OBJECT DETECTION IN CURVED SPACE FOR 360-DEGREE CAMERA

Kuan-Hsun Wang, Shang-Hong Lai

Department of Computer Science, National Tsing Hua University, Taiwan

ABSTRACT

360° camera has recently become popular since it can capture the whole 360° scene. A large number of related applications have been springing up. In this paper, We propose a deep learning based object detector that can be applied directly on 360° images. The proposed detector is based on modifications of the faster RCNN model. Three modification schemes are proposed here, including (1) distortion data augmentation, (2) introducing muilti-kernel layers for improving accuracy for distorted object detection, and (3) adding position information into the model for learning spatial information. Additionally, we create two datasets, 360GoogleStreetView and 360Videos, and perform experiments on these two datasets to demonstrate that our object detector provides superior accuracy for object detection directly on 360° images.

Index Terms— Object detection, 360° image, panorama

1. INTRODUCTION

Recently, the amount of 360° data increases dramatically. Since it can capture objects in 360°, users can feel more immersive to the scene. Some applications based on using 360° cameras have become popular, such as augmented reality, virtual reality, and 360° video surveillance.

Using 360° images makes our detector capable of detecting object in 360° from all angles. However, there are only a few research on object detection for 360° images. For object detection on this kind of 360° images, the most common approach is to project the 360° view into multiple 2D tangent images which contain no distortion. After performing standard object detection on these 2D perspective images, an additional step is required to fuse the detection results on these images back to 360° image. Obviously, this approach takes additional computation time and power. The advantage of using the above approach is that there are abundant impressive object detection methods developed for the perspective images, especially the deep learning methods. Some pre-trained networks, such as AlexNet [1], GoogLeNet [2, 3], VGG [4], ResNet [5], and DenseNet [6] can directly apply on the perspective images and provide great results for object detection. But it will take additional computation on image warping from a 360° image to multiple perspective images.

In this paper, we aim to develop an object detector that can directly apply on 360° images without any image preprocessing and re-projection. However, to accomplish the method, the image distortion problem should be properly handled. As we can see from the example in Fig. 1, some object detection benchmark models, training on traditional perspective images, are unable to detect these distorted objects. To overcome the problem of distortion, we propose the following three methods: (1) distorted data augmentation—generating a variety of additional distorted data for training; (2) multi-kernel layer—applying different sizes of kernels on different regions to eliminate the distortion and (3) position information— the object position in the 360° image corresponds to different distortion in the image. In addition, we manually collect and label two datasets by ourselves to increase the number of training samples. The two datasets presented in this work are 360GoogleStreetView and 360Videos datasets.

Fig. 1. After equirectangular projection from a spherical image to the distorted 360° image, directly applying Faster RCNN [7] on this distorted image cannot accurately detect the distorted object, marked with yellow bounding box.

In Section 2, we discuss some related works for object detection on 360° images. In Section 3, we describe the proposed methods for improving the object detector. Some experimental results for the object detection on 360° images are given in Section 4. Finally, we conclude in Section 5.

2. RELATED WORKS

The previous object detection methods can be roughly divided into two categories: one-stage detection and two-stage detection. One-stage detection checks an input image for only one time to locate where the object is and classify which category the object belongs to. This kind of detectors detects faster but is less accurate, such as SSD [8], and YOLO

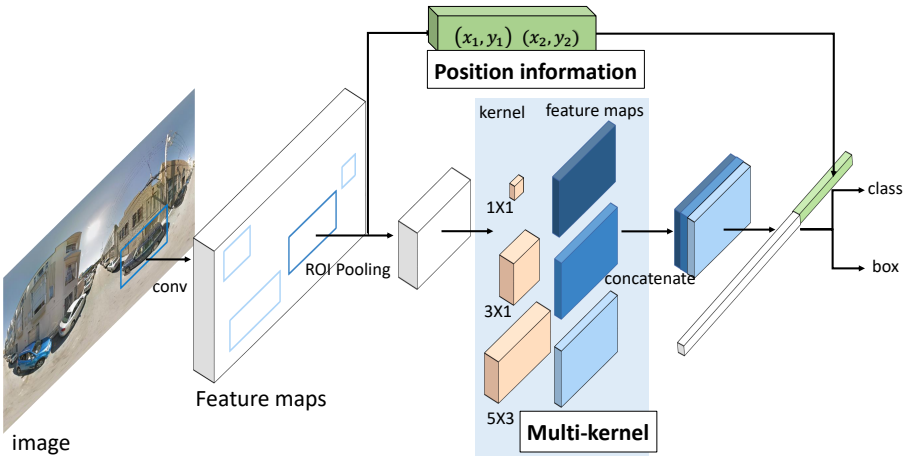

Fig. 2. Proposed network architecture.

