IEEE INFOCOM 2024 - IEEE Conference on Computer Communications | 979-8-3503-8350-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/INFOCOM52122.2024.10621328

Resource-efficient In-orbit Detection of Earth Objects

Qiyang Zhang[1], Xin Yuan[1], Ruolin Xing[1], Yiran Zhang[1],
Zimu Zheng[2], Xiao Ma[1*], Mengwei Xu[1], Schahram Dustdar[3], Shangguang Wang[1]
[1]State Key Laboratory of Networking and Switching Technology, BUPT, Beijing, China
[2]Huawei Technologies Co., Ltd
[3]Distributed System Group, TU Wien, Vienna, Austria

{qyzhang;bupt_yx,xrl,yiranzhang,maxiao18,mwx,sgwang}@bupt.edu.cn;
dustdar@dsg.tuwien.ac.at; zimu.zheng@huawei.com

***Abstract*—With the rapid proliferation of large Low Earth Orbit (LEO) satellite constellations, a huge amount of in-orbit data is generated and needs to be transmitted to the ground for processing. However, traditional LEO satellite constellations, which downlink raw data to the ground, are significantly restricted in transmission capability. Orbital edge computing (OEC), which exploits the computation capacities of LEO satellites and processes the raw data in orbit, is envisioned as a promising solution to relieve the downlink burden. Yet, with OEC, the bottleneck is shifted to the inelastic computation capacities. The computational bottleneck arises from two primary challenges that existing satellite systems have not adequately addressed: the inability to process all captured images and the limited energy supply available for satellite operations. In this work, we seek to fully exploit the scarce satellite computation and communication resources to achieve satellite-ground collaboration and present a satellite-ground collaborative system named `TargetFuse` for onboard object detection. `TargetFuse` incorporates a combination of techniques to minimize detection errors under energy and bandwidth constraints. Extensive experiments show that `TargetFuse` can reduce detection errors by 3.4× on average, compared to onboard computing. `TargetFuse` achieves a 9.6× improvement in bandwidth efficiency compared to the vanilla baseline under the limited bandwidth budget constraint.**

***Index Terms*—EO, Satellite Computing, Counting**

I. Introduction

Earth-observation (EO) satellites collect multispectral images for geospatial analysis, providing valuable sensing and computational applications in the harsh space environment, characterized by highly constrained energy and network connectivity. Visual tasks, which detect vehicles along interstate highways to estimate traffic [1], count buildings from key areas to predict population [2], or monitor animals from the wildness to track their behaviors [3], etc, are among the key use cases for EO satellites. Advances in technology enable satellites to collect vast amounts of Earth images daily, often reaching tens of Terabytes [4], [5]. However, traditional satellites operate as "bent-pipe" and typically downlink all raw observations to the ground, which is significantly restricted in transmission capability due to the scarce bandwidth resources (e.g., tens of Mbps) and limited satellite-ground connection duration.

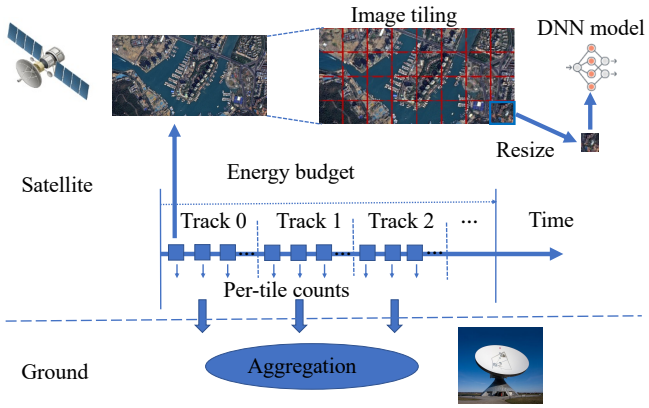


Fig. 1: A satellite periodically captures Earth images. Images are pre-processed into tiles before applying DNN models.

To overcome the limitations of the bent-pipe architecture, Orbital Edge Computing (OEC) has emerged as a promising solution, which mitigates the communication bottleneck by processing the observation data in space and sending the process results to the ground rather than directly downlinking the raw data [6]. However, in-orbit processing faces a computational bottleneck due to the inherent limitations of satellites, including both computational capacity and constrained energy. EO satellite images are large (i.e., hundreds of millions of pixels) and arrive at a high rate, far exceeding what embedded satellite hardware can process. The computational bottleneck arises not only from lower computing power hardware but also from the inability to process all images within the energy budget harvested from the less productive solar panel. For instance, Baoyun satellite can harvest up to 260KJ of energy daily, but not all of it is utilized for computing. When about 150KJ of energy is allocated for computing, satellite can only support the computation of about 22% of the observable high-resolution 3K satellite images. Therefore, as an effort to support analysis for EO, this work focuses on in-orbit detection of Earth objects, considering the inherent limitation

*Corresponding author.
This work is supported by NSFC (No.62032003, 61921003, 62372061, 62302055, 62102045), Beijing Nova Program (No.Z211100002121118), and Young Elite Scientists Sponsorship Program by CAST (No.2021QNRC001).

in computational and energy. Fig. 1 shows that satellites utilize Deep Neural Network (DNN) models in orbit to detect objects. Detection results for images captured on each track are aggregated, and satellites transmit these aggregated counts.

Terrestrial applications typically divide large images into smaller images and process each image on DNN models. However, they focus only on improving application performance, ignoring the vital system factor of computational overhead [7]. Prior attempts to address the computational bottleneck focused on distributing in-orbit processing across a constellation aimed at a particular purpose [6], but the relatively high-cost solutions did not considered the energy budget of each satellite. In-orbit computing still cannot meet onboard application requirements due to energy constraints. Therefore, a collaborative satellite-ground system is needed, where embedded satellite hardware initially processes images under energy budget constraints and transmits crucial images to the ground despite facing limited bandwidth [8]. In situations where a downlink system encounters a scarce link, efficient transmission within a short-term contact time should be prioritized. However, today's satellites transmit data indiscriminately [9]. Less attention has been paid to the scarce downlink bandwidth budgets. Therefore, this work centers on processing images onboard the satellite and selectively transmitting crucial images by utilizing the downlink capability.

There are potential opportunities to deal with these computational and downlink bottlenecks in the collaborative satellite-ground system: (1) By tiling large images into smaller ones and resizing them to fit standard DNN models' input size, there exists an optimal tile size that satisfies higher detection accuracy and lower computational overhead. Preliminary experiments conducted on widely-used datasets reveal that tile size affects both detection accuracy and computing overhead (execution time). Therefore, we propose finding the optimal tile size, balancing accuracy and computational overhead, for executing the onboard DNN model; (2) Optimal tile size also establishes the confidence thresholds for onboard DNN models, influencing the decision of whether to downlink corresponding tiles to the ground. To efficiently utilize the available bandwidth, we implement bandwidth-aware downlinking throttling to dynamically select and downlink crucial tiles within the given bandwidth budget. The selection process will be based on the confidence thresholds set by the onboard DNN models; (3) Satellite images contain some semantically geospatial feature contexts with a high degree of similarity, specifically for the captured images as the satellite passes over its ground track. To alleviate computational and downlink bottlenecks, we introduce a lightweight, clustering-based data deduplication technique that leverages geospatial feature contexts. This technique optimizes data processing by efficiently identifying and removing redundant data, thereby reducing computational and downlink burdens.

