You Only Look Once in Panorama: Object Detection for 360° Videos with MLaaS

Linfeng Shen[1], Miao Zhang[1], Cong Zhang[2], Jiangchuan Liu[1]
[1]Simon Fraser University, [2]The University of Hong Kong
{linfengs@,mza94@,jcliu@}sfu.ca,zcong@hku.hk

ABSTRACT

360° videos are gaining popularity, but immersive analytics, particularly in object detection, confront challenges from complex scenes and high data volume. This imposes significant burdens on individual users and resource-limited edge devices. Fortunately, Machine Learning as a Service (MLaaS) offers an economical solution for quick deployment without specific hardware or expertise. However, current MLaaS are mostly 2D image-designated and not optimized for the distinctive characteristics of raw 360° video frames. In this paper, we propose a novel MLaaS-based system to address this challenge. Our solution partitions 360° frames into distortion-free 2D regions with dynamic region of interest prediction. We then present an image-stitching algorithm featuring Skyline representation, seamlessly combining all the 2D regions into a unified frame. This frame is then transmitted to the MLaaS platform, with the detected objects being back-projected to yield the final results. Our experiments demonstrate the superiority of this system over baselines, proving its effectiveness in 360° video object detection tasks.

CCS CONCEPTS

• **Networks** → **Cloud computing**; • **Information systems** → **Multimedia information systems**.

KEYWORDS

360° video, Object Detection, Machine Learning as a Service

ACM Reference Format:
Linfeng Shen, Miao Zhang Cong Zhang, Jiangchuan Liu. 2024. You Only Look Once in Panorama: Object Detection for 360° Videos with MLaaS. In *The 34th edition of the Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV '24), April 15–18, 2024, Bari, Italy.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3651863.3651876

1 INTRODUCTION

Consumer-level omnidirectional cameras have gained unprecedented availability and affordability (e.g., GoPro[1] and Insta360[2]), and their market size has reached US$ 1.07 Billion in 2022 and is expected to expand to US$ 4.34 Billion by 2028 [9]. Most major video sharing websites, such as YouTube, Netflix, and Facebook, have started providing 360° video access. Besides being popular for offering immersive and interactive experiences for viewers [20, 32], 360° videos can also be consumed by machines to acquire knowledge and actionable insights of the physical world without blind spots [22], which can help various extended reality applications seamlessly bridge the gap between the physical and virtual worlds.

Most global streaming services have introduced machine learning (ML) into their operations nowadays [13]. Unfortunately, the existing video analytics systems [10, 29] that leave the vision model training and implementation to users can present high barriers to entry. For example, state-of-the-art object detection models normally require intensive resources to train and make inferences [12]. Collecting large datasets to ensure generalization presents another challenge. Moreover, the fast iteration speed of vision models further incurs high maintenance and retraining costs. Developing and maintaining ML models is challenging and costly for small and medium-sized businesses (SMBs) that lack the resources and expertise. Thanks to the recent advancements in ML techniques and the development of cloud services, machine learning as a service (MLaaS) is provided by major cloud providers, such as Amazon Web Services[3] and Microsoft Azure[4], for solving various ML tasks. MLaaS turns out to be the best solution for these users as it eliminates the need for specialized coding skills or on-premise infrastructure. In MLaaS, only machine learning APIs are exposed, whereas end users generally do not consider the internal implementation. The high abstraction of MLaaS can greatly free the users from the training and maintenance of the ML models. Another advantage of MLaaS is its high compatibility and portability, which can be quickly iterated internally, adapting to the fast iteration speed of current AI algorithms.

Unfortunately, no cloud providers currently provide MLaaS for 360° videos. Due to the characteristics of 360° videos, the model and criteria for different video analytics tasks need to be designed specifically [27][28][33], and the training becomes much more difficult because of the lack of datasets and the large size of 360° videos [26]. How to utilize the current 2D image-designated MLaaS for 360° videos remains a great challenge. Arguably, object detection [12], which detects where and what objects appear in an image, is one of the most important video analytics tasks. Advanced video analytics pipelines, such as license plate recognition, often start from object detection [30]. Thus, we focus on the object detection task for 360° videos as a pioneering study in this work.

