CalorieCam360: Simultaneous Eating Action Recognition of Multiple People Using an Omnidirectional Camera

Kento Terauchi
The University of Electro-Communications, Tokyo, Japan
terauchi-k@mm.inf.uec.ac.jp

Keiji Yanai
The University of Electro-Communications, Tokyo, Japan
yanai@cs.uec.ac.jp

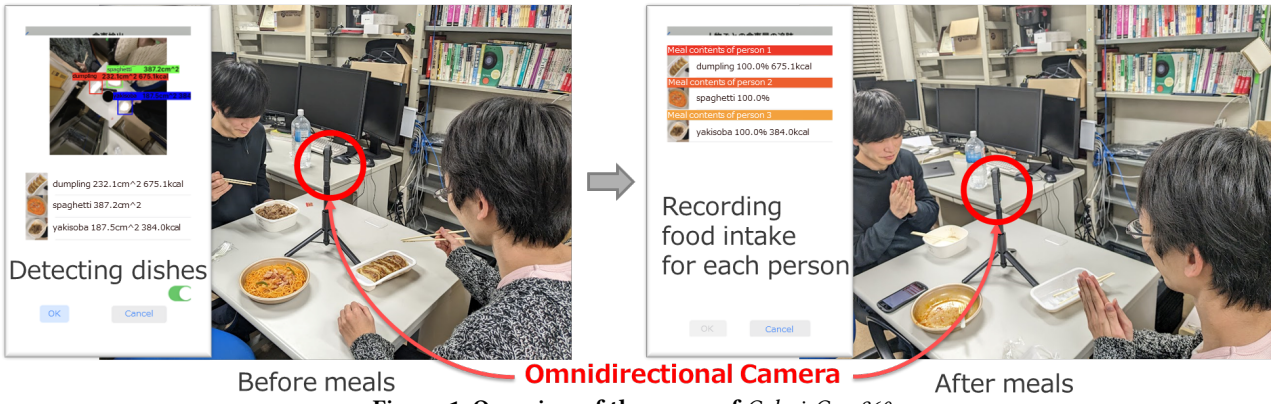

Figure 1: Overview of the usage of *CalorieCam360*.

ABSTRACT

In recent years, as people become more health-conscious, dietary management has become increasingly important. Existing methods record only one person's meals or eating movements, but cannot record the meals of multiple people at the same time. Therefore, we aim to record the meals of all people around a dining table using an omnidirectional camera simultaneously.

In this study, we propose *CalorieCam360*, a system that records the entire dining table using only an omnidirectional camera and a smartphone. Note that all the processing is done inside the smartphone application without using any external servers. Since the images from the omnidirectional camera are distorted and cannot be used for detection as they are, the distortion is corrected using plane projection. The corrected images are used to detect rectangular objects that serve as references for object size, and the area is calculated by combining object detection and region segmentation to estimate the amount of calories from the area. The system then uses person detection and region segmentation to track the person and the food and records the amount of food consumed and its calorie content for each person. We demonstrate that *CalorieCam360* can record an entire meal at once for multiple people around the table.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**; • **Human-centered computing** → **Interactive systems and tools**.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
ICMR '23, June 12–15, 2023, Thessaloniki, Greece
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0178-8/23/06.
https://doi.org/10.1145/3591106.3592221

KEYWORDS

food recognition, neural networks, omnidirectional camera, smartphone application

ACM Reference Format:
Kento Terauchi and Keiji Yanai. 2023. CalorieCam360: Simultaneous Eating Action Recognition of Multiple People Using an Omnidirectional Camera. In *International Conference on Multimedia Retrieval (ICMR '23), June 12–15, 2023, Thessaloniki, Greece.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3591106.3592221

1 INTRODUCTION

In recent years, as people become more health-conscious, dietary management has become increasingly important. Dietary management applications allow users to better understand their own eating habits and improve their eating habits by recording the amount of food they eat and the amount of calories they consume. Existing methods record only a meal for one person or only the eating behavior of a person. It is not possible to estimate foods for multiple people at the same time. In the case of a family gathering around a dining table, it is not efficient for each person to use their own smartphone devices and applications to record the meals. The omnidirectional camera can capture the entire dining table at once, including the table and people around it, with a single device by combining images from two wide-angle cameras. Therefore, we aim to simplify the process by using an omnidirectional camera to record all the dishes at a dining table at once to easily record the meal of all the people around the dining table. Omnidirectional cameras have become more familiar in recent years due to the widespread use of VR and their low price. However, there has been little research on the application of deep learning to omnidirectional cameras in real time, so we need to devise new ways to use omnidirectional cameras. In addition, since the omnidirectional camera is expected to be carried around, we implemented our system as a smartphone application, which can be carried around more easily.