We present `TargetFuse`, a collaborative satellite-ground system under the inherent limited computational and downlink constraints. To ensure a realistic simulation, we collect new data, including satellite operation details such as the computing power of embedded hardware, based on tested in-orbit satellites. We provide a comprehensive evaluation of `TargetFuse` across various energy budgets, embedded hardware setups, and bandwidth budgets. Extensive experiments show that `TargetFuse` can reduce detection errors by 3.4× on average, compared to onboard computing. `TargetFuse` showcases a remarkable 9.6× improvement in bandwidth efficiency compared to the vanilla baseline under the limited bandwidth budget constraint. Our contributions can be summarized as follows:

- We design and implement a satellite-ground collaboration system for object detection, a critical use case for EO satellite, aiming to minimize detection errors. To ensure the system's realism, we collect and publish new data that contains satellite operation details*.
- We propose image tiling to balance detection accuracy and computational overhead, clustering-based data deduplication to alleviate the computational bottleneck, and bandwidth-aware down-linking throttling to address downlink bottlenecks.
- We conduct extensive experiments across various energy budgets, embedded hardware configurations, and bandwidth budgets, and demonstrate that `TargetFuse`'s superior performance against four baselines.

II. Background and Motivation

A. EO Satellites

EO satellites collect raw sensor data for geospatial analytics. These satellites capture images along their ground track, generating geospatial images that cover hundreds of square kilometers and contain hundreds of millions of pixels. The level of detail present in these images is described by the ground sample distance (GSD) [6], which is determined by orbit altitude, sensor size, and camera characteristics [6]. Besides, the high velocity of satellites, up to 7.9 km/s, results in brief periods of visible contact with ground stations. These periods typically last only a few minutes, sometimes less than 8 minutes, and may occur infrequently.

During a single orbit revolution, a single satellite captures more images than it is capable of downlinking [10]. This is due to the downlink capacity of current satellite sensors, which is insufficient to support their data rates. Currently, the majority of Earth-observation satellites are organized in a bent-pipe architecture [6], where raw observations are transmitted to ground stations and then processed by machine learning algorithms. However, satellites only achieve downlinking rates of no more than hundreds of Mbps using Ka-band [11], resulting in a limited daily downlink data volume. For instance, given that a contact session lasts for 6 minutes, the system can downlink a maximum data quantity of 4.39 GB at a downlinking speed of 100 Mbps [12]. The downlink bottleneck prevents these daily global observations from being transmitted to the ground. What's more, not all raw observations contain high-value data. Bent-pipe satellites may waste the

*https://www.tiansuan.org.cn/

precious downlink bandwidth as they indiscriminately transmit raw observations independent of the value of data. Statistics show that 67% of the observations are obscured by clouds, thus becoming low-value to users [13].

B. Orbital Edge Computing

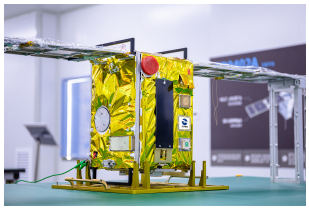
OEC [6] has been proposed to address the downlink bottleneck, in which satellites process raw data in space. OEC aims to address the limitations of "bent-pipe" architectures [6] by distributing processing across a constellation. Nowadays, each satellite can be equipped with hundreds of high-datarate cameras, sensors, and commercial, off-the-shelf (COTS) hardware [14]. The satellite is lightweight, small-sized, and expensive, with each satellite weighing a few kilograms, measuring a few centimeters, and costing millions of USD.

However, satellites still face limitations in providing highly-capable onboard processors. Currently, available space-grade processors are often decades-old, "flight heritage". The satellite systems may operate for decades in the space environment, which means that COTS hardware may be low-risk and highly reliable at the expense of performance. Hence, recent trends in space systems have started to consider COTS-embedded systems [15], which enable the use of in-orbit processing. Applying these terrestrial techniques directly to space is appealing, but computational capacity is subject to unique operating constraints. For instance, unlike on Earth, all energy expended in space must be harvested from solar panels, which is backed by a rechargeable energy buffer [16]. In line with typical 3U Cubesat systems [6], the size of the tested satellite limits the area of the solar panel to 57.2 $\times$ 20.6 cm and thus limits the power to a range of 34-118 W.

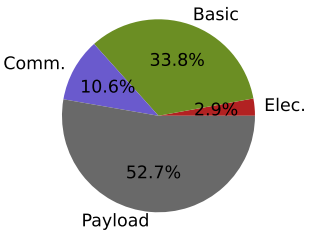
The limited and inelastic computational resources also pose a major space systems challenge. Though the satellites are exposed to sunlight for about 60% of each orbit period (e.g., approximately 90 minutes) [17]. Only 30% of the solar energy is initially converted into battery power, and less than 50% of the battery capacity is utilized for daily satellite operation over its lifetime [13]. We collected real-world data from Baoyun satellite and observed that computing in operation accounts for approximately 50%, compared to other subsystems such as basic and electrical operations, as depicted in Fig. 2. Therefore, during runtime, operating system (OS) of the satellite allocates an energy budget to the computing modules by adjustmenting in input performance and energy parameters.

C. Challenges for EO Computing

EO computing is a crucial application of OEC. However, two challenges arise at the orbital edge: a downlink bottleneck that hinders the transmission of all raw data and a computational bottleneck that restricts the processing of all data in orbit. OEC has been proposed to address computational needs by distributing tasks across a constellation. Although effective in reducing per-satellite compute workload to meet full ground track coverage, this approach is in nature designed for vertically-integrated constellations for a single purpose, which requires a large pipeline population and incurs high monetary



(a) Physical map of Baoyun satellite [9], [18].



(b) The distribution of energy expenditure in satellite operation.

Fig. 2: The real-world data, including energy and computing power, is collected from Baoyun satellite.

costs. Existing OEC work can hardly reduce the per-satellite workload without increasing the constellation population [6]; this shortcoming is a key motivation for our work.

Processing raw observations in space presents significant challenges due to limited computational resources. Satellites have an energy limit that prevents them from processing all images, which creates a computational bottleneck that limits OEC's ability to address the downlink bottleneck. Unfortunately, improving the computational capacity of COTS hardware is difficult because of physical constraints. And there are no feasible options for adjusting the computational capacity of the hardware already in space. Furthermore, in-orbit computing alone is insufficient because each satellite has natural constraints such as volume, mass, and energy that prevent it from processing all images.