Equirectangular Projection (ERP) is one of the most popular formats for 360° videos. Like the conventional 2D images, we can directly send equirectangular panorama frames of 360° videos to

[1]https://gopro.com/
[2]https://store.insta360.com/

[3]https://aws.amazon.com
[4]https://azure.microsoft.com

NOSSDAV '24, April 15–18, 2024, Bari, Italy

ACM ISBN 979-8-4007-0613-4/24/04
https://doi.org/10.1145/3651863.3651876

the MLaaS platforms to get the object detection results. However, the detection results on the equirectangular images are inaccurate because of the limitations of the sphere-to-plane projections, which are further illustrated in the measurement section. Other formats such as Cube Map and Equi-Angular Cubemap (EAC) are distortion-free, but the object detection results on the raw frames are still limited because of the uneven distribution of objects. On the other hand, the large size of the 360° video frames will incur a heavy burden for the network bandwidth. These limitations preclude the object detection tasks on 360° videos with the current MLaaS. To this end, we present a system that tackles the challenge of object detection tasks in 360° videos with MLaaS. Our contributions can be summarized as follows:

- To the best of our knowledge, we are the first to propose a system that utilizes current 2D images-designated MLaaS for object detection tasks in 360° video.
- We conduct a measurement based on real-world 360° videos, and the results show the limitations of the raw ERP frame.
- We propose a region of interest (RoI) prediction algorithm to select regions on the frame that are likely to contain objects.
- We propose an image stitching algorithm to combine RoIs into one single image as the input of the MLaaS APIs.
- We implement a prototype of the proposed system, and the experiment results validate its effectiveness.

2 RELATED WORK

Object Detection for 360° Videos. As one of the most important vision tasks, object detection tasks for 360° videos attract much academic interest. Since 360° images cannot be projected to a single planar image without distortion, the accuracy of the powerful networks researchers have carefully honed for 2D images is limited for raw 360° images. There are two main orientations to achieve high detection accuracy for 360° images. The first is to design a dedicated convolutional network that processes 360° images directly in its equirectangular projection. For example, Su *et al.* [21] propose an approach that learns to reproduce the flat filer outputs on 360° data. When viewing the sphere, this filter is sensitive to the varying distortion effects. In this way, the feature extraction process is both efficient and powerful for the 360° data. Coors *et al.* [5] present SphereNet, which encodes invariance against such distortions explicitly into the networks. By building on regular convolutions, SphereNet can transfer from the conventional perspective models to the omnidirectional case.

Another way is to convert the entire spherical image to multiple distortion-free perspective images via projections. In this way, each projected image corresponds to a partial FoV on the sphere, and the off-the-shelf object detection models can be directly used. For example, Yang *et al.* [26] generate four projections with 180° horizontal and vertical spans, which are then separately processed by the YOLO detector. Eder *et al.* [7] propose "tangent images," which render a spherical image to a set of distortion-mitigated, locally-planar image grids tangent to a subdivided icosahedron. Some works also focus on the dedicated criteria to improve the detection results for 360° videos[33] [2]. All these methods require significant computational resources, making direct processing impossible with limited hardware, such as edge devices. Our work aims to complete vision tasks by only sending images to the MLaaS platform through APIs, which is a more economical choice for these users.

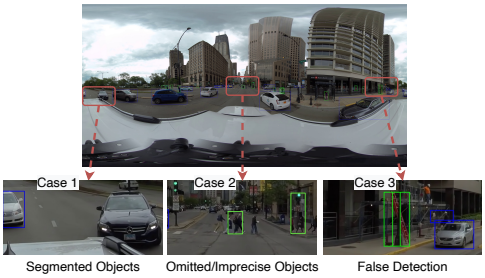


Figure 1: Limitations of equirectangular panorama.

ML Applications with MLaaS. Major cloud providers are deploying large MLaaS platforms to provide a host of ML applications. However, the servers for MLaaS are often equipped with heterogeneous GPU clusters and raise many challenges. Weng *et al.* [24] explain the challenges posed to cluster scheduling in MLaaS and propose solutions to better schedule the workloads by enabling GPU sharing among high-GPU and low-GPU tasks. Zhang *et al.* [31] focus on the privacy-preserving problem in MLaaS and investigate two potential strategies involving the optimization of cost hierarchy in the calculation process and the crypto-friendly pruning in the computation model. To combine the results from different service providers for better performance, Jiang *et al.* [11] propose a framework based on the constructed Probabilistic Graph Model and Expectation Maximization-based iteration algorithm to obtain high-quality results from multiple services within a budget constraint. Xie *et al.* [25] propose another framework to federate the selection of different MLaaS providers to achieve the best analytic performance. This work, however, only solves the problem of object detection tasks in conventional 2D images. The independence of images precludes the potential utilization of the relationship among frames in the video. There is still no work focus on the object detection task for 360° videos with MLaaS.

