REGION AVERAGE POOLING FOR CONTEXT-AWARE OBJECT DETECTION

Kingsley Kuan[1], Gaurav Manek[1], Jie Lin[1], Yuan Fang[1], Vijay Chandrasekhar[1,2]

Institute for Infocomm Research, A*STAR, Singapore[1]
Nanyang Technological University, Singapore[2]

ABSTRACT

Object detection has been a key task in computer vision with deep convolutional neural networks being a significant performer. We propose a method named Region Average Pooling that leverages object co-occurrence to improve object detection performance. Given regions of interest in an image, our method augments object detection networks with pooled contextual features from other regions of interest in the scene. We implement our scheme and evaluate it on the Pascal Visual Object Classes (VOC) 2007 and Microsoft Common Objects in Context (MS COCO) datasets. When used as part of the Faster R-CNN object detection framework with VGG-16, we show an increase in mAP from **24.2%** to **25.5%** over baseline Faster R-CNN and Global Average Pooling when testing on MS COCO.

***Index Terms*—** Object Detection, CNN, Pooling, Faster R-CNN, Object Co-occurrence, Context

1. INTRODUCTION

One of the key tasks in computer vision is object detection, which refers to localisation and classification of objects in a scene. While this task has been researched in the past [1], modern advances in deep convolutional neural networks (CNN) for image classification [2, 3, 4, 5] have brought significant performance gains to object detection over previous approaches[6, 7, 8, 9].

Faster R-CNN [8, 9] has emerged as a baseline model for CNN based object detection, combining both region proposal and region classification into a single network. However, a limitation of this framework is that each region proposal is classified individually without taking the rest of the image into account.

Various works have attempted to address this by adding contextual information from the rest of the image [10, 11]. These contextual features are known to benefit visual tasks, particularly when local features are insufficient such as in object recognition with small or obstructed objects [12, 13, 14].

1.1. Related Work

Global Average Pooling [10, 15, 11] has been used to add context to object detection by average pooling the entire source feature map then unpooling and concatenating it onto each localised object's feature map. This combined feature contains information about the object and its surrounding context, allowing subsequent layers of the network to leverage context from the rest of the scene.

Recent approaches utilise a spatial recurrent neural network to pass information laterally across the entire image feature map. Features pooled from this context map provide global context with respect to each pooled spatial location. This provides an increase in performance over global average pooling at the cost of computational power and time [10].

Li et al. [11] argue that not all information in an image is useful for context, and approaches such as Global Average Pooling provide low quality context features. Their work introduces using an attention-based recurrent neural network to find higher quality context features.

We hypothesise that using object co-occurrence for context will provide similar higher quality context when compared to context from the scene as a whole. Our approach explores this by introducing Region Average Pooling, which augments local object feature maps with a context feature pooled from all region of interests in an image. This method uses only important regions of interest for context, reducing background noise and providing superior context features.

1.2. Contribution

We introduce Region Average Pooling as a method of adding contextual information to object detection tasks. Through this method, object feature maps are augmented with pooled features from all regions of interest in an image. These contextual features allow deep CNNs to leverage object co-occurrence for context in order to improve object detection performance.

We implement our method and evaluate it on the Pascal VOC 2007 [16, 17] and Microsoft COCO [18] datasets. We compare our method against Global Average Pooling as a method of adding contextual information.

When used as part of the Faster R-CNN object detection framework with VGG-16 [9], we show an increase in mean average precision (mAP) from **24.2%** to **25.5%** over baseline Faster R-CNN and Global Average Pooling when testing on MS COCO test with minimal additional computational cost.