Compared to the space environment, a more collaborative approach between satellite and ground station is feasible by utilizing the relatively more favorable computing capacity and higher energy availability of ground stations. As a result, we consider the two bottlenecks of in-orbit processing jointly to adapt the computing hardware of the target satellite in space.

III. System Design

This work introduces a satellite-ground collaborative system tailored to counting, aiming to minimize counting errors in the challenging satellite environment. `TargetFuse` executes a geospatial counter (e.g., a shallower DNN) in space with lower computing power, producing less accurate counts; and a ground counter (e.g., a deeper DNN) on the ground with higher computing power, producing more accurate counts. To meet the computational and downlinking needs of counting applications, `TargetFuse` leverages three techniques: adaptive image tiling, clustering-based data deduplication, and bandwidth-aware downlinking throttling. The workflow of `TargetFuse` is shown in Fig. 3. For each satellite image, `TargetFuse` divides it into several tiles based on the image resolution and input size of the DNN counters. Next, `TargetFuse` automatically performs clustering-based data deduplication considering the similarity of the tiles. After that, `TargetFuse` applies selection logic to determine which logic according to confidence thresholds from the onboard DNN counter. Note that onboard system also downlinks the

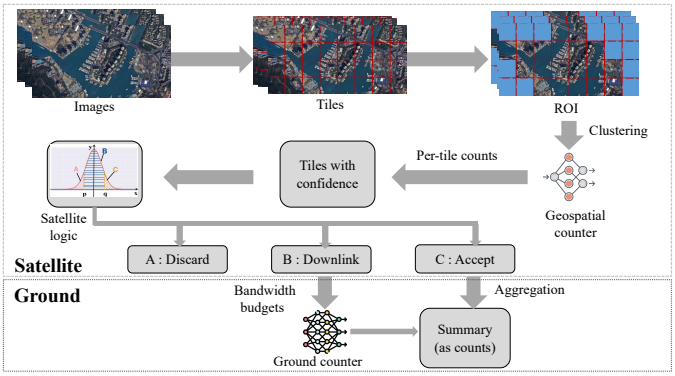
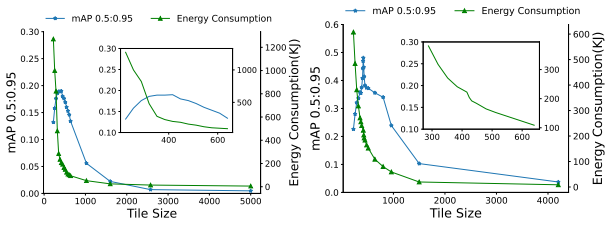


Fig. 3: The overall workflow of TargetFuse.



(a) xView [23] (b) UAVOD10 [2]

Fig. 4: Effect of varied tile size on mAP accuracy and execution time of yoloV3-tiny model. mAP 0.5:0.95 is the average mAP over IoU threshold [24].

counting result in space with vital tiles. Hence TargetFuse emits the aggregated object count across space and ground.

A. System Operation

1) Energy expenditure: TargetFuse performs orbital counting while adhering to the allocated energy budget of satellites along their trajectories. Utilizing data obtained from an in-orbit satellite [9], energy allocation extends beyond fundamental satellite operations, including propulsion and avionics. Energy is allocated for the following computing activities associated with counting: (1) E_{cap} for capturing images; (2) E_{com} for executing counting on images; (3) E_{agg} for deriving aggregated counting results in space; (4) E_{down} for downlinking the satellite images to be counted on the ground. The most energy-intensive activities (2) and (4) account for over 60% of the total energy consumption, as illustrated in Fig. 2: during each orbital track, satellites perform several trillions of FLOPs and downlink some images to the ground. In contrast, activities (1) and (3) consume negligible energy: (1) only involves capturing thousands of images from the onboard camera, and (3) only involves a few hundred arithmetic operations. Therefore, activities (2) and (4) are the focus of this work [19]. These two activities align with the satellites lifetime design [9], [20], which optimally utilizes less than 50% of the available battery energy. This efficiency allows for the estimation of the daily energy budget.

2) Selection logic with different confidence thresholds.: Prior to system execution, the satellite's OS captures numerous images and selects a DNN counter. The deployment of DNN counters on satellites has become increasingly crucial for guaranteeing counting accuracy. When selecting a counter, a confidence threshold is established based on the onboard satellite's DNN counter detection. This confidence threshold indicates the probability of accurately counting the objects and falls within the range [0,1] [21].

3) Objective: Minimizing overall counting error while optimizing energy and bandwidth expenditure.: TargetFuse's objective is to minimize overall counting errors by allocating energy and bandwidth efficiently. In our implementation, the overall counting error is defined as the mean of the counts across all tiles, a widely employed metric in diverse applications [22]. A smaller counting error indicates heightened confidence in counting accuracy, which ultimately translates to greater benefits for customers. To address the computational and downlink bottlenecks in the counting application, TargetFuse leverages the three following techniques.

B. Adaptive Image Tiling

For each image, TargetFuse is designed to divide images into tiles with a comparatively lower execution overhead. Processing large satellite images, typically containing thousands of megapixels, through the utilization of DNN models in space is an essential solution. However, executing standard models directly on these satellite images may lead to excessive memory and potentially exhausting the available memory. This is particularly challenging in typical space environments where the memory capacity is insufficient for handling such large-scale images. Prior work [6] divided the image into several tiles maximizing inference accuracy at the expense of execution overhead. Additionally, downsampling to the input size of standard model architectures is frequently insufficient for achieving optimal performance [7].

A large image can be segmented into either a larger tile size with fewer tiles or a smaller tile size with numerous tiles. After scaling each tile into input size of DNN counter, the execution time per tile remains constant. Theoretically, opting for a larger tile size reduces image processing time, as each tile undergoes less degradation. Conversely, selecting a smaller tile size increases processing time, and each tile also undergoes less degradation. To explore the impact of tile size on both inference accuracy and execution overhead, we conducted measurements on two datasets, as shown in Fig. 4. Interestingly, both datasets display similar curves. As the tile size increases, the execution time decreases due to a smaller number of tiles per image. However, there is an optimal tile size that maximizes accuracy. Accuracy tends to deteriorate when the tile size deviates from this optimal size. This observation aligns with findings from previous studies [6], [10]. The optimal tile size enables substantial improvements in accuracy while maintaining acceptable time. Moreover, the optimal tile size is not constant but varies depending on the DNN counters and image input size. Consequently, we determine the optimal tile size based on the combination of satellite image and DNN counter. Considering the trade-off between geospatial analysis

Algorithm 1: Optimal Tile Size Selection

Input: minimum size s_{left}, maximum size s_{right}, threshold ε, $mAP_{s_{left}}$, $mAP_{s_{right}}$
Output: optimal size s_{best}

$s_{left} \leftarrow s_{min}$; $s_{right} \leftarrow s_{max}$;
while $s_{right} - s_{left} > \varepsilon$
  $s_{midl} \leftarrow s_{left} + (s_{right} - s_{left})/3$;
  $s_{midr} \leftarrow s_{right} - (s_{right} - s_{left})/3$;
  if $mAP_{s_{left}} < mAP_{s_{right}}$
    $s_{left} \leftarrow s_{midl}$;
  else
    $s_{right} \leftarrow s_{midr}$;
$s_{best} = (s_{left} + s_{right})/2$;
Return s_{best}

accuracy (i.e., mAP accuracy) and execution overhead (i.e., processing time), we aim to identify an optimal tile size that aligns with the input size of the DNN counter.