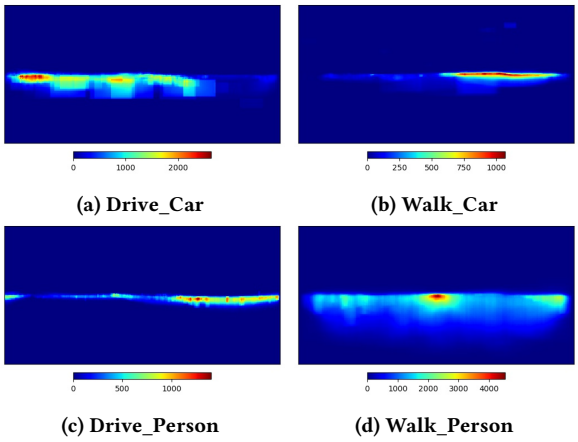
3 MEASUREMENT & MOTIVATION

In this work, we use two public long 360° videos (about 10 minutes and 20 minutes) in 8K resolutions for measurements and further experiments. One is "Drive in Chicago" (**Drive**)[5], and the other is "Walk in Shinjuku, Tokyo, Japan" (**Walk**)[6]. We take people and vehicle detections as a case study to investigate the object distribution and the limitations of the equirectangular projection. According to the analysis of measurements, we design a system that realizes satisfactory object detection performance with the current 2D-image designated MLaaS. We select the MLaaS platform Amazon Rekognition[7] as a case study in this paper.

3.1 Limitations of Equirectangular

Although equirectangular projection can contain all information of 360° data on a single rectangular image, object detection with

[5]https://www.youtube.com/watch?v=Gu1D3BnIYZg&t=4s
[6]https://www.youtube.com/watch?v=YYQufxYrBiU
[7]https://aws.amazon.com/rekognition



(a) Drive_Car (b) Walk_Car

(c) Drive_Person (d) Walk_Person

Figure 2: Heatmap of objects location. The color represents the frequency of the objects.

MLaaS has limitations on such equirectangular panorama. This is because of the distortions (especially in the polar regions) and discontinuities in the boundary caused by the projection. Fig. 1 shows some cases by sending one equirectangular panorama directly to the MLaaS platform. Because of the discontinuities in the boundary region, the detected bounding boxes may contain incomplete objects (case 1), and one object will be counted several times or not detected at all. On the other hand, the central regions have fewer distortions, but the size of the objects may be very small and can be omitted by the detector or cause imprecise bounding boxes (case 2). Since one equirectangular frame can be very large (usually in 8K resolution) to guarantee feasible viewing quality, some objects will occupy a very small portion of the frame and cause many false detections (positives and negatives) in object detection (case 3). These limitations show that directly using the equirectangular panorama is not a feasible solution for object detection tasks with MLaaS.

3.2 Ground Truth Setup

Unlike conventional 2D images, which have mature annotation tools and large datasets like Microsoft COCO [15], the evaluation criteria and ground truth of 360° videos datasets have not come to an agreement in academia. There is consensus to use model-generated datasets as ground truth in video analytics [10][30] since we focus on the algorithm of streaming or other processes instead of the accuracy of the model itself. To this end, we use a similar pipeline as in [30] to generate ground truth in this work. As the efficiency is not considered in this process, we use the largest model *YOLOv5x6* of YOLOv5[12] for best accuracy and run the model on a local machine to get the ground truth.

This pipeline contains two stages. In the first stage, we run the model on the six faces of the CubeMap projection (each face represents a 90-degree FoV horizontally and vertically) to get the rough detection results. In the second stage, for each of the detected objects in the first stage, we project a 60°×45° region centered at the bounding box to a distortion-free image via gnomonic projection [6]. We run the model on each projected image to get more precise detection results in each region. To avoid the segmented objects, we only keep the bounding boxes that do not locate at the boundary of the regions. Then each bounding box will be back-projected to the origin equirectangular panorama. This back projection will be further explained in the system section. At last, we get the final ground truth by using non-maximum suppression (NMS) [18] on all bounding boxes to remove duplicates.

3.3 Observation and Motivation

After getting the ground truth, we explore the selected 360° videos for some observations that may be useful for the design of our system. At first, we get and plot the location of each type of object on the ERP panorama. The heatmap in Fig. 2 shows the distribution of objects in the **Drive** and **Walk** 360° videos. We find that the objects in 360° videos only occupy a small portion of the panorama, and most regions are trivial for the object detection tasks. Sending these regions to the MLaaS platform not only wastes the network bandwidth but also affects the detection results of other regions since the detection models need to resize the input[8]. These regions make the objects much smaller in the frame and become difficult to be detected. Another observation is that these regions are not static and should change dynamically, as the content of the frames in one video can vary a lot. In our measurement, the number of each object type can vary a lot among different frames. Some frames have more objects, while others may have less objects. These observations motivate us to dynamically select the regions where the objects are most likely to appear and omit other regions. The selection of the region of interest (RoI) remains a challenge in the system, and we proposed a RoI prediction algorithm to solve it. Sending the gnomonic projection of each RoI as an independent image to the MLaaS platform, of course, can get good detection results. This will increase the cost several times since most MLaaS are charged by the number of processed images. For example, the price of the US East (Ohio) server of Amazon Rekognition is 0.0001$ per image[9]. To include the content of all RoIs in a single image, we propose an image-stitching algorithm that combines all the projected RoIs in one image. Predicting the RoIs and stitching them into one image are two major challenges in our system.

