360-Indoor: Towards Learning Real-World Objects in 360° Indoor Equirectangular Images

Shih-Han Chou[1], Cheng Sun[1], Wen-Yen Chang[1], Wan-Ting Hsu[1,*], Min Sun[1], Jianlong Fu[2]
[1]National Tsing Hua University, Hsinchu[2] Microsoft Research, Beijing
shchou75@gmail.com, chengsun@gapp.nthu.edu.tw, {s0936100879, cindyemail0720}@gmail.com,
sunmin@ee.nthu.edu.tw, jianf@microsoft.com

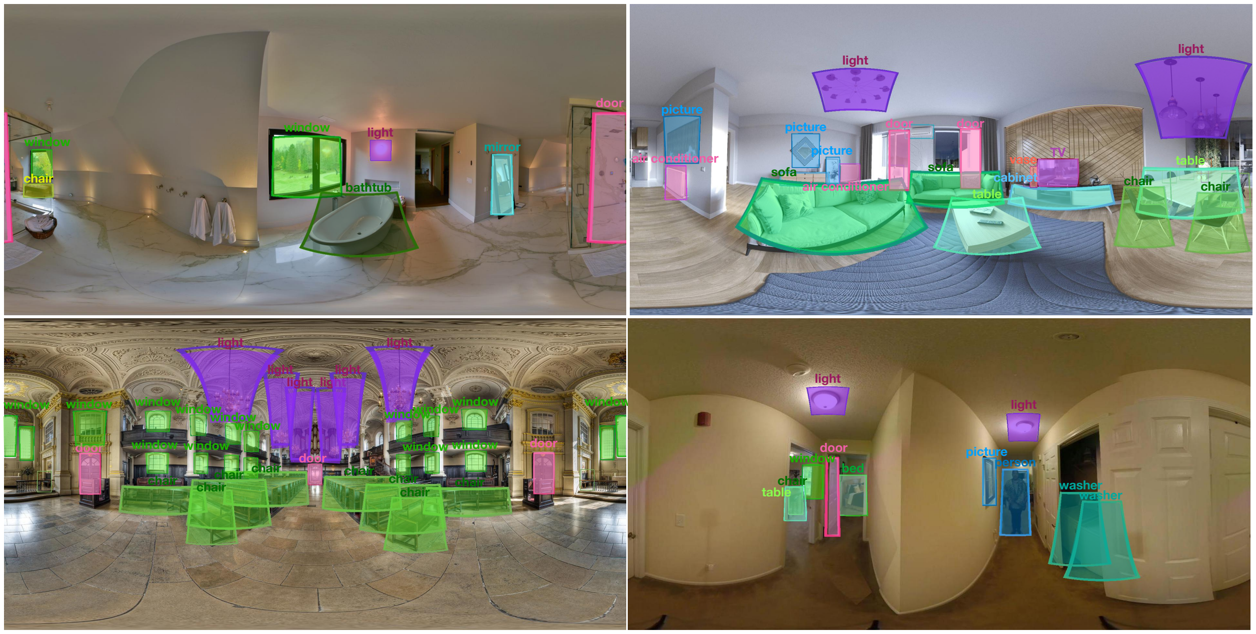

Figure 1: Examples of the images overlaid with labeled Bounding Field-of-Views (BFoVs) as well as object categories (text and color-code) in our 360-Indoor dataset.

Abstract

While there are several widely used object detection datasets, current computer vision algorithms are still limited in conventional images. Such images narrow our vision in a restricted region. On the other hand, 360° images provide a thorough sight. In this paper, our goal is to provide a standard dataset to facilitate the vision and machine learning communities in 360° domain. To facilitate the research, we present a real-world 360° panoramic object detection dataset, 360-Indoor, which is a new benchmark for visual object detection and class recognition in 360° indoor images. It is achieved by gathering images of complex indoor scenes containing common objects and the intensive annotated bounding field-of-view. In addition, 360-Indoor has several distinct properties: (1) the largest category number (37 labels in total). (2) the most complete annotations on average (27 bounding boxes per image). The selected 37 objects are all common in indoor scene. With around 3k images and 90k labels in total, 360-Indoor achieves the largest dataset for detection in 360° images. In the end, extensive experiments on the state-of-the-art methods for both classification and detection are provided. We will release this dataset in the near future.

1. Introduction

Object detection is an essential task in computer vision. The widely used datasets such as MSCOCO [17], Pascal VOC [10] have endorsed current researches make huge breakthroughs on object detection tasks [23, 19, 18, 13, 20, 21, 22]. Recently, 360° cameras become more popular and closer to our life because of the wide field of view and the applications to robots and virtual reality [15, 25]. Enormous 360° videos, such as house guiding, sports, are becoming

[*] This work was performed when Wan-Ting Hsu was visiting Microsoft Research as a research intern.

Table 1: Existing 360° dataset comparison in 2D domain. We list the released dataset to-date.