We present a detailed approach for optimizing image size in tile-based processing using Algorithm 1. The process initiates by initializing the tile sizes. Drawing from the measurement results before NN counter deployment, we promptly narrow down the search interval and empirically establish the minimum and maximum tile sizes. We then iterate until the first image size meets the preset empirical size difference threshold. Specifically, we divide the search interval into three equal fractions and compare the mAP accuracy at different sizes to identify optimal search intervals. The optimal tile size lies in the interval $[s_{midl}, s_{right}]$ when $mAP_{s_{left}} < mAP_{s_{right}}$, and vice versa. Finally, we obtain an approximate optimal tile size by taking the midpoint of the interval. This method achieves a balance between accuracy and computational efficiency, providing a user-friendly solution with improved speed and accuracy of counting application.

C. Clustering-based Data Deduplication

`TargetFuse` is tasked with classifying each tile into geographic contexts while complying with the computational constraints imposed by satellite hardware. A geographic context refers to a subset of images characterized by a high degree of similarity, along with geographic and transformation features. It is common for these images to exhibit a substantial degree of similarity, often remaining relatively static over time. EO satellites periodically pass over identical locations on Earth's surface, capturing images that exhibit significant similarity or near-identical characteristics at different times along their orbital path [25]. As shown in Fig. 5, the two tiles acquired after tiling include an identical number of similar images. This is due to the short average revisit cycle of each satellite, such as GF-3, which revisits the same area at least twice every day, enabling the capture of the same geographical area multiple times [26].

The presence of numerous semantically similar images exerts substantial pressure on the limited computational capacity within a satellite. This heightened demand for processing images may pose a computational challenge. To tackle this challenge, we propose a data deduplication strategy that involves processing representative tiles based on geographic context, rather than processing all similar tiles. Certain images are computationally less demanding in specific contexts than in others. Due to the high predictability of satellite orbits, determining the contexts can be readily achieved. However, there may be image tiles for which the contexts are not immediately apparent, posing a challenge in their generation.

(a) (b)

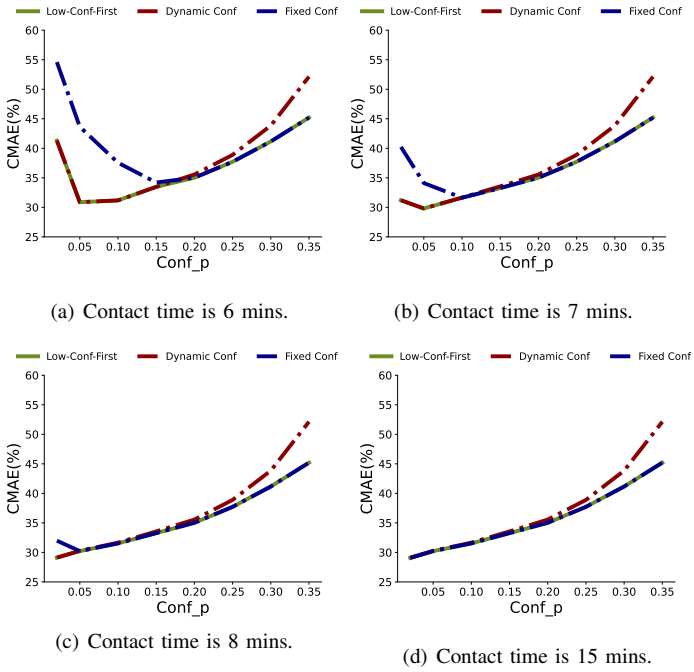
Fig. 5: The tiles generated through image tiling in the DOTA [28] dataset we utilize contain several images with similarities.

The technique efficiently generates contexts for image tiles by dividing them into multiple contexts. To cluster the representative image tiles based on similarity, the technique utilizes a low-dimensional label vector indicating the geographic features described by computing moments [27] present in each image tile. The technique creates a set of contexts by performing k-means clustering while exploring the Euclidean distance of the label vectors to measure similarity. Moreover, to enhance the DNN counter's robustness to diverse sensors, we consider geographic label transformations like translations and rotations as objects may have arbitrary headings between 0 and 360 degrees. We also explore a range of cluster counts when partitioning the dataset into several clusters. Further investigation of this hyperparameter space represents an exciting avenue for future research.

D. Bandwidth-aware downlinking throttling

The downlinking bottleneck also poses a significant challenge in in-orbit counting, impacting the overall performance. When receiving the tiles with confidence thresholds, `TargetFuse` employs selection logic to determine the policy for handling these tiles. The optimal policy ensures that downlinking remains within the bandwidth budget constraint while transmitting as many tiles as possible to the ground to minimize counting errors.

The selection logic, based on the confidence threshold from the space-based DNN counter, is categorized into three groups (in Fig. 3): when the confidence threshold is relatively smaller (i.e., $< conf_p$), `TargetFuse` discards them directly; when confidence threshold is large enough (i.e., $> conf_q$), `TargetFuse` accepts the counting result; only when confidence threshold is between $conf_p$ and $conf_q$ (i.e.,



(a) Contact time is 6 mins.

(b) Contact time is 7 mins.

(c) Contact time is 8 mins.

(d) Contact time is 15 mins.

Fig. 6: A comparison of *CMAE* (in Section IV) in three methods based on the choice of confidence thresholds when downlinking in different contact times.

$[conf_p, conf_q]$), `TargetFuse` downlinks the tiles and processes them on the ground DNN counter. To comprehensively explore how the confidence threshold affects *CMAE* (i.e., the mean difference between the estimated count and the ground truth), we vary the confidence threshold $conf_p$ under different contact times between satellite and ground (i.e., different downlinking data volume). As the tiles are sorted by the confidence threshold, and the objective is to downlink as many tiles as possible, we must consider the following three methods based on the choice of confidence thresholds when downlinking the tiles within $[conf_p, conf_q]$ under the limited bandwidth budget constraint:

- *Low-Conf-First:* If there are still tiles remaining when the bandwidth is exhausted, we proceed to directly count these tiles and downlink the results.
- *Fixed Conf:* If there are still tiles remaining when the bandwidth is exhausted, we only count tiles whose confidence thresholds are higher than the fixed $conf_q$.
- *Dynamic Conf:* We first count the tiles with confidence thresholds higher than the preset $conf_q$ and then count the tiles whose confidence thresholds are within $[conf_p, conf_q]$ until the bandwidth is exhausted. The value of $conf_q$ varies depending on the downlinking constraint.