4 SYSTEM DESIGN

Our measurement and motivation study has revealed the characteristics of 360° videos and the challenges of using current MLaaS for object detection tasks. To address these challenges, we propose a system for improved detection results with MLaaS while using a similar cost as directly sending the ERP frame. The overview of the system is shown in Fig. 3. For each raw input frame in ERP format, the RoIs are predicted from the detection results of the previous frame (Step 1). For the first frame of the video, the RoIs can be predicted from the raw results by sending the ERP image to MLaaS. Given the results of step one, we combine the sets of RoIs after gnomonic projection to a single image by an image stitching algorithm (Step 2). Next, we send this image which contains the content of all the RoIs to the MLaaS platform and obtain the detection results (Step 3 and Step 4). At last, each bounding box is back-projected on the origin frame to obtain the final results (Step

[8]https://docs.aws.amazon.com/rekognition/latest/dg/images-information.html
[9]https://aws.amazon.com/rekognition/pricing/

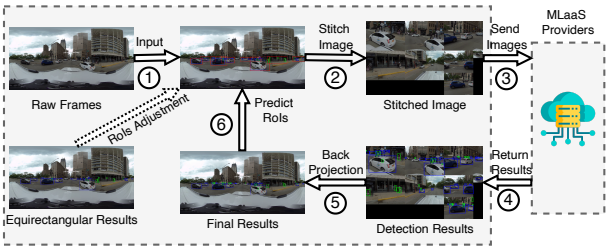


Figure 3: Overview of the system design.

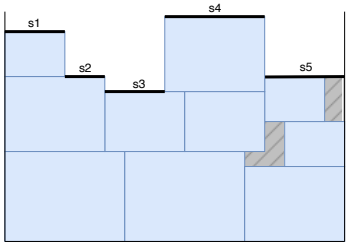


Figure 4: Skyline representation of the packing pattern.

5), and the results are utilized for the RoIs prediction of the next frame. Since the imprecise detection results may be accumulated among a sequence of frames, we add a RoIs adjustment process with a specific interval in the video. The RoIs are predicted from the results of the raw ERP input for these frames. Although the idea of stitching ROIs is also proposed in [8], this work tries to stitch the frames from different devices and is based on conventional 2D videos. While our work stitches the ROIs of a single 360° frame. The details of each step are illustrated in the following of this section.

4.1 RoI Prediction

Although the content in 360° videos is intensive and the number of objects has great variations in the video, the consecutive frames will not have great differences and it is reasonable to use the detection results of the most recent frames for RoI prediction. Considering the movement of the object between consecutive frames, each RoI has paddings at the boundaries. We merge all the bounding boxes into several RoIs for the further image stitching process. There are also some cases in which the object appears in the video first time or the object is not detected in the previous frames. To handle these special cases, we also add a RoIs adjustment process that runs the RoIs prediction algorithm directly with the detection results on the ERP frame. The details of the algorithm are shown in Algorithm 1.

At first, for each bounding box, we check if they can be merged with an existing RoI. If the size of the merged RoI is in the accepted range (h_SoI, v_SoI), the merged RoI is updated and added to the set of predicted RoIs (lines 4-6). We create a new RoI centered at the bounding box if it cannot be merged with an existing RoI, and the new RoI is added to predicted RoIs (lines 8-9). At last, we also record the minimum bounding box size of each RoI by adding a new item *Min_bb* to the data structure of the RoIs. The *Min_bb* is also updated accordingly during the merging of RoIs. Through the measurements, we found that the small objects are most likely to appear together in a region (passengers on the crossroad), and we want these regions to have high resolution for better detection results from MLaaS. This information is kept as guidance for the image-stitching process to create an image that has the best detection results with MLaaS.

Algorithm 1: RoI Prediction

Input : Detection results of the most recent frame B and the padding parameter p
Output: The set of predicted RoIs R

1 Initialize the set of predicted RoIs $R := \emptyset$;
2 **for** *each bounding box* $b \in B$ **do**
3 **if** *can merge* b *with an existing RoI* r **then**
4 Pop r from R;
5 Merge b and r to a new RoI r';
6 $R := R \cup r'$;
7 **else**
8 Create a new RoI r centered at b;
9 $R := R \cup r$;
10 Record the smallest bounding box of each RoI r;
11 **return** R

4.2 Stitch Image based on the Predicted RoIs

In the image stitching step, we first convert each RoI to distortion-free images using gnomonic projection. Gnomonic projection is a method of projecting points from a sphere onto a plane tangent to the sphere at a central point [6]. To ensure the RoIs with small objects have higher resolutions after the projection. We rate the resolution in four levels: 320×240, 640×480, 960×720, and 1280×960. During the projection, the resolution of the planar image depends on the smallest bounding box of each RoI, which is recorded in *Min_bb*. After each RoI is projected to a planar image, we stitch them and include all content in a single image. Stacking them one by one regularly is an obvious solution. The stitched image will have too much wasted area and result in poor performance of MLaaS, however. Indeed, this is a 2D rectangular packing problem (2DRP) [16] (packing sets of 2D images in a rectangular). We propose an image stitching algorithm to solve this problem using the Skyline [23] representation of a packing pattern.