Dataset	**Type**	**Domain**	**Purpose**	**Annotation**	**#Category**	**#Boxes**
Pano2Vid [26]	Video	Outdoor Activities	Automatic Cinematography	-	-	-
Sports-360 [15]	Video	Sports	Visual Pilot	Viewing Angles		
YouTube/Vimeo [30]	Video	Wedding/Music	Highlight Detection	-	-	-
Narrated-360 [3]	Video	House/Tour Guiding	Visual Grounding	Bounding Boxes	-	-
Wild-360 [2]	Video	Nature/Wildlife	Saliency Detection	Saliency Map	-	-
SUN360[1] [28]	Image	Indoor/Outdoor	Scene/viewpoint recognition	Place Categories/Viewpoints	80	-
ERA [29]	Image	Dynamic Activities	Object Detection	Bounding FoVs	10	7,199
FlyingCars [5]	Image	Synthesis Cars	Object Detection	Bounding FoVs	1	6,000
360-Indoor	Image	Indoor Objects	Object Detection	Bounding FoVs	**37**	**89,148**

viral on YouTube. With the growing amount of data, there is an increasing interest in computer vision to dig into 360° visual recognition. Among numerous 360° projection (i.e., cubemaps, equirectangular and equiangular cubemaps), the most popular representation of 360° images is the equirectangular projection. It maps the latitude and longitude of the spherical to horizontal and vertical grid coordinates.

However, significant distortions of equirectangular images is a crucial problem, especially in the polar regions. Although many works propose spherical convolutional neural networks, such as [4, 24, 5], and dedicate to solve the distortion issue, there still lacks a suitable dataset to evaluate their approaches on object detection domain. Both in [4, 5], they experiment the proposed spherical convolutional neural networks on MNIST dataset to perform the classification task. In [24], they project PASCAL [10] to 360° format and do the object detection. In addition, in [5], they evaluate the proposed SphereNet on FlyingCars dataset, which is a synthesis dataset combines the real-world background equirectangular images with rendered 3D car models. We evaluate [5] the proposed 360-Indoor dataset in Section 5.

Motivated by the above observation, we present the 360-Indoor dataset in this paper. 360-Indoor is the first released and the largest object detection and classification dataset up to now. It consists of 3k equirectangular indoor images and 90k Bounding FoVs (BFoVs) annotations among 37 categories in current version. 360-Indoor benchmark is characterized by the following major properties.

- 360-Indoor is the first released and the largest object detection dataset in 360° domain, where each image is annotated with 27 BFoVs on average. The amount of BFoVs provides sufficient data for training and evaluating.

- 360-Indoor contains the most diverse categories, which include 37 categories. This will benefit the validation of the generalization capability of any approach.

- 360-Indoor is built in real images. It plays an important role since we can easily adapt to real-world applications. Figure 1 shows some examples of the images and their annotated BFoVs.

For experiments, we benchmark several deep neural network models performing object detection and classification. The best-performing system, FPN, can achieve 33.6% mean average precision (mAP). In the end, by observing the overall performance of existed methods, we believe 360-Indoor has not yet saturated and still have room to be improved.

Our contributions in this paper are two-fold:

- We collect the first object detection and classification dataset on 360° domain which contains 3k equirectangular indoor images and 90k BFoVs annotations among 37 categories.

- We comprehensively evaluate three different object detection models on the proposed 360-Indoor dataset. The results show that standard object detection methods train on the proposed dataset do have large improvements than using NFoVs

2. Related Work

Datasets in the computer vision research domain play a critical role. They not only provide a means to train and evaluate algorithms, but they also help researchers to explore new and more challenging directions. Nowadays, the ImageNet dataset [8] makes breakthroughs in both object classification and detection research. PASCAL, MSCOCO dataset [10, 17] with thoroughly annotations provide more complex object recognition tasks to be developed. Recently, with the growing attention on 360° domain, several 360° datasets have been proposed. We address this as follows.

Existed 360° Dataset. 360° visual is a thriving topic nowadays due to the advance of technology in 360° cameras. In recent two years, several datasets in 360° come up. In [26], they propose the first 360° video dataset which contains several outdoor activities. Similar to [26], Hu et al. [15] collect a sports 360° dataset which covers sports videos and

[1] The 80 categories of SUN360 is for image (scene) classification, without instance-level bounding boxes.

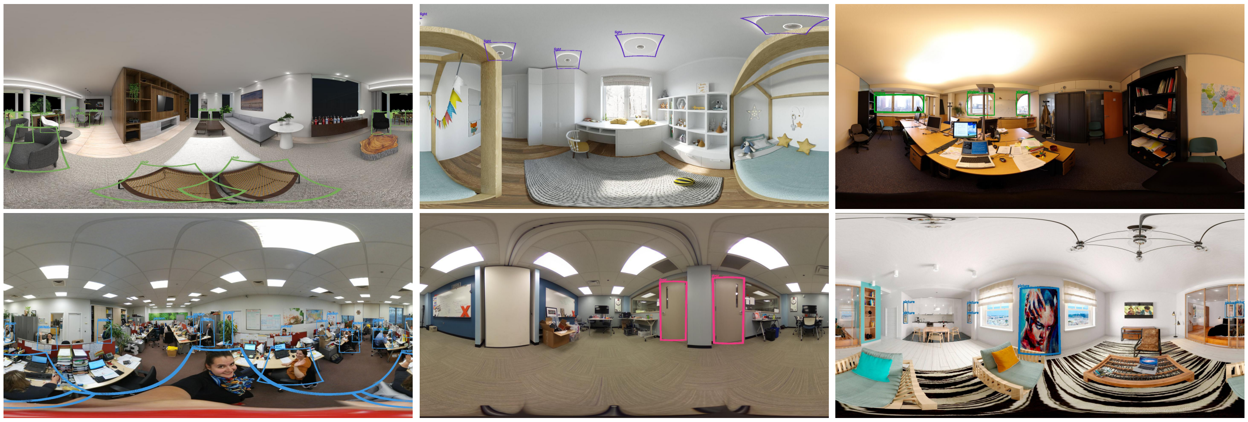

Figure 2: Top six categories in 360-Indoor including chair, light, window (from left to right in the top row), person, door and picture (from left to right in the bottom row). The polygons with different colors indicate different categories. The annotations using BFoVs can better fit the objects in 360° images. [Best viewed by zooming in.]