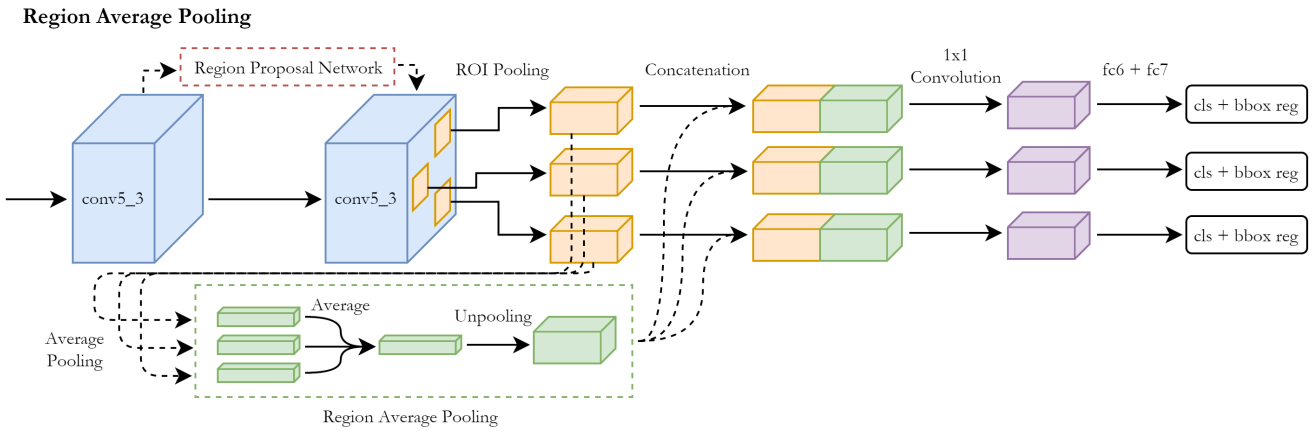

Fig. 1. Structure of Region Average Pooling (RAP) built on Faster R-CNN and VGG-16. In RAP, we compute the pooled average of all region of interests (ROI) before concatenating the result to each ROI. This concatenated feature map provides contexual information to subsequent layers of the network.

2. TECHNICAL APPROACH

We build our work on the framework provided by Faster R-CNN and VGG-16 [8, 9]. This framework provides region proposal through a Region Proposal Network (RPN), which is used to pool region of interest (ROI) feature maps from the final convolutional layer of VGG-16 (conv5_3). In the original framework, these ROI features are passed through the fully connected layers (fc6, fc7) before splitting into two separate fully connected layers for object classification and bounding box regression.

To add context to this framework, we compute a context feature and concatenate it to each ROI feature map. This combined feature goes through 1x1 convolution for dimensionality reduction before the fully connected layers. This provides the subsequent layers of the network with information about both the region of interest as well as additional contextual information.

To compute this context feature, we introduce Region Average Pooling and compare it with a similar scheme for adding context, Global Average Pooling.

2.1. Global Average Pooling

To compute a context feature using Global Average Pooling, the final convolutional feature map from VGG-16 (conv5_3) is average pooled to produce a single vector. This global vector is then unpooled by tiling the feature to match the size of the ROI feature maps.

This results in a context feature that contains information about the entire scene, which the following layers can use to augment individual object detection.

2.2. Region Average Pooling

In our Region Average Pooling scheme, each ROI feature map pooled from conv5_3 first undergoes average pooling to produce multiple vectors. These vectors are then averaged into a single vector and then unpooled by tiling into a context feature similar to that in Global Average Pooling.

The resulting context feature contains information about all region of interests in the scene which can be used to infer context via object co-occurrence. We believe that this will provide higher quality context features than that gained from looking at the global scene as a whole.

The structure of our scheme is shown in Figure 1.

3. EXPERIMENTS

We perform experiments using Faster R-CNN with VGG-16 as the base framework, and compare two methods of adding contextual features, Global Average Pooling and Region Average Pooling as described in Section 2. We build these models on the publicly available Python Caffe implementation of Faster R-CNN.

Both models are trained and evaluated on the PASCAL VOC and MS COCO datasets. For training on PASCAL VOC, we combine training and validation (trainval) sets of PASCAL VOC 2007 as well as PASCAL VOC 2012 [17, 19]. The training and validation sets of MS COCO [18] are similarly combined, but with a small subset of 5000 images left out for validation.