An example of this Skyline representation is shown in Fig. 4. During the packing process, the images are placed one by one, and the contour of the current packing is represented as a sequence of N horizontal line segments $(s1, s2, ..., sN)$. These line segments satisfy the following properties: (1) the length of each line segment is larger than the width of one remaining image at least; (2) two consecutive line segments have different heights, or they can be merged into one line segment. When an image is placed on one line segment, the line segments should be updated accordingly. The remaining line segment will be raised and merged with a consecutive line segment if it cannot hold another image. These regions are labeled as wasted areas as the shaded area shown in Fig. 4. During the packing of each image, the position of its left-top point is recorded in a tuple *<Pos_x, Pos_y>* along with its height and width *<h, w>* for the back projection process in the next step.

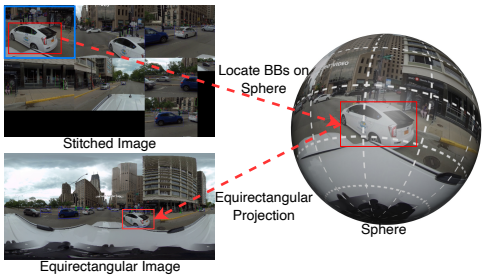
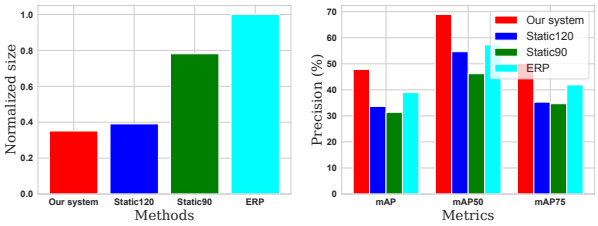


Figure 5: Back projection of one bounding box.



(a) Frame size **(b) Precision**

Figure 6: Comparison with baselines.

4.3 Back Projection

After the ROI prediction and image stitching steps, the stitched image is sent to the MLaaS platform. The detection results are much better than the ERP frame since the images now contain the most useful content for object detection. At last, the bounding boxes should also be back-projected on the raw frame to obtain the final results. Fig. 5 shows the back projection process where the red rectangular is the bounding box of one object and the blue rectangular is the RoI. First, we need to locate the bounding boxes on the sphere. According to the recorded <*Pos_x, Pos_y*> in the stitched image, we can identify the RoI to which each bounding box belongs. Given the coordinates of the point and the center of the FoV, the latitude and longitude of the point on the ERP image can be calculated through equirectangular projection.

In recent years, some specific types of bounding boxes (BBs) such as ERPBB, CirBB, and UnbBB (TanBB/SphBB) are proposed [33] for objection detection in 360° videos. The object detection results on ERP frames given by the MLaaS can only be conventional bounding boxes. In order to compare the performance of our system with the results on raw ERP, we still use the conventional bounding box in this work since we obtain the final BBs on the raw equirectangular frames instead of the spherical image. The BBs on the sphere are irregular in shape after back projection. To get regular rectangular BBs on the raw frame, we only locate the left-top and right-bottom points of the BBs on the sphere and then back-project the BBs on the raw frame. The RoIs have paddings in the RoIs prediction step, and one object can appear in several RoIs. To remove these duplicated detections, we use non-maximum suppression (NMS)[18] on the BBs with different labels. At last, the final BBs are utilized for the RoI prediction of the next frame.

5 IMPLEMENTATION AND EXPERIMENTS

5.1 Implementation Details

We implement a prototype of the proposed system on a local desktop computer with an Intel i7-12700 CPU and an Nvidia GeForce RTX 3080 GPU (GPU is only used in the process of getting ground truth). The system is implemented in *Python* and runs on Ubuntu 23.04 OS. We use *OpenCV* [19] for all video and image operations. We select Amazon Rekognition as the MLaaS provider in the experiments and use *Boto3* [1], the AWS SDK for Python to interact with the MLaaS APIs. The padding parameter in the RoI prediction step is set to 0.1. The threshold of IoU in the NMS process is set to 0.5. We keep all the bounding boxes with confidence scores larger than 0.2 to include as many objects as possible.