[9]. The two-stage detectors also called region-based methods consist of two stages. In the first step, region proposals, for the input image are generated. In the second step, the possible region proposals are fed into the network to predict the final bounding boxes. Although this kind of methods takes more time, they provide higher accuracy, such as R-CNN [10], Fast-RCNN [11], and Faster-RCNN [7]. In addition, there are many follow-up improvements based on Faster-RCNN (*e.g.*,[12, 13, 14, 15, 16]). However, all of the above methods are trained on the traditional perspective images.

Panocontext [17] predicts 3D bounding boxes of the objects from input 360° images. By Using the information from object detection, they are able to reconstruct 3D room layout. To extract features directly from 360 images, Su *et al.* [18] use knowledge distillation to learn a spherical convolution network that teach a planar CNN to process on 360° images directly. Deng *et al.* [19] use three fisheye cameras to build a panoramic images dataset and train a region based CNN on their indoor 360° image dataset.

3. PROPOSED APPROACH

Compared to 2D perspective images, obviously 360° image is more complicated for object detection. We decide to adopt the 2-stage object detection framework due to its high accuracy of object detection for complex scene. Thus, we choose Faster R-CNN [7] as the baseline method. Fig. 2 is the overview of our network architecture. The input image is a distorted 360° image. After using convolutional layers to exeract feature maps, Region Proposol Network (RPN) is applied to predict some proposals with different scales and ratios. We apply a multi-kernel layer after cropping the feature maps through the ROI Pooling layer to alleviate the distortion problem. At the last fully connected layer, we even add the position information of the object region into the network. Finally, the last feature vector will be fed to two fully connected layers, one is for object classification and the other is for bounding box regression. The details for these proposed modifications will be described in the subsequent sections.

3.1. Distorted Data Augmentation

The lack of 360° images makes it difficult to train an object detector. In 360GoogleStreetView, we annotated 2,098 distorted objects which are much less compared to 61,428 normal objects. Thus, we propose a distorted data augmentation method to increase the number of distorted training data.

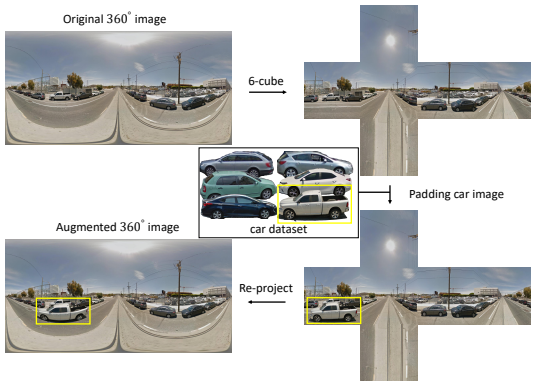

Fig. 3. We take a 360° image as input and then transform it to 6-cubic image. After imposing the car image to the bottom of image, we re-project it back to equirectangular image. The distorted car in the yellow bounding box is generated.

We first project the 360° image into 6-cube format. We can obtain some information that the objects in the bottom of cubic images have more significant distortion while passing through the projection. Using this characteristic, we can generate distorted object through the projection procedure. We impose the car images, which are cropped from the street

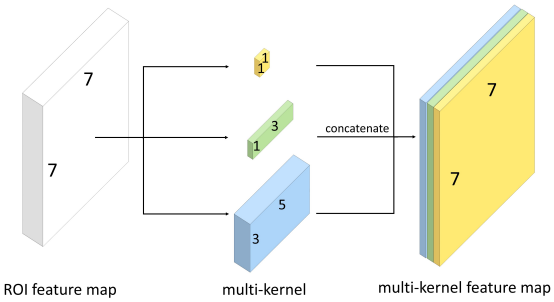

Fig. 4. Three different sizes of non-rigid kernels (1x1, 3x1, 5x3) are used in our network model after the ROI pooling to extract feature map from distorted objects.

view images and removed background manually, to the positions on the six-cube image which generate significant distortion on the 360° image. The position is randomly chosen but it should be located near the bottom of image. The size of imposed car image is also randomly picked since different size can cause different degree of distortion. Finally, re-projecting the 6-cubic image back to equirectangular images through equirectangular projection. The overlaid cars become distorted after this procedure. The complete procedure is shown in fig. 3.