The implementation of deep learning models on smartphones has become easier in recent years thanks to the development of libraries such as CoreML and Pytorch Mobile. At the same time, research on models that save computational resources, save memory, and run at high speed for mobile devices has been widely conducted. Therefore, our system does not need external servers, and all the processing is done inside a smartphone.

In this study, we construct a system, CalorieCam360, that records the entire meals at a dining table using only an omnidirectional camera and a smartphone, as shown in Figure 1. The system estimates the amount of calories in each food at the entire dining table and records the amount of calories in each food consumed by each person at the entire dining table. The amount of calories in a food can be estimated based on the food category and the actual size area of the food. The actual size area is obtained by detecting a rectangular object as a reference, and the user inputs the area of the object. In order to record the amount of calories consumed by each person at the entire dining table, we detect persons who are eating and assigning the detected foods to each of the detected persons. The size change of each of the detected foods is also detected to calculate food calories consumed by each of the persons by subtracting the difference of the segmented areas between frames.

2 RELATED WORKS

2.1 Dietary management applications

Some meal management applications take pictures of food and estimate the amount of calories [2, 6–10, 13, 15, 17, 18], while others recognize eating movements and estimate the amount of calories consumed [1, 14]. However, existing applications only recognize a meal for one person and do not assume the case where multiple people are sitting around a dining table. This means that each application needs to be used even though multiple people are seated at a single table.

In this study, we attempt to recognize the entire dining table at once by using an omnidirectional camera.

2.2 Object recognition of omnidirectional images

In object recognition of omnidirectional images, existing studies use the same object recognition methods as those used for general images. There are several ways to apply object detection methods. Chou *et al.* [4] trained with directly annotated equirectangular image. Zhang *et al.* [21] trained by adding the distortions of the equirectangular view to the training image. Yang *et al.* [20] trained on general images and corrected the distortion of the equirectangular image during inference.

In this study, we correct the distortion of the equirectangular image using projection and detect dishes using a planar projection suitable for food images.

3 CALORIECAM360

3.1 Overview of the proposed method

We aim to estimate the amount of calories in each person's diet for the entire dining table using an omnidirectional camera and a smartphone. The system is implemented so that all processing can be completed only with the omnidirectional camera and smartphone, without using an external server. The deep learning model is implemented by converting it to a CoreML model so that it can be completed entirely in Apple's Vision framework [11]. The omnidirectional camera is Insta 360 ONE X2. CalorieCam360 can be divided into four phases as shown in Figure 2. (1) For reference size determination, the user selects a rectangular object of a known area and inputs the actual area size to estimate the area per pixel, which enables estimation of the actual size of the food. (2) For food object detection, we use YOLO v7 [19] trained with UEC-FOOD100 [12] to detect the location and category of food objects. (3) For calorie estimation of food objects, we perform food region segmentation using DeepLab v3+ [3], calculates the actual size area of food objects, and estimates the calorie content from the area. (4) For the estimation of calorie intake for each person, we continuously perform food area segmentation and track the area. The proposed method also tracks the person at the same time, and by mapping the person to the dish, it is possible to calculate the proportion of each meal eaten by each person. Based on the percentage of food eaten, it is possible to calculate the amount of calories in each person's diet. In each step, an image of the table is used, corrected for distortion using plane projection, in order to handle the information about the food placed on the table.