the annotated fixed Normal Field-of-View (NFoV) in each frame. Furthermore, a highlight detection dataset is proposed in [30]. They crawl the videos from Youtube and Vimeo using keywords 'wedding' and 'music'. In order to match the narrative and content in 360° videos, Chou et al. [3] provides a narrated 360° dataset. In this dataset, they annotate the objects mentioned in narratives on panoramas chronologically according to the start and end time. However, narrated 360° dataset only annotates in validation and testing set. Furthermore, Wild-360 [2] provides the saliency maps to facilitate machine to help users watch the 360° videos. SUN360 dataset [28] focuses on scene and viewpoint recognition and has scene/viewpoint labels which are different from the proposed dataset, 360-Indoor. Recently, Yang et al. [29] collect a dynamics activities dataset and Coors et al. [5] collect a flying cars dataset. However, the size of the dynamics activities dataset is not large enough. Besides, the cars in flying cars dataset are synthesized and added to the images which are hard to apply to real-world directly. As a result, we propose 360-Indoor which is the first release object detection and class classification in 360° domain. We also take the distortion in 360° images into consideration. The annotation format is tailored for equirectangular images. Table 1 lists the existed 360° dataset for comparison.

3. 360-Indoor Dataset

360-Indoor dataset is collected with $3,335$ indoor images and $89,148$ annotated BFoVs. In the following, we first introduce the Dataset sections and address each in turn. In Section 3.1, the procedure for object category selection is provided. Meanwhile, in Section 3.2, we will describe how to collect the candidate images. Next, we introduce the novel tool for annotating the 360° images in Section 3.3. In the end, we provide the statistics about the 360° Indoor Detection Dataset (360-Indoor) in Section 3.4.

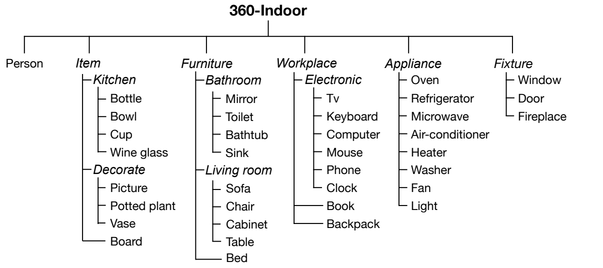

Figure 3: Categories in 360-Indoor dataset. The italic font denotes the super-categories, and the standardized font denotes the 37 object categories.

3.1. Category Selection

For categories selection, we consider the original categories defined in COCO dataset [17]. However, since some categories are hard to be seen in the 360° images, we manually remove these categories having small sizes. For example, we remove tableware such as 'spoon' and 'fork' as well as the categories in the food super category. In addition, we unify the categories according to the similar properties. For example, we merge 'lamp' and 'light' into 'light'. Furthermore, we add some categories which are common in the indoor scenes, such as 'washer', 'heater', 'cabinet', etc. Next, we group the object categories into 5 super-categories, except for 'person'. Each super-category represents a kind of purpose. Since 'person' does not belong to any super-categories, we separate it to an independent one. Figure 3 shows the 37 categories selected for annotation and the super categories in the 360-Indoor dataset.

3.2. Image Collection

The images used for 360-Indoor dataset are collected from Flickr[2], Kuula[3] and Narrated 360° videos dataset [3].

[2] https://www.flickr.com/

[3] https://kuula.co/

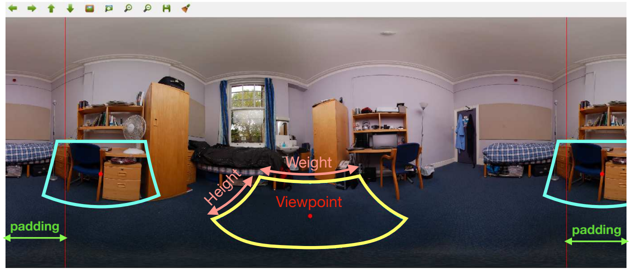

Figure 4: The annotation tool for annotating 360-Indoor. Each BFoV is represented by the center viewpoint and the width and height of the field-of-view. Annotators first select the category and click the center viewpoint. Then, they use the up/down buttons to enlarge/reduce the height and left/right buttons to enlarge/reduce the width. Note that we extend the image boundary using circular padding to take care of objects at image boundary.

For images from Flickr, we use related keywords such as '360degrees' and 'Equirectangular' to find 360° images. While Kuula is a 360°-image sharing platform, so we collect as many images as we can from this website. However, some of the images retrieved from Flickr and Kuula are duplicated or non-real. Hence, we first leverage a duplicate finder to remove the duplicated images, and then manually remove the non-real images. For the Narrated 360° videos dataset, we take only one frame in each video to avoid redundant scenes. After collecting the 360° images from these three resources, we manually select the indoor images and split the images into four different scenes: 'activity', 'home', 'shop', and 'work'. To balance the categories in the proposed dataset, we remove/add some images in the major/minor scenes. More specifically, scenes 'work' and 'home' are major in our dataset at first. We remove some images in 'work' and 'home' scene and add more images in 'shop' and 'activity'. In the end, we have $3,335$ images in total. All images are with $960 \times 1,920$ resolution.