The observations from Fig. 6 are summarized as follows:

- **Dynamic $conf_p$ leads to performance improvement.** When the downlink capacity is insufficient, both *Low-Conf-First* and *Dynamic Conf* show similar performance, as all remaining tiles are counted and *Dynamic Conf* is not effective. In this scenario, a larger $conf_p$ leads to a lower *CMAE*, as it facilitates the downlinking of more high-confidence tiles. Moreover, when downlink capacity is sufficient, both *Fixed Conf* and *Dynamic Conf* exhibit comparable performance, as all tiles are counted. Consequently, *Fixed Conf* is not enabled. In this case, a larger $conf_p$ results in higher *CMAE*, as it discards the high-value tiles with confidence thresholds below $conf_p$. Therefore, choosing an appropriate dynamic $conf_p$ is crucial for improving counting performance. It motivates to downlink high-confidence images first and downlink some low-confidence images within the available bandwidth.
- **Optimal $conf_p$ improves performance.** When downlink capability is sufficient and $conf_p$ increases to a certain value, all methods discard the tiles below $conf_p$, resulting in a counting error. Here both *Low-Conf-First* and *Fixed Conf* exhibit identical performance, since *Fixed Conf* fails to work effectively. Additionally, *Dynamic Conf* incurs a higher counting error, as counting tiles larger than $conf_q$ in space is not as accurate as on the ground. Therefore, optimizing $conf_p$ is crucial.
- **$conf_q$ alleviates the downlink constraint without greatly affecting performance.** In Fig. 6(d), when the downlink capability is sufficient (i.e., dynamic $conf_q$ does not change), and the initial $conf_q$ is not too large (i.e., smaller than 0.2), *Dynamic Conf* and *Low-Conf-First* show similar performance. However, *Dynamic Conf* has the advantage of increasing the downlink volume compared to *Low-Conf-First*.

Therefore, strategically selecting the optimal confidence threshold is crucial to downlink more high confidence tiles. However, we face the challenge of determining the downlinking method for confidence thresholds and fully exploiting the bandwidth budget. Algorithm 2 describes the bandwidth-aware downlinking throttling procedure, which takes the bandwidth requirement and tiles obtained by tiling and clustering (as shown in Fig. 3) as input and produces a set of tiles that can be downlinked by the scarce bandwidth constraint. We first identify all the clustered tile sets on the satellite and discard tiles with a confidence threshold lower than a specified empirically $conf_p$ (lines 5-6). Tiles with high confidence $conf_q$ are included directly in the C_{space} (lines 7-8). We calculate the current remainder bandwidth and maximize it to downlink images (lines 13-18), In other words, the remaining tiles are sorted based on data size in descending order, and the count results are added to the transmitted tile set S_{trans} if sufficient available bandwidth is present.

IV. EVALUATION

A. Methodology

1) Heterogeneous computational hardware: A computational satellite enhances onboard sensing, communications, and control operations. In this context, the tested satellites are equipped with two industrial modules: Raspberry Pi 4B

Algorithm 2: Bandwidth-aware downlinking throttling

Input: minimum confidence $conf_p$, maximum confidence $conf_q$, bandwidth requirement $Band_{max}$, filtered ROI set $tiles$

Output: transmitted tile set S_{trans}, count from space C_{space}

Initialize S_{trans}, C_{space}, $conf_{tiles}$;
$Band_{rest} \leftarrow Band_{max}$;// Remaining bandwidth assignment
foreach $tile$ ***in*** $tiles$ **do**
    $scores, labels \leftarrow Geospatial\ DNN\ Counter(tile)$;
    if $scores.mean() < conf_p$
        $continue$;
    if $scores.mean() > conf_q$
        $C_{space} += C_{tile}$;
    else
        $conf_{tiles}.add()$;
end
$conf_{tiles}.sort()$;// Sort by confidence score
foreach $tile$ ***in*** $conf_{tiles}$ **do**
    $Size \leftarrow tile.size()$;// Measure size of tile
    if $Band_{rest}.size() \geq Size$
        $S_{trans}.add()$;
    else
        $break$;
end
Return S_{trans}, C_{space}.

Datasets	Size	GSD (m)	Volume (GB)
xView	3000	0.3	20
DOTA	4000	0.1∼0.81	34.3
UAVOD10	1000∼4800	0.15	0.9

TABLE I: Datasets in the evaluation. Size: large-scale geospatial image resolution. GSD: geographic distance between adjacent pixels.

(RPI4) and Atlas 200 DK (Atlas). These modules are preferred for their cost-effectiveness and programming simplicity. This work focuses on characterizing onboard computing utilizing these two modules, recognized as widely-used computational hardware in satellite applications.

2) ROI-based instance selection: To address redundant counting in small tiles containing invalid information, such as backgrounda, recognized challenge in spatial vision, we utilize established techniques for regions of interest (ROI) [29], as depicted in Fig. 3. We also implement non-maximal suppression to the global matrix of bounding box predictions to alleviate overlapping detections [30]. The operation is compatible with this work, processing in-orbit images to save bandwidth and aligning seamlessly with our core contributions in producing statistical counting results.

3) Experiment Setups: Theoretical satellite-to-ground bandwidth can achieve 100 Mbps, matching the air interface transmission rate [12]. However, real-world measured satellite-ground bandwidth is limited to 30-50 Mbps due to reception losses. Additionally, the default satellite-ground contact time is set at 6 minutes, and the default DNN counters on the satellite and ground are YOLOV3-tiny and YOLOV3, respectively.

DNN Counters	Input	mAP
`YOLOV3` (Ground)	416*416	55.3
`YOLOV3`-tiny	416*416	33.1
ssd mobilenetv2	200*300	22.0

TABLE II: The DNN counters architecture used in this work. mAP: mAP accuracy on COCO dataset [24].

4) Datasets: We utilize three geospatial datasets covering various scenes, including buildings, planes, and tracks (in Table I), for evaluation. Each satellite captures images along its ground track, and although object counts may exhibit high temporal correlation, there are currently no publicly available datasets covering multiple days and providing diverse object contexts with sufficient instances. To address this limitation, we employed rotations and data augmentation. We simulated image captures as the satellite passed over its ground track by flipping and rotating 50% of the images in the dataset.

5) DNN counters and ground truth counts: In Table II, we present publicly available DNN counters customized for each data sample, primarily small or medium-sized to accommodate the resource-constrained nature of satellites. The evaluation is conducted based on the ground truth returned by the dataset.

6) Metrics: To quantify the performance of the counting system, we use *Count Mean Absolute Error (CMAE)*, defined as $\sum|y_i - g_i| / \sum g_i$, where y_i and g_i are our counter count and true count, respectively. The metric indicates the extent to which our approximate counts deviate from the ground truth, and a narrower *CMAE* is considered better. In addition, we leverage the data size to demonstrate changes in the volume of data, offering an indirect reflection of bandwidth utilization.