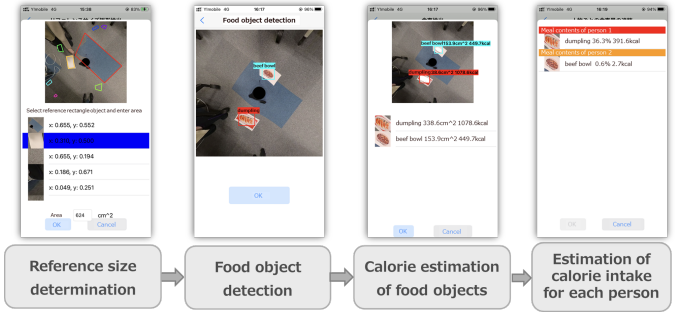

Figure 2: The porocessing steps of CalorieCam360.

3.2 Planar projection over a table

The image from the omnidirectional camera can be acquired frame by frame as an equirectangular image. To correct the distortion and make the image suitable for detection, the proposed system uses a plane projection of the table as the target. In-plane projection, the meal is assumed to be on the horizontal plane below the camera, and the equirectangular image, which can be represented as a sphere, is projected onto the horizontal plane. An example result of the projection is shown in Figure 3. The distorted table at the bottom of the pre-projection image and the meal on it are shown as flat surfaces in the post-projection image.

Figure 3: Planar projection over a table.

3.3 Determination of reference size

To estimate the amount of calories in a dish, the area of the object is used in this study. However, to obtain the area, the actual dimensions of the object captured by the camera are required. Then, the real dimensions are calculated by detecting rectangular objects whose areas are known to the user. Since the food detection is performed on the image after plane projection, the calculation of the reference size is also performed on the image after plane projection. After the rectangle detection, the user selects a rectangular object of a known area and enters the area. The rectangle detection uses the rectangle detection function of Apple's Vision framework. The detected rectangles are represented by four points (top-left, bottom-left, bottom-right, and top-right), and the number of pixels surrounded by these four points are calculated. The area per pixel can be calculated by dividing the input area by the number of pixels in the detection rectangle.

3.4 Food object detection

For object detection, we project images acquired by the omnidirectional camera, and then perform meal detection using YOLO v7 [19] trained with bounding box annotations of foods in UEC-FOOD100 [12]. To handle images of the entire table, the dishes in the images are small. To accommodate small objects, the image scale is resized from 0.04x to 0.3x in the data expansion of the training image. For the omnidirectional image after projection, the image is rotated from -180° to 180° because the dish is seen from all directions. The image resolution is fixed to 640 for training and 1280 for inference. The center of the projected image is directly under the omnidirectional camera and is black since it is a blind spot. Therefore, the bounding box in the middle of the image is removed because it may be recognized as a dish.

3.5 Food area segmentation

After detecting the food object, we calculate the number of pixels of the food by dividing the detection bounding box into regions and estimating the area based on the number of pixels. Segmentation is performed on the image after plane projection as in the case of dish object detection. The semantic segmentation model Deeplab v3+ [19] is used to segment the dish area. We implement Deeplab v3+ using the domain segmentation toolset MMSegmentation [5] and train it using standard settings. The dataset used for training is UEC-FoodPIX Complete [16]. UEC-FoodPIX Complete is a dataset of 10,000 food images in 100 categories, each image is manually annotated in pixels per category.

3.6 Calorie estimation of food objects

The proposed method estimates the amount of calories in a meal based on the area obtained from the food area segmentation. In estimating the amount of calories in a meal, we calculate the amount of calories from the area and categories, following the method of Okamoto *et al.* [15] Using the regression curve of the amount of calories created by Tanno *et al.* [18], we calculate the amount of calories from the area of the food obtained by food area segmentation and the amount of calories per area corresponding to the category detected by object detection, as shown in Figure 4 Although we detect 100 food categories, currently we have the amount of calories per area for only 17 food categories. Therefore, no calorie estimation is performed for the categories for which no data are available.

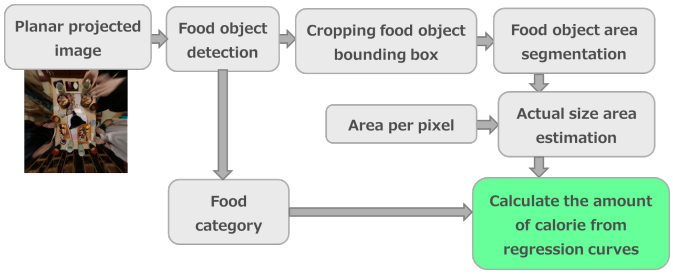

Figure 4: Processing flow of food calorie estimation.