3.3. Image Annotation

Objects in equirectangular images appear distorted depend on its spatial location, especially in the polar region (the top and bottom region in Figure 1). Therefore, conventional bounding boxes is no longer suitable for labeling 360° images. Hence, we choose bounding field of views (BFoVs) presented in [29] as annotations in our 360-Indoor dataset. Unlike the conventional bounding boxes represent by top-left and bottom-right corners (x_{min}, y_{min}, x_{max}, y_{max}), BFoVs is defined by (ϕ, θ, h, w) (Figure 4). ϕ and θ are latitude/longitude coordinates of the object's tangent plane and h is the object height w is the object width. To facilitate annotators labeling 360° images, we design an annotation tool which can select the viewpoints and adjust the height and width. The annotation tool is shown in Figure 4. Annotators are asked to choose the center of the object as the viewpoint (red points in Figure 4) and using the up/down buttons to adjust the enlarge/reduce height and left/right buttons to enlarge/reduce width. The BFoV will simultaneously show on the image and annotators can easily adjust the BFoVs to match the shape of objects in 360° images. In addition, since the boundary of the 360° images continues, the BFoV might across the right and left boundary (i.e., the blue BFoV shown in Figure 4). For the convenience of the annotators, we pad 45° field-of-view region to the left and right side. If the BFoV is across the padding area, the BFoV will show at the other side simultaneously. In order to maintain the unity and coherence of the dataset, each category is labeled by one expert annotator.

Table 2: Summary of 360-Indoor dataset.

Split	**#images**	**#BFoV**	**Avg.**	**Max.**	**Min.**
Train	2,325	62,430	26.9	211	1
Test	1,010	26,718	26.5	223	1

Table 3: Viewpoints distribution of 360-Indoor dataset.

#BFoVs	0 to ±30°	±30° to ±60°	±60° to ±90°
Train	49,340	11,561	1,529
Test	21,582	4,519	617

3.4. Dataset Statistics

Overall, our 360-Indoor consists of a total of $3,335$ images. We split the dataset into training/testing set with 70% and 30%, respectively. We summarize the statistics of the 360-Indoor dataset in Table 2 and show the distribution of the top 10 categories in Figure 6. In addition, we also provide the statistics of the distribution of viewpoints, height, and width of 360-Indoor dataset. The results are shown in Table 3 and Figure 5, respectively. For viewpoints distribution, we first separate latitudes into $\phi \in \Phi = \{0, \pm 30, \pm 60, \pm 90\}$ according to the distortion of objects. Between 0 to ±30°, object distortion is less. The distortion will become severe when ϕ become bigger. In latitude from ±60° to ±90°, objects will have the largest distortion which has the most significant difference from conventional images. As illustrated in Figure 3, most of the objects appear in latitudes from 0 to ±30°. This is a common scenario since objects appear in the middle of the indoor scene. In addition, in high latitudes region (±60° to ±90°), 360-Indoor dataset has sufficient portion objects which are able to help machine to recognize. For height and width distribution, they are alike in training and testing set. The mode of height and width in training and testing set

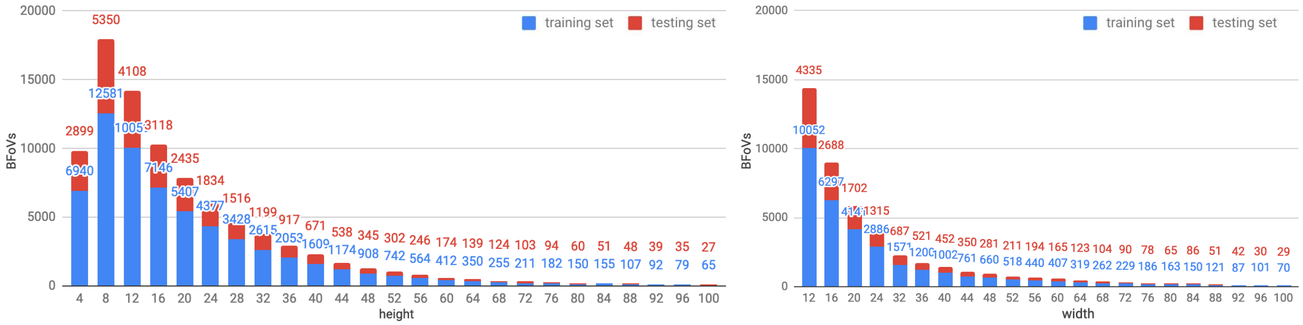

Figure 5: Distribution of height and width in 360-Indoor.