7) Baselines: We compare `TargetFuse` to four baseline methods: (1) *Space-Only:* All images are processed using the onboard DNN counter, and the resulting counting data is transmitted to the ground; (2) *Ground-Only [6]:* Satellites operate as "bent-pipe", collecting images and downlinking them to the ground; Due to bandwidth limitations, the approach is to downlink as many observations as possible within the limited contact time; (3) *TIANSUAN [9]:* Following onboard counting, satellites exclusively transmit results with confidence thresholds surpassing a predetermined empirical threshold, downlinking the remaining data to the ground within the constrained contact time; (4) *Kodan [10]:* The images processed in orbit are categorized into different levels, with a preference for transmitting high-value pictures. Note that while we share similarities with *Kodan* in onboard processing, there is a distinction in transmitting — *Kodan* does not consider bandwidth limitations. Theoretically, the outcome obtained from *Kodan* functions as an upper bound.

B. End-to-End performance

1) `TargetFuse` enhances performance: Fig. 7 illustrates the experimental results with the varying bandwidths. In all methods, except for *Space-Only*, *CMAE* decreases as

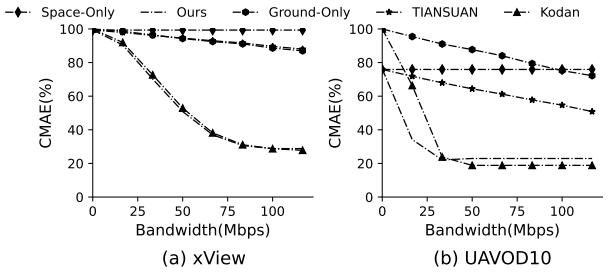


Fig. 7: The performance of *CMAE* on the varying satellite-ground bandwidth.

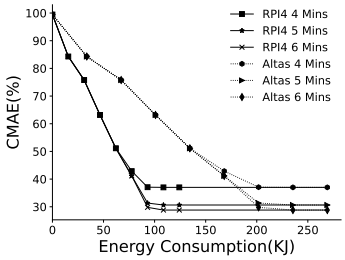


Fig. 8: `TargetFuse` operates on COTS hardware with varying computational energy budgets, which is a fraction of the total solar energy available.

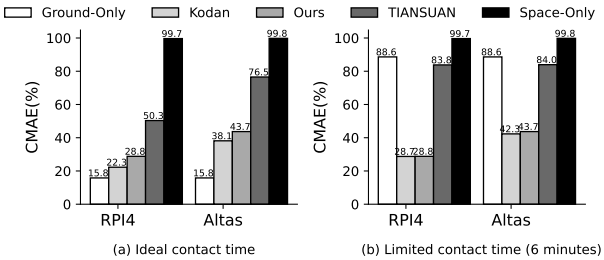


Fig. 9: A comparison of *CMAE* on different hardware with different contact times (dataset: xView, energy budget: 150KJ/day, onboard counter: `YOLOV3`-tiny).

more bandwidth resources are utilized and more images are downlinked. `TargetFuse` outperforms the vanilla baseline TIANSUAN, which relies on a fixed confidence threshold, by reducing the counting error by 1.93× and 2.51× on average across different datasets. This improvement is achieved because `TargetFuse` intelligently selects representative and high-value images for downlinking, leading to enhanced counting performance. Additionally, `TargetFuse` achieves a 9.6× bandwidth-efficient improvement compared to TIANSUAN under limited bandwidth constraints.

Counting performance is subject to the characteristics and composition of the dataset used for evaluation. Despite variations in the datasets, `TargetFuse` shows similar performance to *Kodan*, with differences mainly attributable to the downlinking. The key advantage of our system over *Kodan* lies in its incorporation of bandwidth-aware downlinking throttling. In scenarios with extremely scarce bandwidth, especially close to real-world values (i.e., below 50 Mbps in Fig. 7(b)), `TargetFuse` enables efficient bandwidth allocation and directing the remaining resources to other applications.

2) `TargetFuse` provides effective counting with limited energy constraints: Fig. 8 explores the impact of computational energy on the performance of `TargetFuse`, considering various hardware configurations and contact times. Each satellite is theoretically restricted to a daily energy collection of up to 260KJ, allocated to computing operations within a specified energy budget. Results show that, under identical computational energy constraints, longer contact time leads to lower *CMAE* as the increased downlinking of images to the ground. Additionally, both hardware setups achieve comparable CMAE values within the same contact time. However, RPI4 (with 6W power) featuring computing-limited hardware outperforms Atlas (with 13W power) by saving approximately 50% of energy. This advantage stems from the RPI4's ability to efficiently process more data, thus minimizing the *CMAE*.

3) Low-power hardware also improves performance: The performance of `TargetFuse` is relevant with accelerators running DNN counters, as shown in Fig. 9. Compared to Altas, RPI4 significantly reduces *CMAE* by 34%, primarily attributed to its ability to process and downlink more images per track. Specifically, for the same *CMAE*, RPI4 requires a shorter contact time, resulting in less downlinked data; with the same contact time, RPI4 can achieve a lower *CMAE*.

4) Exploiting diverse DNN counters: Fig. 10 illustrates the counting performance of different DNN counters under the same energy constraint. Despite the diverse choices of counters on the satellite, their performance remains similar. Both `TargetFuse` and *Kodan* exhibit very comparable performance, as both methods select less accurate DNN counters in space, with the ground counter providing more accurate counts. This observation highlights the significance of adaptive image tiling, as it enables different selection logics for each tile based on confidence thresholds. Execution with a smaller tile size (e.g., ssd mobilenetv2) experiences less degradation, resulting in a narrower *CMAE*.

5) Exploiting diverse datasets: Fig. 11 shows the counting performance on widely-used datasets, comparing the baselines with `TargetFuse` under unlimited downlinking to the ground. The performance of *TIANSUAN* consistently varies across different datasets, as this method depends on fixed empirical confidence thresholds, thereby affecting counting performance. `TargetFuse` reduces counting error by 3.4× on average, compared to *Space-Only*. With *Ground-Only*, computational constraints are further mitigated and ground counters can process all tiles with higher precision, resulting in the theoretically lowest achievable *CMAE*. *Kodan* and `TargetFuse` implement ROI-based instance selection, image deduplication, and onboard computing. However, these methods lead to a comparable performance with *Ground-Only*.

C. Validation of Key Designs

1) Clustering-based data deduplication enhances bandwidth efficiency: In Fig. 12(a), performing *Clustering* leads to a reduction in downlink data. Compared to *No-Clustering*, the

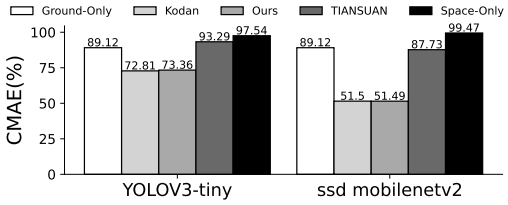


Fig. 10: A comparison of counting performance on various onboard DNN counters under the limited bandwidth (50 Mbps).