3.7 Estimation of calorie intake for each person

Consider the estimation of calorie intake for each person using a table. If we only deal with the dish information, we cannot observe the amount of calories consumed by each person using the table. Therefore, it is necessary to track the people using the table. We detect all the persons who are eating, assign meals to each of the persons, calculate the amount of food intake for each person based on the amount of decrease in food regions, and convert the amount of food intake into the amount of calorie intake. We use the human pose detection function of the iOS Vision framework for person detection. The pose detection function provides information on the location of landmarks such as the neck, head, hands, and feet of each detected person. The detected person is treated as the same person as the nearest person detected in the previous frame to track the person. If a person is too far away or has more detected persons than in the previous frame, it is treated as a newly detected person. We estimate the amount of food eaten by each person by assigning each dish to the person currently eating it. We assign each dish to the person with the nearest wrist and elbow from the locatin of the dish. We determine the amount of food intake by the amount of decrease in the food area. We first set the amount of food consumed by all persons for all dishes to 0, and then we estimate the amount of food consumed by each person for each dish in each frame. In each frame, for each dish, we add the change in the food area in that frame to the intake of the corresponding dish of the person. In this way, we record the calorie intake for each dish for each person.

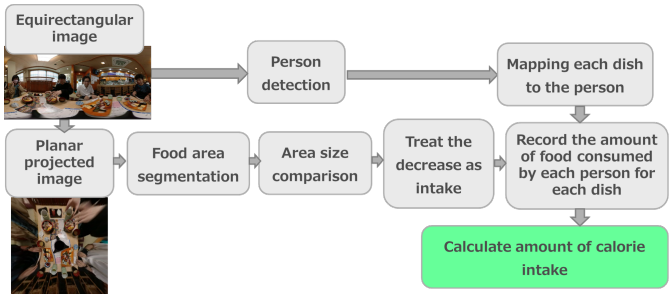

Figure 5: The whole processing flow of *CalorieCam360*.

4 EXAMPLE USAGE OF *CALORIECAM360*

The actual results used are shown in Figure 6, 7, 8, and 9. As a preparation, as shown in Figure 6, (a) we set up a camera on the table and (b) detect the rectangle surrounding the A4 paper by projecting the A4 paper on the reference size determination screen, (c) after confirming that the rectangle is detected, select the A4 paper rectangle, enter $624cm^2$, the area of the A4 paper, and finish the reference size determination.

Next, the food object detection screen is displayed, as shown in Figure 7, (a) we place the dishes on the table and detect the placed food. (b) When the dishes are detected, we press the button to confirm that the dishes are actually detected. Once it is confirmed that the food has been detected, the system proceeds to track the amount of calories consumed by each person.

As we eat the dishes, we can see that the remaining amount of food in percentage notation decreases as shown in Figure 8.

After finishing the meal, we can confirm that (a) the remaining amount of each meal has decreased as shown in Figure 9. (b) Clicking the "View meal result" button will take us to the "View Meal Result" screen, where we can see the amount of calories consumed by each person for each dish. If the amount of calories can be estimated for the food category, the amount of calorie intake is also displayed. As a result, the beef bowl was incorrectly detected as yakisoba, and detection and tracking sometimes do not work well. Spaghetti was detected, but the tracking of spaghetti was unstable from a certain time. Dumplings were detected and tracked well.

The estimated correspondence between the dishes and the persons is shown in Figure 10, where the person in the left of the 360 directional image is shown as "person 1", the person in the center as "person 2", and the person in the right as "person 3". It can also be seen that the nearest dishes are assigned to the corresponding persons.