3.2. Multi-kernel Layer

During the equirectangular projection, every pixel in the specific altitude will be projected to the fixed height in the image. However, the number of pixels at each latitude is not the same. As we observe the spatially-varying distortion phenomenon on 360° images, it may be helpful to use different sizes of feature extraction on different latitude. Since the objects located at the high latitude are stretched, the bigger kernel size should be applied to reduce the distortion. However, if changing all the square kernel to the non-rigid one, it is possible that the detection accuracy is reduced for general objects. Moreover, to cut down the computation, instead of applying to the whole image we add the multi-kernel layer after the ROI pooling layer. We apply 3 different sizes (1x1, 3x1, 5x3) of kernels for computing object feature maps, as illustarted in Fig. 4. Both 1x1 and 3x1 have 448 filters and 5x3 has 128 filters to maintain the performance on usual objects and distorted objects. After the feature extraction, we concatenate all feature maps for the final detection.

3.3. Position Information

In 360° image, the object may be significantly stretched when it is close to the top or the bottom of the 360° image. In addition to the location, the object size also plays an important role. Large objects contain more significant distortion. With the above observation, position and size of the object are important information for the detector to detect. Therefore, The position information is added to the feature vector to increase the feature knowledge after Region Proposal Network predicts the proposals. The position information is composed of 6 components, including bounding box coordinates and the width and height of the bounding box. The definition of position information is as follows:

$$P^i = \left(\frac{x_1^i}{W}, \frac{y_1^i}{H}, \frac{x_2^i}{W}, \frac{y_2^i}{H}, \frac{w^i}{W}, \frac{h^i}{H} \right) \quad (1)$$

where P^i represents position information for anchor box with index i. x_1 and y_1 denote the top left corner coordinate. x_2 and y_2 denote the bottom right corner coordinate. w^i and h^i denote the width and height of the anchor box i, respectively. W and H are the width and hight of the input image, and they are used to normalize the value.

3.4. Implementation Details

First, our network is initialized with a pre-trained model ResNet-101 which is pre-trained on VOC2007 and VOC2012 trainval dataset and then finetuned on the 360° image dataset. The model is trained with a learning rate of 0.001 which is decreased 10 after 50k iterations, a weight decay of 0.0005 and a momentum of 0.9. Each mini-batch has 1 image and each image has 256 regions of interest (RoIs), with a ratio of 1:3 of foreground to background. We train on a GPU (Nvidia Titan X) for 130k iterations.

4. EXPERIMENTAL RESULTS

4.1. Datasets

4.1.1. 360GoogleStreetView

We collect 360° images from Google Street View, some street view photos taken by Google. To increase the diversity of the dataset, we collect images from 6 different cities, including San Francisco, Moscow, Petersburg, Praha, Tokyo, and Taipei. A total of 2,095 images are collected and labeled, which contain 1,417 distorted people and 681 distorted cars.

4.1.2. 360Videos

The images are captured and annotated by ourselves. The camera that we use is LG360CAM. We select 3 different cities to collect the data. The data is originally recorded by video. After sampling some frames for labeling, the total number of 360° images is 494 in this dataset, which contain 2,257 distorted people and 191 distorted cars.

4.2. Evaluation metrics

Object detection average precision (mAP) is the fundamental metric to measure the accuracy of object detector. The definition is the same as that in [20].

Table 1. Detection results on 360GoogleStreetView testing set , including distorted object detection results. The results under different IoU (Intersection over Union) thresholds are given (0.5/0.7). D-car and D-person represents AP for distorted person and car, respectively. A_D represents accuracy for distorted object.

Method	mAP@0.5/0.7	D-car/D-person	A_D@0.5/0.7
[11]	72.88/56.24	88.06/79.94	87.01/75.87
[21]	69.15/52.28	69.97/79.12	89.69/80.00
Ours(P)	73.46/56.38	88.95/**80.53**	88.87/76.70
Ours(Mk)	**76.78**/56.67	89.45/80.41	89.48/80.00
Ours(Ag)	76.13/56.01	90.37/79.96	89.89/77.53
Ours(All)	76.38/**56.76**	**90.60**/80.01	**90.30/81.03**

Average Precision for distorted object (AP_D) only focuses on computing average precision for distorted objects. The definition of distorted object is that the object is located at the bottom of image or not.

Accuracy for distorted object is used to measure that the distorted objects are detected or not. The definition of distorted object is the same as that in the AP for distorted object.