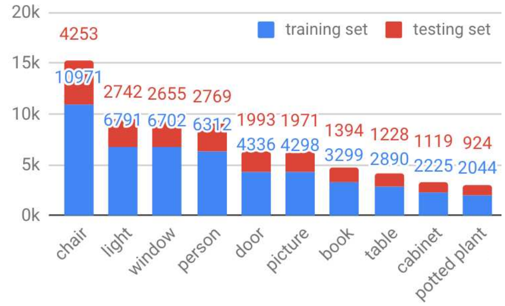

Figure 6: Distribution of top 10 categories in 360-Indoor.

are both 8 angular dimensions. Please note that the height and width do not correspond to the height and width in conventional bounding boxes since the regions in the equirectangular projection are not rectangular. For example, in the polar region, although the height and width are small, the projected region in the equirectangular image will cover a large portion.

4. Approaches to Object Detection

We briefly describe different object detection approaches that we benchmark on our proposed 360-Indoor dataset. Most of the state-of-the-art approaches for object detection are based on the Convolutional Neural Networks (CNNs) which can capture spatial correlation by the filter with patterns. As this dimension of research achieves better performance than the traditional hand-craft feature (e.g., the histogram of oriented gradients, Bag-of-visual Words model), we summarize CNN-based approaches as two kinds of methods, one-stage object detection, and two-stage object detection. In one-stage object detection approach [18, 20, 21, 22] (as shown in Figure 7 (a)), given an input image, the model utilizes CNNs to extract the visual representation for objects and directly predict bounding box information, foreground/background confidence and class confidence in every grid cell. The model first divides the image feature into grids. Each grid cell predicts B bounding boxes and confidence scores for those boxes. The confidence scores are provided based on how confident the model regards the box contains an object and also how accurate. Each grid cell also predicts C conditional class probabilities. These probabilities are conditioned on the grid cell containing an object. In testing phase, class-specific confidence scores can be derived by multiplying the conditional class probabilities and the individual box confidence predictions. These scores encode both the probability of that class appearing in the box and how well the predicted box fits the object. In sum, the one-stage model will optimize the localization task and classification task at the same time by framing object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities.

On the other hand, in the two-stage object detection approach [12, 27, 11, 6, 13, 23, 7, 16], given an input image, the model will use CNNs to extract feature and predict the bounding box with foreground confidence score at first. Then, using a detector to aggregate feature map and proposed regions, the model is able to predict bounding boxes and give class probabilities (as shown in Figure 7 (b)). The model first uses a deep fully convolutional network to propose regions in stage one (the yellow part in Figure 7 (b)). To generate region proposals, sliding windows are used over the feature map. At each sliding-window location, an anchor is centered and is associated with a scale and aspect ratio. Leveraging non-maximum suppression, candidate region proposals can be derived. In stage two, a region-based CNN (R-CNN) detector uses the proposed regions (the green part in Figure 7 (b)). The detector takes the feature map and proposed regions as inputs to predict bounding boxes and give class probabilities. The entire system is a single, unified network for object detection. Using the terminology of neural networks with attention mechanisms, the region proposal module tells the R-CNN module where to look. There are multiple ways to get more general features. The first one is conventional extract feature [23]. The second one is using multi-level semantic information from different layers [16].

Among all CNN-based state-of-the-art methods for ob-

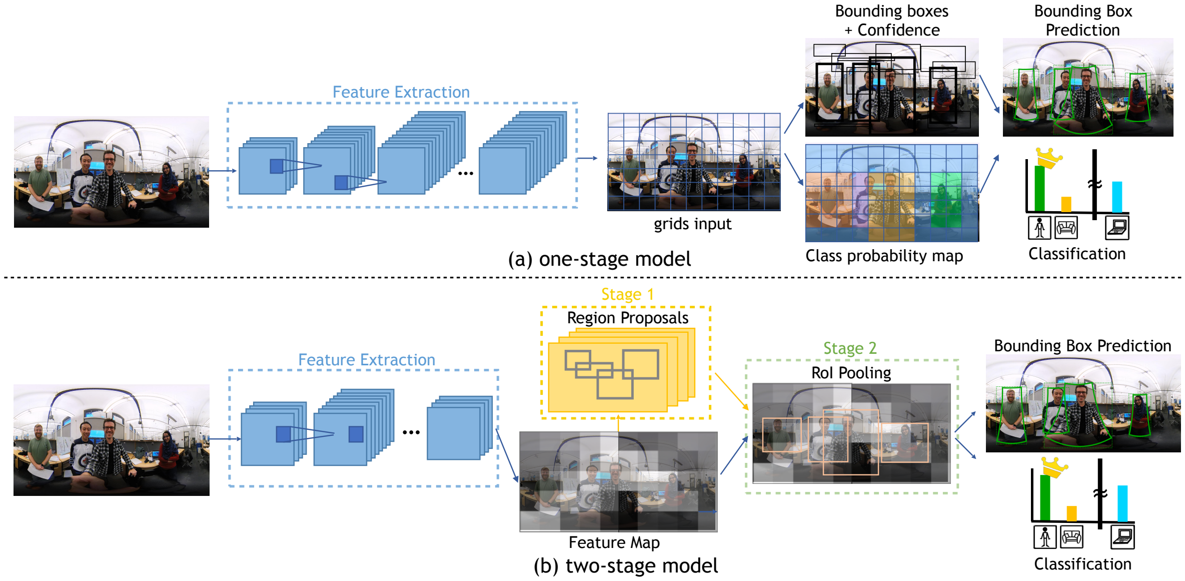

Figure 7: Approach. Summarization of two different object detection approaches. (a) shows the framework of one stage detection. The model utilizes CNNs to extract the visual representation for objects and directly predict bounding box information, foreground/background confidence and class confidence. (b) shows the framework of two-stage detection. After deriving feature map, the model predicts the bounding box with foreground confidence score at first. Then, using a detector to aggregate feature map and proposed regions, the model is able to predict bounding boxes and give class probabilities.

ject detection, we mainly investigate and evaluate in terms of three directions on our 360-Indoor dataset: one one-stage object detection, YOLOv3 [22], and two two-stage object detection, a conventional feature extraction, Faster R-CNN [23] and multi-level feature extraction, FPN [16].