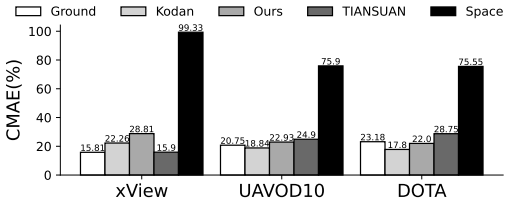


Fig. 11: A comparison of counting performance on three datasets under the unlimited downlinking.

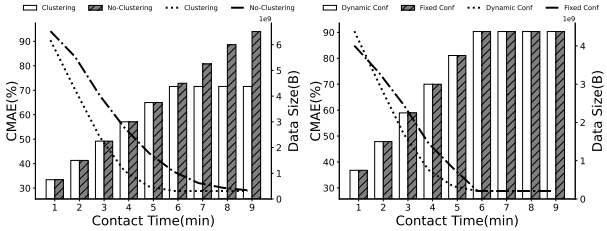
downlink volume in the *Clustering* scenario is approximately 5.6% less, accounting for 32.8% of the volume observed in *No-Clustering*. This is primarily due to the fact that, *Clustering* transmits only representative similar or duplicated tiles to the ground after the clustering process. Additionally, within the downlinking constraints, *Clustering* enables the transmission of additional small tiles to the ground, which can be integrated into the final counting result.

2) Bandwidth-aware downlinking throttling improves performance: Fig. 12(b) demonstrates that *Dynamic Conf* outperforms *Fixed Conf* as the contact time increases. When contact time is limited, *Dynamic Conf* can selectively downlink high-confidence tiles, reducing the counting error compared to the indiscriminate downlinking of tiles in *Fixed Conf*. Both *Dynamic Conf* and *Fixed Conf* exhibit similar performance when all the tiles can be downlinked.

V. Related Work

Satellite networking has witnessed substantial growth, with a predominant focus in research on inter-satellite networking [31], [32]. However, the challenges posed by the scarce downlink bandwidth and the associated bottleneck are also crucial. OEC [6] focuses on the downlink bottleneck and shifts it to the inelastic computation capacities. Another recent work, Kodan [10] has proposed filtering low-value data and prioritizing high-value data for downlinking to mitigate the downlink bottleneck. However, Kodan considers the constraints of scarce satellite-ground bandwidth, a limitation that we aim to address in this work. This work not only tackles the downlink bottleneck through bandwidth-aware downlinking throttling but also addresses the computational bottlenecks.

The computational bottleneck represents a major challenge for satellite systems. Some works, such as [33] and [34], have explored the viability of utilizing DNN models for in-orbit processing. However, these approaches do not directly tackle the specific challenges addressed by `TargetFuse`, such as



(a) Data Deduplication. (b) Downlinking throttling.

Fig. 12: Ablation study on key designs.

operating with real-world energy budgets. Moreover, several works focus on optimizing DNN models for accuracy or speed in terrestrial applications [35]–[41]. Various terrestrial and embedded systems that operate on harvested energy [42], [43] can transmit data at any time within energy constraints. However, this continuous data transmission capability is impractical for satellites as they can only transmit data when they are in proximity to ground stations. Moreover, the limited bandwidth available for satellite communication is significantly smaller than that of ground-based connections.

Vision tasks in EO satellites have been extensively studied and proven valuable for scientific investigations [44]–[46]. These applications span various domains, including computer systems, satellite systems, satellite networks, and machine learning systems. Developing a comprehensive scheduling system that considers image size, processing speed, energy constraints, and orbital mechanics poses a challenging research problem in computer systems. We are actively working to resolve the computer systems design challenges associated with satellite computing under real-world satellite constraints. Our system leverages a tradeoff between accuracy and execution time, and effectively addresses downlink bottleneck by bandwidth-aware downlinking throttling.

VI. Conclusion

This work introduces an analytics system tailored for addressing object counting queries on EO satellites. `TargetFuse` utilizes both less accurate in space and more accurate ground-based DNN models to determine earth object counts within the constraints of computation and communication. `TargetFuse` is designed to minimize counting errors under energy and bandwidth constraints. Extensive experiments show that `TargetFuse` can reduce counting error by 3.4× on average, compared to onboard computing.

References

[1] X. Liu, Z. Wang, J. Feng, and H. Xi, "Highway vehicle counting in compressed domain," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3016–3024.

[2] Z. Xiong, F. Zhang, Y. Wang, Y. Shi, and X. Zhu, "Earthnets: Empowering ai in earth observation," *arXiv preprint arXiv:2210.04936*, 2022.

[3] J. C. Hodgson, S. M. Baylis, R. Mott, A. Herrod, and R. H. Clarke, "Precision wildlife monitoring using unmanned aerial vehicles," *Scientific reports*, vol. 6, no. 1, pp. 1–7, 2016.

[4] H. K. Ramapriyan. (2015) The Role and Evolution of NASA's Earth Science Data Systems. NASA. [Online]. Available: https://ntrs.nasa.gov/citations/20150018076

[5] D. Mohney. (2020) Terabytes From Space: Satellite Imaging is Filling Data Centers. Data Centers Frontier. [Online]. Available: https://www.datacenterfrontier.com/internet-of-things/article/11429032/

[6] B. Denby and B. Lucia, "Orbital edge computing: Nanosatellite constellations as a new class of computer system," in *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, 2020, pp. 939–954.

[7] A. Van Etten, "You only look twice: Rapid multi-scale object detection in satellite imagery," *arXiv preprint arXiv:1805.09512*, 2018.

[8] H. Desai and B. Lucia, "A power-aware heterogeneous architecture scaling model for energy-harvesting computers," *IEEE Computer Architecture Letters*, vol. 19, no. 1, pp. 68–71, 2020.

[9] S. Wang, Q. Zhang, R. Xing, F. Qi, and M. Xu, "The first verification test of space-ground collaborative intelligence via cloud-native satellites," *arXiv preprint arXiv:2311.06078*, 2023.

[10] B. Denby, K. Chintalapudi, R. Chandra, B. Lucia, and S. Noghabi, "Kodan: Addressing the computational bottleneck in space," in *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, 2023, pp. 392–403.

[11] S. Caldwell. (2022) SOTA of Small Spacecraft Technology. NASA. [Online]. Available: https://www.nasa.gov/smallsat-institute/sst-soa

[12] (2022) The Downlink rate of satelllite-ground. [Online]. Available: https://www.51cto.com/article/704582.html

[13] Q. Li, S. Wang, X. Ma, A. Zhou, and F. Yang, "Towards sustainable satellite edge computing," in *2021 IEEE International Conference on Edge Computing (EDGE)*. IEEE, 2021, pp. 1–8.

[14] L. Leung, V. Beukelaers, S. Chesi, H. Yoon, D. Walker, and J. Egbert, "Adcs at scale: Calibrating and monitoring the dove constellation," 2018.