5.2 Experiment Setup

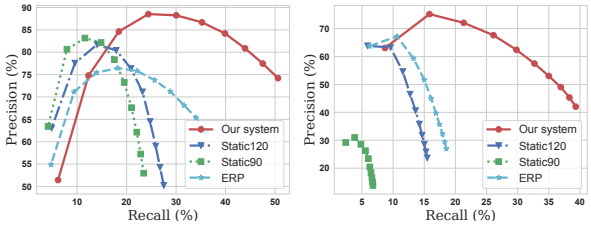
We evaluate our system and the baselines on the same videos as in the measurement section, which have over 100,000 frames. Mean Average Precision (mAP) is a common metric for measuring performance for object detection tasks and information retrieval. We use the three metrics (mAP, mAP50, and mAP75) defined in the COCO dataset [4] to evaluate the detection results in the experiment. The mAP represents the primary challenge metric (AP at IoU=.50:.05:.95), mAP50 represents the PASCAL VOC metric (AP at IoU=.50) and mAP75 represents the strict metric (AP at IoU=.75). Since there is no previous work that uses MLaaS for object detection in 360° videos, we define three baselines as follows:

- **ERP**: Send the raw equirectangular frames to the MLaaS provider and return the detection results.
- **Static120**: Use the similar way as in [26] to partition the raw frames into regions of 120°×120° FoV with 30° overlap statically, then the gnomonic projection of each region is combined into one image and sent to the MLaaS provider.
- **Static90**: The regions are of 90°×90° FoV without overlaps, and others are the same as **Static120**.

All of the methods have the same cost since only one image is sent to the MLaaS platform for each frame in the 360° videos (MLaaS is charged by the number of API requests). **ERP** sends the raw equirectangular frames to the MLaaS platform with distortions and contain useless content. **Static120** and **Static90** send distortion-free images but the choice of ROIs are static. Our system uses a dynamic RoI prediction algorithm and image stitching algorithm that can adapt to the content of 360° videos. To validate our motivation and the improvement of the proposed system, we compare it with baselines in both quantitative and qualitative evaluations.

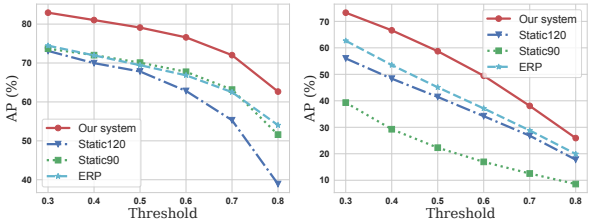
5.3 Evaluation Results

At first, we calculate the average normalized size (the size of the raw ERP frame is defined as 1) of all frames. Fig. 6a shows the results of four methods. The stitched images in our system are much smaller than the raw ERP frame (about 40%). Although previous work has shown that the transmission latency is much less than the inference latency for the conventional 2D images [25], the transmission latency is not negligible for the high-resolution 360° videos due to the large size of the frames. This improvement can save significant network bandwidth, which is very important for the streaming of 360° videos [3, 14, 17]. Next, we present the overall object detection performance of our system compared to the baselines. Fig. 6b shows the results of our system and three baselines. Our system



(a) PR Curve of Car (b) PR Curve of Person

Figure 7: PR Curve of different objects.



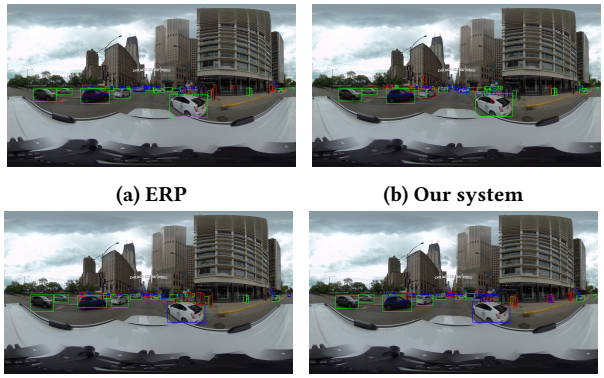
(a) AP of Car (b) AP of Person

Figure 8: AP with different thresholds.

can achieve the best performance with all three evaluation metrics. Although **Static 120** and **Static 90** can also save bandwidth by reducing frame size, the lack of RoI prediction results in bad detection results, even worse than directly sending ERP frames.

To further investigate the performance of different methods, we delve into two specific object categories (*Person* and *Car*). Fig. 7 compares the Precision-Recall (PR) Curve obtained from different methods. Our system performs best compared to the baselines on the objects of the *Car* and *Person*. For all the methods, the precision of *Car* is higher because these objects are usually larger than *Person* in the 360° frames, and small objects are more challenging to detect by MLaaS. The performance of **Static 90** for *Person* is significantly worse since the objects are too small and the RoI partitions do not overlap, leading to the missing of many objects. We also plot the AP of different objects with different thresholds in Fig. 8. When comparing the detection results with the ground truth, the threshold of IoU determines the precision of the bounding boxes. The higher threshold means the criteria for good detection are more stringent. The results in Fig. 8 show that our system has more precise bounding boxes compared to the baselines. Although the performance of all the methods will decrease with the larger thresholds, our system can still have relatively precise detection results from the MLaaS.