In addition, there are several spherical CNNs dedicate to eliminate the distortion problem in the equirectangular images, such as [24, 1, 4, 5, 9]. We choose SphereNet [5] as its ease of integration into an object detection model. In one-stage and two-stage object detection, we replace the `Conv2d` and `MaxPool2d` in the feature extraction to `SphereConv2d` and `SphereMaxPool2d` [5]. The rest of the object detection model remains the same.

5. Experiments

We conduct three widely used object detection approaches on 360-Indoor dataset. The approach can be separated into two types: (1) one-stage object detection approach (e.g., YOLOv3 [22]) and (2) two-stage object detection approach (e.g., Faster R-CNN [23] and FPN [16]). In addition, the conventional CNNs also replace by the SphereNet in order to learn the invariance of these distortions.

Bounding Field-of-View Transform To match the input formats of the above-mentioned approaches, we first transform the bounding FoVs to conventional bounding boxes. The vertexes of the conventional bounding boxes can be derived by projecting ground truth annotations to the tangent plane. That is, we use a mapping function from [3] to map the annotations in spatial coordinates to tangent coordinates. The mapping function takes viewpoints, width, and height as input, and outputs the corresponding pixels in tangent coordinates. Hence, the projected pixels can be derived. After having the projected pixels, we choose the boundary of these pixels to draw the transformed bounding boxes. Therefore, the $(x_{min}, y_{min}, x_{max}, y_{max})$ in the conventional bounding box can be derived and fed into the object detection networks.

Faster R-CNN. We use the official settings to implement Faster R-CNN. The anchor scales are set as $\{64^2, 128^2, 256^2, 512^2\}$ with aspect ratio $\{\frac{1}{2}, 1, 2\}$ which are the same as the setting in COCO dataset. We use ResNet101 [14] pretrained on ImageNet as our backbone as it gives us better mAP. We train Faster-RCNN with 1 GPU with batch size 1, $12,000$ RoIs per image before applying NMS to RPN proposals and 2000 after applying NMS to RPN proposals. We use SGD with momentum 0.9 and weight decay 10^{-4}. The learning rate is set to 0.001.

FPN. We follow the official implementation. The anchor scales are set as $\{32^2, 64^2, 128^2, 256^2, 512^2\}$ with aspect ratio $\{\frac{1}{2}, 1, 2\}$. We also use the same method to map RoI to pyramid level. Instead of using RoI pooling, we use RoI align [13] to extract the proposed candidates for second-stage. We use ResNet101 [14] pretrained on ImageNet as our backbone as it gives us better mAP. We train FPN with 1 GPU with batch size 2, 512 RoIs per image to train first-

Table 4: Experiment on 360-Indoor dataset. We compare with the baseline which is pretrained on COCO as well as different backbone networks. mAP (overlap) denotes we only use the overlap categories between COCO and 360-Indoor to calculate mAP. mAP denotes using the result from all categories in 360-Indoor dataset.

Model	Backbone	Anchors	mAP (overlap) (%)	mAP (%)
YOLOv3	baseline (DarkNet53 trained on COCO)	kmeans++ for COCO (9) [22]	10.9	-
	ResNet50 trained on 360-Indoor	kmeans++ for COCO (9) [22]	11.9	12.4
	DarkNet53 trained on 360-Indoor	kmeans++ for COCO (9) [22]	**22.7**	**24.5**
Faster R-CNN	baseline (ResNet101 trained on COCO)	default [23]	11.5	-
	ResNet50 trained on 360-Indoor	default [23]	24.5	29.1
	ResNet101 trained on 360-Indoor	default [23]	**25.3**	**30.2**
FPN	baseline (ResNet101 trained on COCO)	default [16]	12.2	-
	ResNet50 trained on 360-Indoor	default [16]	27.4	33.1
	ResNet101 trained on 360-Indoor	default [16]	**28.2**	**33.6**

stage and $2,000$ RoIs per FPN level to train second-stage. We use SGD with momentum 0.9 and weight decay 10^{-4}. The learning rate is set to 0.0025 because our batch size is 8 time smaller than official. The running statistic of Batch-Norm is fixed following standard practice for small batch size. The images' resolution is all resized to 960×960 as it shows good accuracy and computation resource trade-off.

YOLOv3. We leverage the following training strategy proposed by the J. Redmon *et al*. [22]: multi-scale training, data augmentation and batch normalization. We choose the Darknet-53 pretrained on ImageNet as our backbone, which performance approaches the ResNet-101 and 1.5× faster (as mentioned in [22]). However, we notice that the original preprocess of YOLOv3 is not suitable for the high respect ratio image. Because images will be padded to a square shape and resize to 416×416, the final predictions will have low precision due to lack of feature. Hence, we directly resize 360° images with the original size $960 \times 1,920$ to 960×960. We use SGD with momentum 0.9 and weight decay 10^{-3}. The learning rate is set as $5 * 10^{-4}$ which is the same as in [22].