For quicker training, all models are initialised with weights from pre-trained Faster R-CNN models. For training on PASCAL VOC, the models' weights are initialised from a Faster R-CNN model pre-trained on PASCAL VOC 2007. For MS COCO, the models are similarly initialised

Method	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
FRCNN	73.2	76.5	79.0	70.9	**65.5**	52.1	83.1	84.7	86.4	52.0	81.9	65.7	**84.8**	84.6	77.5	76.7	38.8	73.6	**73.9**	83.0	72.6
GAP	**74.8**	77.6	**79.5**	**74.7**	65.4	57.9	83.6	87.2	**87.5**	**55.1**	81.6	66.1	83.4	85.5	**78.5**	78.7	47.0	**76.5**	70.6	**84.8**	74.2
RAP	**74.8**	**77.8**	79.1	73.9	63.9	**59.6**	**84.3**	**87.5**	87.2	**55.1**	**83.8**	**67.6**	**84.8**	**86.1**	77.4	**78.8**	**49.0**	75.2	69.1	82.2	**74.4**

Table 1. Comparing performance of our proposed Region Average Pooling (RAP) with Global Average Pooling (GAP) and baseline Faster R-CNN (FRCNN), on **PASCAL VOC 2007 test**.

	mAP, IoU			mAP, Area			mAR, Max Dets			mAR, Area		
Method	**0.50:0.95**	**0.50**	**0.75**	**Small**	**Medium**	**Large**	**1**	**10**	**100**	**Small**	**Medium**	**Large**
MS COCO 2015 test-dev												
FRCNN	24.2	45.3	23.5	7.7	26.4	37.1	23.8	34.0	34.6	12.0	38.5	54.4
GAP	24.2	44.9	23.9	7.5	26.4	36.7	24.0	35.0	35.7	**12.7**	**40.5**	54.0
RAP	**25.4**	**46.6**	**25.2**	**8.1**	**28.0**	**38.6**	**24.4**	**35.3**	**35.9**	12.6	40.4	**55.6**
MS COCO 2015 test-std												
FRCNN	24.2	45.3	23.4	7.2	26.4	36.9	23.8	34.1	34.7	11.5	38.9	54.4
RAP	**25.5**	**46.8**	**25.1**	**7.7**	**28.1**	**38.4**	**24.5**	**35.5**	**36.1**	**12.3**	**40.9**	**55.6**

Table 2. Comparing performance of our proposed Region Average Pooling (RAP) with Global Average Pooling (GAP) and baseline Faster R-CNN (FRCNN), on **MS COCO 2015 test-dev** and **MS COCO 2015 test-std**. We provide mean average precision (mAP) and mean average recall (mAR) over different intersection over union (IoU), area, and maximum number of detections.

using weights from a Faster R-CNN model pre-trained on MS COCO.

We train the models on PASCAL VOC using a learning rate of 0.001 for 50k iterations, 0.0001 for 100k iterations, and 0.00001 for a final 50k iterations. For MS COCO, we use a longer training schedule to allow the models to fully converge. We train at a learning rate of 0.001 for 250k iterations, 0.0001 for 300k iterations, and 0.00001 for a final 250k iterations. We set momentum to 0.9 and weight decay to 0.0005 for training on both datasets. Both models were trained end-to-end.

We evaluate the performance of both models on PASCAL VOC 2007 test, as well as MS COCO 2015 test-dev and test-std, and show results in Table 1 and Table 2.

3.1. Results

When testing on PASCAL VOC 2007 test, we find that while both Global Average Pooling and Region Average Pooling increase mean average precision (mAP) from **73.2%** to **74.8%** over baseline Faster R-CNN, neither method outperforms the other.

On MS COCO 2015 test-dev, while Global Average Pooling gives an increase in mean average recall (mAR), from **34.6%** to **35.7%** over baseline at 100 maximum detections, no increases in mean average precision (mAP) were observed.

Region Average Pooling on the other hand shows increases in both mAP as well as mAR, showing increases in mAP (IoU 0.5:0.95) from **24.2%** to **25.4%** and mAR (100 max detections) from **34.6%** to **35.9%** when compared to baseline Faster R-CNN.

Testing on MS COCO 2015 test-std shows a similar increase for Region Average Pooling to **25.5%** for mAP (IoU 0.5:0.95) and **36.1%** for mAR (100 max dets).

At test time, Region Average Pooling takes 0.150 seconds per image when compared to baseline Faster R-CNN at 0.145 seconds per image on an Nvidia Titan X GPU. This shows that Region Average Pooling displays increased performance while introducing minimal additional computational cost. This includes proposal generation, object classification, and bounding box regression.

3.2. Discussion

From our results, we observe that while Region Average Pooling and Global Average Pooling show improvements over baseline on both datasets, Region Average Pooling performs better than Global Average Pooling only when testing on MS COCO.

