editors, *ACCV (1)*, volume 4843 of *Lecture Notes in Computer Science*, pages 800–810. Springer, 2007.