Baseline. To demonstrate the effectiveness of the proposed 360-Indoor dataset, we further consider the following models. We take the COCO dataset pretrained model and directly use the testing set of 360-Indoor dataset to evaluate. For every approach, we use ResNet101 [14] as the backbone network and use the same anchor box settings from their reference.

5.1. Discussion

The results are shown in Table 4 indicate that all three detectors trained on our 360-Indoor dataset significantly outperform detectors trained on COCO dataset. We utilize mean average precision (mAP) as the evaluation method. We first use the pretrained detectors and directly test on the proposed 360-Indoor testing set, which refers to the first row in each model (baseline). In addition, mAP (Overlap) denotes that we only use the overlap categories between COCO and 360-Indoor to calculate mAP. Comparing with the detectors fine-tune on 360-Indoor dataset (third row in each model), it is critical to train detectors on our 360-Indoor dataset to achieve high object detection accuracy on equirectangular images. Compare with three approaches, FPN achieves the best performance which indicates that it has a better ability to deal with the distorted images. Among the three detectors, YOLOv3 achieves slightly worse accuracy. Since YOLOv3 is sensitive to the anchor proposals, we further conduct an analysis of the anchor proposals.

Table 5: Analysis of anchor proposals in YOLOv3. We show the comparison of different anchor proposal setting.

Procedure	Anchor Proposals	mAP (%)
COCO	(10×13),(16×30), (33×23), (30×61),(62×45), (59×119), (116×90),(156×198), (373×326)	24.5
2.3×	(23×30),(37×69), (76×53), (69×141),(143×104), (136×275), (268×208),(358×455), (858×750)	22.4
360-Indoor	(17×21),(11×60), (31×58), (38×128),(101×113), (65×242), (165×249),(327×313), (959×201)	**27.2**

Analysis of anchor proposals in YOLOv3. We evaluate three kinds of anchor proposals with YOLOv3. Firstly, we use anchor boxes size calculated from COCO datasets, which is the original setting in YOLOv3. Secondly, we modify the anchor boxes size to be 2.3× bigger than the anchor boxes calculated from COCO datasets since our input image size is 2.3× large than the original setting. Finally, we directly calculate 9 anchor boxes size from 360-Indoor dataset using kmeans++ as suggested in [22].

The results in Table 5 show that it is critical to calculating anchor boxes size from our 360-Indoor dataset. We also found that YOLOv3 takes longer to converge during

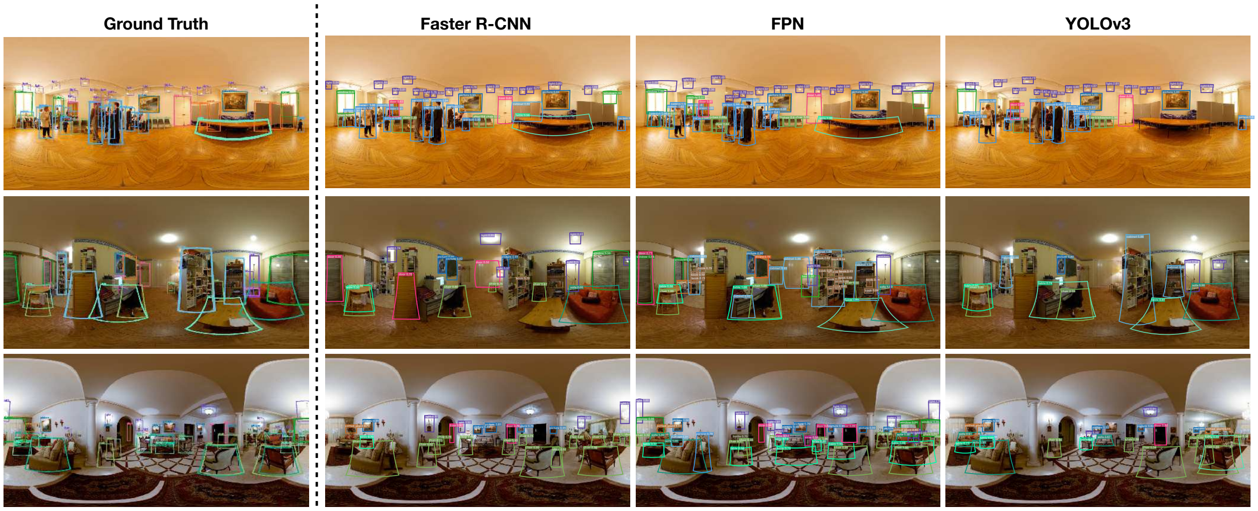

Figure 8: Qualitative results. We show three examples to illustrate the results of object detection from different approaches and ground truth. FPN is able to detect the objects with distortion and small objects. Hence, it achieves the best performance among the three approaches.

training when anchors boxes are not calculated from our 360-Indoor dataset. Our 360-Indoor dataset is again the key contributor to the performance improvement for YOLOv3.