4.3. Results and analysis

We perform experiments on the 360GoogleStreetView dataset. We compare our detector with the state-of-the-art object detectors and the result is reported in Table 1. All the methods, including Faster RCNN [11] and Deformable CNN [21], are trained on both 360GoogleStreetView and 360Videos.

Our object detector provides higher AP and accuracy for distorted object compared to Faster RCNN [11]. 2.54% higher on AP for distorted car and 3.29% higher on accuracy of distorted objects. Although deformable convolution can handle a little change on the object, the large distortion such as 360° distortion is hard for the model to handle. In addition, the proposed detector significantly outperforms these state-of-the-art detectors on mAP in both IoU thresholds of 0.5 and 0.7. As can be seen from the figure, AP for distorted person is just improved a little. The reason is that person is much smaller than car in these datasets and they only involve very slight distortion. Thus, if the model does not handle the distortion issue, it can still provide pretty good performance.

Additionally, we train different models (position, mutikernel, augmentation) to evaluate the different methods proposed in this paper. As can be seen in Table 1, all the three methods can improve the performance on mAP and AP for distorted cars, compared to Faster RCNN and Deformable CNN. Our combined model which integrates the three methods achieves the highest AP_D and $Accuracy_D$. However, it obtains the second-best mAP. We think the model may be a little bit overfitted if it considers too much distortion information. Some object detection results by using our detector are depicted in Figure 5.

Fig. 5. Some examples of object detection results on the 360GoogleStreetView test set by using the proposed method.

Table 2. Detection results on 360Videos testing set, including distorted object detection results. D refers to distortion.

Method	mAP_D@0.5/0.7	AP_{D-car}	$AP_{D-person}$
[11]	86.05 / 76.25	85.53 / 77.30	86.56 / **75.20**
[21]	87.76 / 75.30	89.85 / 76.45	85.67 / 74.15
Ours	**91.14 / 76.80**	**95.23 / 78.75**	**87.04** / 74.84

We also perform experiments on 360Videos dataset. 360Videos testing set has 103 images. The detection result is reported in Table 2, which contains the comparison between our detector and some state-of-the-art object detectors. The proposed detector shows great improvement on detecting distorted objects.

5. CONCLUSIONS

We propose a modified Faster RCNN model for object detection directly on 360° images. The distorted data augmentation method generates additional distorted objects for training. We propose to include a multi-kernel layer that incorporates different kernel sizes to alleviate distortion effect. In addition, we include object position information into the network to obtain better prediction. Furthermore, we created two datasets for performance evaluation of object detection on 360° images, and demonstrated the proposed object detector provided superior object detection accuracy compared to the state-of-the-art object detectors.

6. ACKNOWLEDGMENTS

The authors would like to thank Qualcomm Technologies Inc. for supporting this research work.

7. REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, et al., "Going deeper with convolutions," Cvpr, 2015.

[3] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[4] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[5] 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*, 2016, pp. 770–778.

[6] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, vol. 1, p. 3.

[7] 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*, 2015, pp. 91–99.

[8] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[9] 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*, 2016, pp. 779–788.

[10] 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*, 2014, pp. 580–587.

[11] Ross Girshick, "Fast r-cnn," in *Computer Vision (ICCV), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1440–1448.

[12] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick, "Training region-based object detectors with online hard example mining," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 761–769.

[13] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al., "Speed/accuracy trade-offs for modern convolutional object detectors," in *IEEE CVPR*, 2017.

[14] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017, vol. 1, p. 4.

[15] Xiaolong Wang, Abhinav Shrivastava, and Abhinav Gupta, "A-fast-rcnn: Hard positive generation via adversary for object detection," *arXiv preprint arXiv:1704.03414*, vol. 2, 2017.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, "Mask r-cnn," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2980–2988.

[17] Yinda Zhang, Shuran Song, Ping Tan, and Jianxiong Xiao, "Panocontext: A whole-room 3d context model for panoramic scene understanding," in *European Conference on Computer Vision*. Springer, 2014, pp. 668–686.

[18] Yu-Chuan Su and Kristen Grauman, "Learning spherical convolution for fast features from 360 imagery," in *Advances in Neural Information Processing Systems*, 2017, pp. 529–539.

[19] Fucheng Deng, Xiaorui Zhu, and Jiamin Ren, "Object detection on panoramic images based on deep learning," in *Control, Automation and Robotics (ICCAR), 2017 3rd International Conference on*. IEEE, 2017, pp. 375–380.

[20] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[21] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei, "Deformable convolutional networks," *CoRR, abs/1703.06211*, vol. 1, no. 2, pp. 3, 2017.