To see the advantages of our system more intuitively, we also do quantitative evaluations that compare our system with the baselines. Fig. 9 shows the detection results of one frame. We plot both the detection results and the ground truth. *True Positive* (objects that are detected correctly) results are plotted as green bounding boxes. *False Negative* (objects that are not detected) results are plotted as blue bounding boxes, and red bounding boxes represent the *False Positive* (wrong detections returned by MLaaS) results. **ERP** has bad results, especially in the central region where the objects are small. There are also some *False Positive* detections because of the distortion



(a) ERP (b) Our system

(c) Static120 (d) Static90

Figure 9: Detection results on one frame.

of the equirectangular projection. The results show that directly sending the ERP frame to the MLaaS platform is not an ideal solution for object detection tasks in 360° videos. **Static120** and **Static90** have improved performance in some regions. These regions have fewer *False Positive* detections since the images sent to the MLaaS platform are distortion-free after the gnomonic projection. However, some obvious objects are omitted since the partition of regions is static, and some objects may be segmented during the projection. Compared to the baselines, our system has the best performance. Although there are still some *False Positive* detections, our system can detect most of the small objects.

In conclusion, from both the qualitative evaluation and quantitative evaluation, our system has the best performance compared to the baselines. With a RoI prediction algorithm and an image-stitching algorithm, our system obtains the most accurate detection results compared to the baselines. It only sends one image for each frame in 360° videos to the MLaaS platform, which minimizes the cost of object detection tasks with MLaaS.

6 CONCLUSION

In this paper, we proposed a system that utilizes off-the-shelf MLaaS for object detection tasks in 360° videos. Motivated by our measurements and the analysis of the characteristics of 360° videos, we proposed a dynamic RoI prediction algorithm based on the detection results of the most recent frame. The predicted RoIs represent the regions where the objects are most likely to appear. Then each RoI is projected to a distortion-free planar through gnomonic projection, and all the projected RoIs are combined into a single image by the proposed image-stitching algorithm. At last, the detection results from the MLaaS on the stitched image are back-projected to get the final results. Extensive evaluations verified the feasibility of our motivations, and our system has the best performance compared to the baselines.

ACKNOWLEDGMENTS

We appreciate the constructive comments from the reviewers. This research is supported by an NSERC Discovery Grant, a British Columbia Salmon Recovery and Innovation Fund (BCSRIF_2022_401), and a MITACS Accelerate Cluster Grant.

REFERENCES

[1] Boto3. 2023. *AWS SDK for Python (Boto3).* Retrieved April 28, 2023 from https://aws.amazon.com/sdk-for-python/

[2] Miao Cao, Satoshi Ikehata, and Kiyoharu Aizawa. 2022. Field-of-View IoU for Object Detection in 360° Images. *arXiv e-prints* (2022), arXiv–2202.

[3] Lovish Chopra, Sarthak Chakraborty, Abhijit Mondal, and Sandip Chakraborty. 2021. Parima: Viewport adaptive 360-degree video streaming. In *Proceedings of the Web Conference (WWW'21).*

[4] COCO. 2023. *COCO: Detection Evaluation.* Retrieved Sep 28, 2023 from https://cocodataset.org/#detection-eval

[5] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. 2018. SphereNet: Learning Spherical Representations for Detection and Classification in Omnidirectional Images. In *Proceedings of the 15th European Conference on Computer Vision (ECCV'18).*

[6] Harold Scott Macdonald Coxeter. 1961. Introduction to geometry. (1961), 93 and 289–290.

[7] Marc Eder, Mykhailo Shvets, John Lim, and Jan-Michael Frahm. 2020. Tangent Images for Mitigating Spherical Distortion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'20).*

[8] Ila Gokarn, Hemanth Sabbella, Yigong Hu, Tarek Abdelzaher, and Archan Misra. 2023. MOSAIC: Spatially-multiplexed edge AI optimization over multiple concurrent video sensing streams. In *Proceedings of the 14th Conference on ACM Multimedia Systems (MMSys'23).*

[9] IMARC. 2023. *360-Degree Camera Market: Global Industry Trends, Share, Size, Growth, Opportunity and Forecast 2023-2028.* Retrieved Sep 28, 2023 from https://www.imarcgroup.com/360-degree-camera-market

[10] Junchen Jiang, Ganesh Ananthanarayanan, Peter Bodik, Siddhartha Sen, and Ion Stoica. 2018. Chameleon: Scalable Adaptation of Video Analytics. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication (SIGCOMM'18).*

[11] Shanyang Jiang and Lan Zhang. 2022. Quality-aided Annotation Service Selection in MLaaS Market. In *Proceedings of the IEEE/ACM 30th International Symposium on Quality of Service (IWQoS'22).*

[12] Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, ChristopherSTAN, Liu Changyu, Laughing, tkianai, Adam Hogan, lorenzomammana, yxNONG, AlexWang1900, Laurentiu Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Francisco Ingham, Frederik, Guilhen, Hatovix, Jake Poznanski, Jiacong Fang, Lijun Yu, changyu98, Mingyu Wang, Naman Gupta, Osama Akhtar, PetrDvoracek, and Prashant Rai. 2020. *ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements.* https://doi.org/10.5281/zenodo.4154370