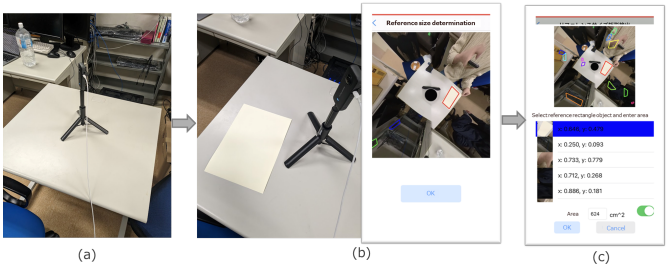

Figure 6: Example usage of *CalorieCam360*: Preparation

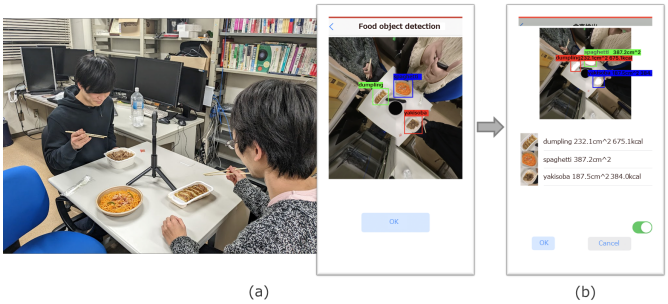

Figure 7: Example usage of *CalorieCam360*: Before meals

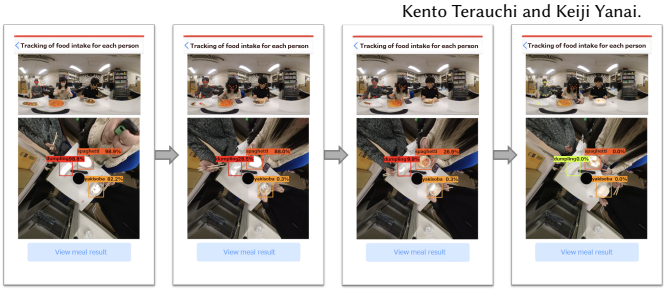

Figure 8: Example usage of *CalorieCam360*: During meals

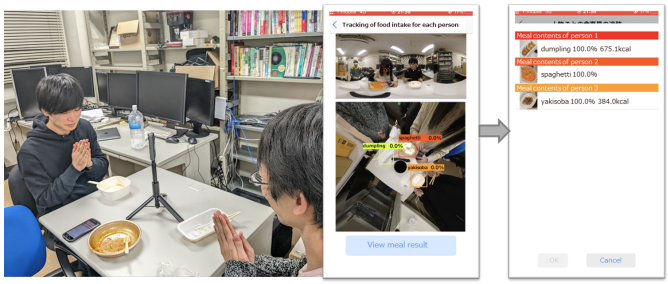

Figure 9: Example usage of *CalorieCam360*: After meals

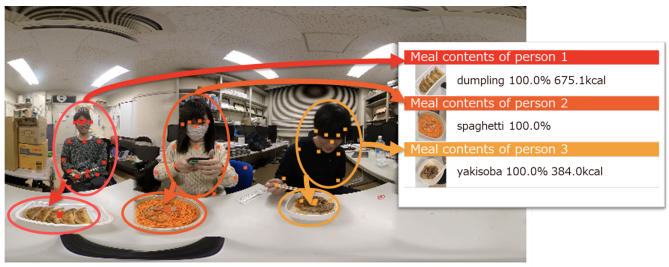

Figure 10: The estimated correspondence between the persons and the dishes.

5 CONCLUSION

For easy recording of the foods of everyone around the dining table by capturing all the food at once, we proposed an application that recognizes all the food at the table using an omnidirectional camera and a smartphone. The application, CalorieCam360, records the meal from the beginning to the end using the functions of reference size determination, food object detection, food calorie estimation, and calorie estimation for each person and calculation of the amount of food for each dish for each person. The experiments have shown that CalorieCam360 can record the entire dining table at once when several people are actually seated around the table.

As future work, we plan to improve the usability of CalorieCam360 by improving the model at each stage and by adding some modules such as face detection to track people and object tracking to track food objects.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Numbers, 22H00540, 22H00548, and 22K19808.