[15] T. M. Lovelly and A. D. George, "Comparative analysis of present and future space-grade processors with device metrics," *Journal of aerospace information systems*, vol. 14, no. 3, pp. 184–197, 2017.

[16] Y. Yang, L. Feng, X. Que, F. Zhou, and W. Li, "Energy-and quality-aware task offloading for webvr service in terminal-aided mobile edge network," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 8, pp. 8825–8838, 2022.

[17] F. Davoli, C. Kourogiorgas, M. Marchese, A. Panagopoulos, and F. Patrone, "Small satellites and cubesats: Survey of structures, architectures, and protocols," *International Journal of Satellite Communications and Networking*, vol. 37, no. 4, pp. 343–359, 2019.

[18] (2023) Tiansuan Constellation. [Online]. Available: http://www.tiansuan.org.cn/sate-t1.html

[19] R. Xing, M. Xu, A. Zhou, Q. Li, Y. Zhang, F. Qian, and S. Wan, "Deciphering the enigma of satellite computing with cots devices: Measurement and analysis," *arXiv preprint arXiv:2401.03435*, 2024.

[20] Y. Chen, Q. Zhang, Y. Zhang, X. Ma, and A. Zhou, "Energy and time-aware inference offloading for dnn-based applications in leo satellites," in *2023 IEEE 31st International Conference on Network Protocols (ICNP)*. IEEE, 2023, pp. 1–6.

[21] S. Wenkel, K. Alhazmi, T. Liiv, S. Alrshoud, and M. Simon, "Confidence score: The forgotten dimension of object detection performance evaluation," *Sensors*, vol. 21, no. 13, p. 4350, 2021.

[22] N. Agrawal and A. Vulimiri, "Low-latency analytics on colossal data streams with summarystore," in *Proceedings of the 26th Symposium on Operating Systems Principles*, 2017, pp. 647–664.

[23] D. Lam, R. Kuzma, K. McGee, S. Dooley, M. Laielli, M. Klaric, Y. Bulatov, and B. McCord, "xview: Objects in context in overhead imagery," *arXiv preprint arXiv:1802.07856*, 2018.

[24] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[25] D. P. Roy, M. A. Wulder, T. R. Loveland, C. E. Woodcock, R. G. Allen, M. C. Anderson, D. Helder, J. R. Irons, D. M. Johnson, R. Kennedy *et al.*, "Landsat-8: Science and product vision for terrestrial global change research," *Remote sensing of Environment*, vol. 145, pp. 154–172, 2014.

[26] (2019) The introduction of satelllite. [Online]. Available: http://www.hensat.org.cn/web/gf-03/

[27] N. Keen, "Color moments," *School of informatics, University of Edinburgh*, pp. 3–6, 2005.

[28] J. Ding, N. Xue, G.-S. Xia, X. Bai, W. Yang, M. Yang, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "Object detection in aerial images: A large-scale benchmark and challenges," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[29] (2019) Vehicle prediction using tensorflow object counting API. [Online]. Available: https://www.ucsusa.org/resources/satellite-database

[30] O. Bailo, F. Rameau, K. Joo, J. Park, O. Bogdan, and I. S. Kweon, "Efficient adaptive non-maximal suppression algorithms for homogeneous spatial keypoint distribution," *Pattern Recognition Letters*, vol. 106, pp. 53–60, 2018.

[31] M. Handley, "Delay is not an option: Low latency routing in space," in *Proceedings of the 17th ACM Workshop on Hot Topics in Networks*, 2018, pp. 85–91.

[32] T. Klenze, G. Giuliari, C. Pappas, A. Perrig, and D. Basin, "Networking in heaven as on earth," in *Proceedings of the 17th ACM Workshop on Hot Topics in Networks*, 2018, pp. 22–28.

[33] G. Giuffrida, L. Diana, F. de Gioia, G. Benelli, G. Meoni, M. Donati, and L. Fanucci, "Cloudscout: a deep neural network for on-board cloud detection on hyperspectral images," *Remote Sensing*, vol. 12, no. 14, p. 2205, 2020.

[34] G. Giuffrida, L. Fanucci, G. Meoni, M. Batič, L. Buckley, A. Dunne, C. van Dijk, M. Esposito, J. Hefele, N. Vercruyssen *et al.*, "The ϕ-sat-1 mission: the first on-board deep neural network demonstrator for satellite earth observation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2021.

[35] M. Xu, M. Zhu, Y. Liu, F. X. Lin, and X. Liu, "Deepcache: Principled cache for mobile deep vision," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018, pp. 129–144.

[36] Q. Zhang, X. Li, X. Che, X. Ma, A. Zhou, M. Xu, S. Wang, Y. Ma, and X. Liu, "A comprehensive benchmark of deep learning libraries on mobile devices," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 3298–3307.

[37] M. Xu, T. Xu, Y. Liu, F. X. Lin, and E. Purdue, "Video analytics with zero-streaming cameras." in *USENIX Annual Technical Conference*, 2021, pp. 459–472.

[38] J. Cao, R. Hadidi, J. Arulraj, and H. Kim, "Thia: Accelerating video analytics using early inference and fine-grained query planning," *arXiv preprint arXiv:2102.08481*, 2021.

[39] Q. Zhang, X. Che, Y. Chen, X. Ma, M. Xu, S. Dustdar, X. Liu, and S. Wang, "A comprehensive deep learning library benchmark and optimal library selection," *IEEE Transactions on Mobile Computing*, 2023.

[40] K. Zhou, C. Chen, Y. Ma, Z. Leng, H. P. Shum, F. W. Li, and X. Liang, "A mixed reality training system for hand-object interaction in simulated microgravity environments," in *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2023, pp. 167–176.

[41] K. Zhou, R. Cai, Y. Ma, Q. Tan, X. Wang, J. Li, H. P. Shum, F. W. Li, S. Jin, and X. Liang, "A video-based augmented reality system for human-in-the-loop muscle strength assessment of juvenile dermatomyositis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 5, pp. 2456–2466, 2023.

[42] M. Xu, X. Zhang, Y. Liu, G. Huang, X. Liu, and F. X. Lin, "Approximate query service on autonomous iot cameras," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, 2020, pp. 191–205.

[43] M. Nardello, H. Desai, D. Brunelli, and B. Lucia, "Camaroptera: A batteryless long-range remote visual sensing system," in *Proceedings of the 7th International Workshop on Energy Harvesting & Energy-Neutral Sensing Systems*, 2019, pp. 8–14.

[44] X. Jiang, Z. Xiao, B. Zhang, X. Zhen, X. Cao, D. Doermann, and L. Shao, "Crowd counting and density estimation by trellis encoder-decoder networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6133–6142.

[45] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[46] P. Chattopadhyay, R. Vedantam, R. R. Selvaraju, D. Batra, and D. Parikh, "Counting everyday objects in everyday scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1135–1144.