[13] Sudarshan Lamkhede, Praveen Chandar, Vladan Radosavljevic, Amit Goyal, and Lan Luo. 2023. Machine Learning for Streaming Media. In *Companion Proceedings of the ACM Web Conference (WWW'23 Companion).*

[14] Jiaxi Li, Jingwei Liao, Bo Chen, Anh Nguyen, Aditi Tiwari, Qian Zhou, Zhisheng Yan, and Klara Nahrstedt. 2023. Latency-Aware 360-Degree Video Analytics Framework for First Responders Situational Awareness. In *Proceedings of the 33rd Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'23).*

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and Larry Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Proceedings of the 13th European Conference on Computer Vision (ECCV'14).*

[16] Andrea Lodi, Silvano Martello, and Michele Monaci. 2002. Two-dimensional packing problems: A survey. *European journal of operational research* 141, 2 (2002), 241–252.

[17] Yixiang Mao, Liyang Sun, Yong Liu, and Yao Wang. 2020. Low-Latency FoV-Adaptive Coding and Streaming for Interactive 360° Video Streaming. In *Proceedings of the 28th ACM International Conference on Multimedia (MM'20).*

[18] A. Neubeck and L. Van Gool. 2006. Efficient Non-Maximum Suppression. In *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06).*

[19] OpenCV. 2023. *OpenCV: Open source computer vision library.* Retrieved April 28, 2023 from https://opencv.org/

[20] Feng Qian, Bo Han, Qingyang Xiao, and Vijay Gopalakrishnan. 2018. Flare: Practical Viewport-Adaptive 360-Degree Video Streaming for Mobile Devices. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom'18).*

[21] Yu-Chuan Su and Kristen Grauman. 2017. Learning Spherical Convolution for Fast Features from 360° Imagery. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17).*

[22] Kuan-Hsun Wang and Shang-Hong Lai. 2019. Object Detection in Curved Space for 360-Degree Camera. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'19).*

[23] Lijun Wei, Wee-Chong Oon, Wenbin Zhu, and Andrew Lim. 2011. A skyline heuristic for the 2D rectangular packing and strip packing problems. *European Journal of Operational Research* 215, 2 (2011), 337–346.

[24] Qizhen Weng, Wencong Xiao, Yinghao Yu, Wei Wang, Cheng Wang, Jian He, Yong Li, Liping Zhang, Wei Lin, and Yu Ding. 2022. MLaaS in the Wild: Workload Analysis and Scheduling in Large-Scale Heterogeneous GPU Clusters. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI'22).*

[25] Shuzhao Xie, Yuan Xue, Yifei Zhu, and Zhi Wang. 2022. Cost Effective MLaaS Federation: A Combinatorial Reinforcement Learning Approach. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM'22).*

[26] Wenyan Yang, Yanlin Qian, Joni-Kristian Kämäräinen, Francesco Cricri, and Lixin Fan. 2018. Object Detection in Equirectangular Panorama. In *Proceedings of the 24th International Conference on Pattern Recognition (ICPR'18).* 2190–2195.

[27] Heeseung Yun, Sehun Lee, and Gunhee Kim. 2022. Panoramic Vision Transformer For Saliency Detection In 360° Videos. In *Proceedings of the 17th European Conference on Computer Vision (ECCV'22).*

[28] Ilwi Yun, Hyuk-Jae Lee, and Chae Eun Rhee. 2022. Improving 360 monocular depth estimation via non-local dense prediction transformer and joint supervised and self-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'22).*

[29] Haoyu Zhang, Ganesh Ananthanarayanan, Peter Bodik, Matthai Philipose, Paramvir Bahl, and Michael J. Freedman. 2017. Live Video Analytics at Scale with Approximation and Delay-Tolerance. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI'17).*

[30] Miao Zhang, Yifei Zhu, Linfeng Shen, Fangxin Wang, and Jiangchuan Liu. 2023. OmniSense: Towards Edge-Assisted Online Analytics for 360-Degree Videos. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM'23).*

[31] Qiao Zhang, Tao Xiang, Yifei Cai, Zhichao Zhao, Ning Wang, and Hongyi Wu. 2022. Privacy-Preserving Machine Learning as a Service: Challenges and Opportunities. *IEEE Network* (2022).

[32] Yuanxing Zhang, Pengyu Zhao, Kaigui Bian, Yunxin Liu, Lingyang Song, and Xiaoming Li. 2019. DRL360: 360-degree Video Streaming with Deep Reinforcement Learning. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM'19).*

[33] Pengyu Zhao, Ansheng You, Yuanxing Zhang, Jiaying Liu, Kaigui Bian, and Yunhai Tong. 2020. Spherical Criteria for Fast and Accurate 360° Object Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'20).*