REFERENCES

[1] Kento Adachi and Keiji Yanai. 2022. DepthGrillCam: A Mobile Application for Real-Time Eating Action Recording Using RGB-D Images. In *Proc. of the 7th International Workshop on Multimedia Assisted Dietary Management.*

[2] Yoshikazu Ando, Takumi Ege, Jaehyeong Cho, and Keiji Yanai. 2019. Depthcaloriecam: A mobile application for volume-based food calorie estimation using depth cameras. In *Proc. of International Workshop on Multimedia Assisted Dietary Management.*

[3] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *Proc. of European Conference on Computer Vision.*

[4] Shih-Han Chou, Cheng Sun, Wen-Yen Chang, Wan-Ting Hsu, Min Sun, and Jianlong Fu. 2020. 360-Indoor: Towards Learning Real-World Objects in 360° Indoor Equirectangular Images. In *Proc. of IEEE Winter Conference on Applications of Computer Vision.*

[5] MMSegmentation Contributors. 2020. MMSegmentation: OpenMMLab Semantic Segmentation Toolbox and Benchmark. https://github.com/open-mmlab/mmsegmentation.

[6] Takumi Ege, Yoshikazu Ando, Ryosuke Tanno, Wataru Shimoda, and Keiji Yanai. 2019. Image-Based Estimation of Real Food Size for Accurate Food Calorie Estimation. In *Proc. of IEEE International Conference on Multimedia Information Processing and Retrieval.*

[7] Takumi Ege, Wataru Shimoda, and Keiji Yanai. 2019. A New Large-scale Food Image Segmentation Dataset and Its Application to Food Calorie Estimation Based on Grains of Rice. In *Proc. of ACMMM Workshop on Multimedia Assisted Dietary Management.*

[8] Takumi Ege and Keiji Yanai. 2017. Estimating Food Calories for Multiple-dish Food Photos. In *Proc. of Asian Conference on Pattern Recognition.*

[9] Takumi Ege and Keiji Yanai. 2017. Image-Based Food Calorie Estimation Using Knowledge on Food Categories, Ingredients and Cooking Directions. In *Proc. of the on Thematic Workshops of ACM Multimedia.*

[10] Takumi Ege and Keiji Yanai. 2018. Multi-task Learning of Dish Detection and Calorie Estimation. In *Proc. of International Workshop on Multimedia Assisted Dietary Management.*

[11] Apple Inc. 2023. Vision | Apple Developer Documentation. https://developer.apple.com/documentation/vision.

[12] Yuji Matsuda, Hajime Hoashi, and Keiji Yanai. 2012. Recognition of Multiple-Food Images by Detecting Candidate Regions. In *Proc. of IEEE International Conference on Multimedia and Expo.*

[13] Shu Naritomi and Keiji Yanai. 2021. Hungry Networks: 3D Mesh Reconstruction of a Dish and a Plate from a Single Dish Image for Estimating Food Volume. In *Proc. of ACM International Conference on Multimedia in Asia.*

[14] Koichi Okamoto and Keiji Yanai. 2014. Realtime eating action recognition system on a smartphone. In *Proc. of IEEE International Conference on Multimedia and Expo Workshops.*

[15] Koichi Okamoto and Keiji Yanai. 2016. An Automatic Calorie Estimation System of Food Images on a Smartphone. In *Proc. of International Workshop on Multimedia Assisted Dietary Management.*

[16] Kaimu Okamoto and Keiji Yanai. 2021. UEC-FoodPIX Complete: A Large-scale Food Image Segmentation Dataset. In *Proc. of ICPR Workshop on Multimedia Assisted Dietary Management.*

[17] Ryosuke Tanno, Takumi Ege, and Keiji Yanai. 2018. AR DeepCalorieCam: An iOS App for Food Calorie Estimation with Augmented Reality. In *Proc. of International Conference on Multimedia Modeling.*

[18] Ryosuke Tanno, Takumi Ege, and Keiji Yanai. 2018. AR DeepCalorieCam V2: Food Calorie Estimation with CNN and AR-Based Actual Size Estimation. In *Proc. of ACM Symposium on Virtual Reality Software and Technology.*

[19] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696* (2022).

[20] Wenyan Yang, Yanlin Qian, Joni-Kristian Kämäräinen, Francesco Cricri, and Lixin Fan. 2018. Object Detection in Equirectangular Panorama. In *Proc. of International Conference on Pattern Recognition.*

[21] Yiming Zhang, Xiangyun Xiao, and Xubo Yang. 2017. Real-Time Object Detection for 360-Degree Panoramic Image Using CNN. In *Proc. of International Conference on Virtual Reality and Visualization.*