Object detection model with SphereNet. To better encodes invariance against such distortions explicitly into convolutional neural networks, we replace the `Conv2d` and `MaxPool2d` in the conventional CNNs to `SphereConv2d` and `SphereMaxPool2d`, respectively. The results are shown in Table 6. Comparing with row one and three, `Conv2d` (pretrained on ImageNet) and `MaxPool2d` is better than `SphereConv2d`/`SphereMaxPool2d`. We think the reason is that `SphereConv2d` is trained from scratch, the model needs more time to converge. This indicates the importance of pretrained model. For a fair comparison, we train `Conv2d` from scratch (second row in Table 6) and use `MaxPool2d`. We can notice that `Conv2d` (w/o pretrain)/`MaxPool2d` is slightly better than `SphereConv2d`/`SphereMaxPool2d`. We argue that because conventional object detectors are designed for the normal images, the compatibility of the conventional CNNs and object detectors is better than SphereNet. Hence, the object detectors specialize for 360° images is needed. As a result, we believe the proposed 360-Indoor dataset provides a good start point for 360° domain in future studies. 360-Indoor is large enough to train detectors and benefit validation of the generalization capability of any approach.

Qualitative results. The qualitative results are shown in Figure 8. To better compare with the ground truth, we project rectangular boxes to spherical plane. As illustrated in Figure 8, FPN can detect the objects more correctly, and even the objects are with more distortion or small.

Table 6: Performance of object detection models with and without SphereNet. We show the comparison with different model settings.

Procedure	Settings	mAP (%)
Faster R-CNN	`Conv2d` (pretrain)/ `MaxPool2d`	**30.2**
Faster R-CNN	`Conv2d` (w/o pretrain)/`MaxPool2d`	24.1
Faster R-CNN	`SphereConv2d`/`SphereMaxPool2d`	21.7

6. Conclusion and Future Work

We present a real-world 360° panoramic object detection dataset, 360-Indoor, which is a new benchmark for visual object detection and class recognition in 360° images. It consists of complex indoor images containing common objects and the intensive annotated bounding field-of-view. To the best of our knowledge, it is the largest one in category numbers and number of bounding boxes. Extensive experiments on the detection methods show that training using our 360-Indoor dataset is the key to achieve state-of-the-art accuracy on 360° images. Thus, our 360-Indoor dataset can contribute to future development on applications (e.g., robot perception and virtual reality) requiring detecting objects in 360° images. In the future, we aim at developing dedicated object detectors overcoming image distortion and leveraging context from the complete field-of-view in a scene.

Acknowledgments. We thank MOST 108-2634-F-007-006 Joint Research Center for AI Technology and All Vista Healthcare, Taiwan for their supports.

References

[1] W. Boomsma and J. Frellsen. Spherical convolutions and their application in molecular modelling. In *NIPS*, 2017. 6

[2] H.-T. Cheng, C.-H. Chao, J.-D. Dong, H.-K. Wen, T.-L. Liu, and M. Sun. Cube padding for weakly-supervised saliency prediction in 360° videos. In *CVPR*, 2018. 2, 3

[3] S.-H. Chou, Y.-C. Chen, K.-H. Zeng, H.-N. Hu, J. Fu, and M. Sun. Self-view grounding given a narrated 360 video. In *AAAI*, 2018. 2, 3, 6

[4] T. S. Cohen, M. Geiger, J. Köhler, and M. Welling. Spherical cnns. *arXiv*, 2018. 2, 6

[5] B. Coors, A. P. Condurache, and A. Geiger. Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In *ECCV*, 2018. 2, 3, 6

[6] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016. 5

[7] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NIPS*, 2016. 5

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2

[9] C. Esteves, C. Allen-Blanchette, A. Makadia, and K. Daniilidis. Learning so (3) equivariant representations with spherical cnns. In *ECCV*, 2018. 6

[10] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *ICCV*, 2015. 1, 2

[11] R. Girshick. Fast r-cnn. In *ICCV*, 2015. 5

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 5

[13] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017. 1, 5, 6

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6, 7

[15] H.-N. Hu, Y.-C. Lin, M.-Y. Liu, H.-T. Cheng, Y.-J. Chang, and M. Sun. Deep 360 pilot: Learning a deep agent for piloting through 360° sports videos. In *CVPR*, 2017. 1, 2

[16] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 5, 6, 7

[17] 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 *ECCV*, 2014. 1, 2, 3

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016. 1, 5

[19] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 1, 5

[21] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In *CVPR*, 2017. 1, 5

[22] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018. 1, 5, 6, 7

[23] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1, 5, 6, 7

[24] Y.-C. Su and K. Grauman. Learning spherical convolution for fast features from 360° imagery. In *NIPS*, 2017. 2, 6

[25] Y.-C. Su and K. Grauman. Making 360deg video watchable in 2d: Learning videography for click free viewing. In *CVPR*, 2017. 1

[26] Y.-C. Su, D. Jayaraman, and K. Grauman. Pano2vid: Automatic cinematography for watching 360° videos. In *ACCV*, 2016. 2

[27] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *IJCV*, 2013. 5

[28] J. Xiao, K. A. Ehinger, A. Oliva, and A. Torralba. Recognizing scene viewpoint using panoramic place representation. In *CVPR*, 2012. 2, 3

[29] W. Yang, Y. Qian, J.-K. Kämäräinen, F. Cricri, and L. Fan. Object detection in equirectangular panorama. In *ICPR*, 2018. 2, 3, 4

[30] Y. Yu, S. Lee, J. Na, J. Kang, and G. Kim. A deep ranking model for spatio-temporal highlight detection from a 360° video. In *AAAI*, 2018. 2, 3