We believe that this difference in performance is due to the number of categories and object instances per image in each dataset. As stated in [18], MS COCO averages 3.5 categories and 7.7 object instances per image while PASCAL VOC averages less than 2 categores and less than 3 object instances per image. Furthermore, while only 20% of images in MS COCO have a single category per image, up to 70% of images in PASCAL VOC have only a single category per image. An example is shown in Figure 2.

As Region Average Pooling leverages context through object co-occurrence, we thus argue that it performs better on MS COCO when compared to PASCAL VOC due to the increased average number of categories and object instances, allowing more co-occurrence relationships to be learned by

the network.

One possible improvement to our current approach is to selectively pool ROIs that contain higher quality features. To implement this, the object scores produced by the Region Proposal Network (RPN) of Faster R-CNN may be used as an indicator of feature quality for each region. While this may provide improvements by using only high quality region features for context, this approach is beyond the scope of this paper and can be explored in further work.

4. CONCLUSION

We introduce Region Average Pooling as a method of adding contextual information to deep CNN based object detection networks such as Faster R-CNN. By pooling region of interests in a scene into a high quality context feature, we are able to augment local object feature maps and improve object detection performance by allowing the network to leverage context through object co-occurrence. We compare our method with Global Average Pooling and baseline Faster R-CNN on the Pascal Visual Object Classes (VOC) 2007 [16, 17] and Microsoft Common Objects in Context (MS COCO) [18] datasets, and find that Region Average Pooling outperforms Global Average Pooling as a method of adding contextual information to Faster R-CNN on the MS COCO dataset.

Fig. 2. Similar scenes from both PASCAL VOC (Top) and MS COCO (Bottom) with groundtruths shown. MS COCO averages a higher number of categories and object instances per image, allowing object co-occurrence to be better leveraged.

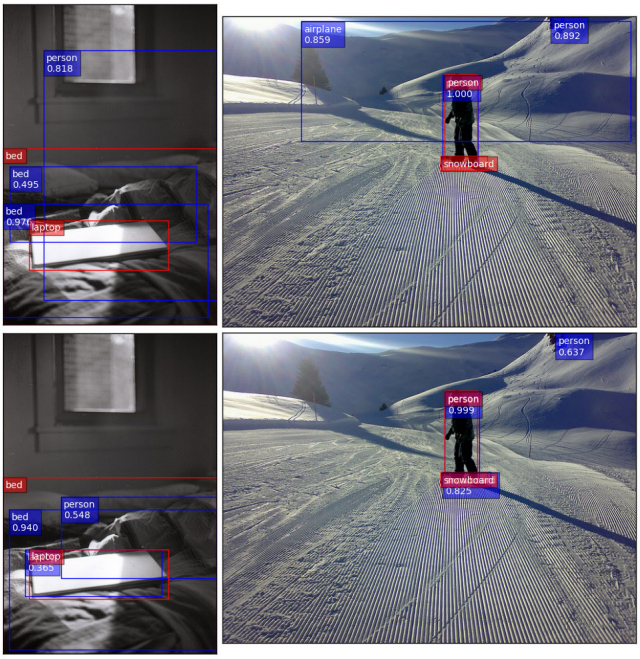

Fig. 3. Examples comparing results from Global Average Pooling (Top) and Region Average Pooling (Bottom) when testing on MS COCO. Groundtruths are shown in red and top 3 detections are shown in blue.

5. REFERENCES

[1] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[6] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[8] R. Girshick, "Fast r-cnn," in *International Conference on Computer Vision (ICCV)*, 2015.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[10] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2874–2883.

[11] J. Li, Y. Wei, X. Liang, J. Dong, T. Xu, J. Feng, and S. Yan, "Attentive contexts for object detection," *IEEE Transactions on Multimedia*, 2016.

[12] A. Oliva and A. Torralba, "The role of context in object recognition," *Trends in cognitive sciences*, vol. 11, no. 12, pp. 520–527, 2007.

[13] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert, "An empirical study of context in object detection," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1271–1278.

[14] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 891–898.

[15] W. Liu, A. Rabinovich, and A. C. Berg, "Parsenet: Looking wider to see better," *arXiv preprint arXiv:1506.04579*, 2015.

[16] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[17] ——, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results," http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*. Springer, 2014, pp. 740–755.

[19] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.