MRIM: Lightweight saliency-based mixed-resolution imaging for low-power pervasive vision[1]

WU, Ji-Yan; SUBASHARAN, Vithurson; TRAN, Tuan; GAMLATH, Kasun; ARCHAN, Misra

Singapore Management University, 81 Victoria St, Singapore, 188065, Republic of Singapore

Published in Pervasive and Mobile Computing (2023) 96, 101858. DOI: 10.1016/j.pmcj.2023.101858

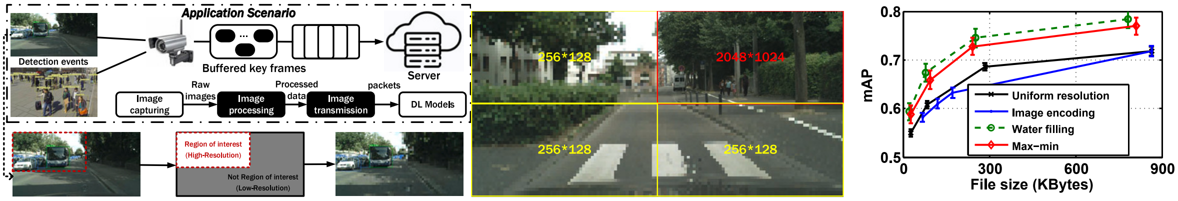
Abstract: While many pervasive computing applications increasingly utilize real-time context extracted from a vision sensing infrastructure, the high energy overhead of DNN-based vision sensing pipelines remains a challenge for sustainable in-the-wild deployment. One common approach to reducing such energy overheads is the capture and transmission of lower-resolution images to an edge node (where the DNN inferencing task is executed), but this results in an accuracy-vs-energy tradeoff, as the DNN inference accuracy typically degrades with a drop in resolution. In this work, we introduce MRIM, a simple but effective framework to tackle this tradeoff. Under MRIM, the vision sensor platform first executes a lightweight preprocessing step to determine the saliency of different sub-regions within a single captured image frame, and then performs a saliency-aware non-uniform downscaling of individual sub-regions to produce a "mixed-resolution" image. We describe two novel low-complexity algorithms that the sensor platform can use to quickly compute suitable resolution choices for different regions under different energy/accuracy constraints. Experimental studies, involving object detection tasks evaluated traces from two benchmark urban monitoring datasets as well as a prototype Raspberry Pi-based MRIM implementation, demonstrate MRIM's efficacy: even with an unoptimized embedded platform, MRIM can provide system energy conservation of 35+ % (~80% in high accuracy regimes) or increase task accuracy by 8+ %, over conventional baselines of uniform resolution downscaling or image encoding, while supporting high throughput. On a low power ESP32 vision board, MRIM continues to provide 60+% energy savings over uniform downscaling while maintaining high detection accuracy. We further introduce an automated data-driven technique for determining a close-to-optimal number of MRIM sub-regions (for differential resolution adjustment), across different deployment conditions. We also show the generalized use of MRIM by considering an additional license plate recognition (LPR) task: while alternative approaches suffer 35%–40% loss in accuracy, MRIM suffers only a modest recognition loss of ~10% even when the transmission data is reduced by over 50%.

Keywords: Mixed resolution, pervasive vision tasks, energy consumption

1. Introduction

Vision-based sensing, typically using a network of infrastructurally-deployed cameras, is an important enabler for a variety of pervasive computing applications, such as situation awareness [2], human activity detection [3], shopper behavior analytics [4]

[1] A preliminary version of this work Wu et al. (2022) was presented in the main conference of IEEE Percom 22' [1].



(a) Mixed-Resolution Image Transfer (b) Example of Mixed-Resolution Image (c) Accuracy gain (MRIM vs. Baselines)

Fig. 1. MRIM Paradigm and overall benefit.

and vehicular traffic monitoring [5]. Such expanded use of vision-based sensing has been accelerated by the reduced cost of high-resolution vision sensors (e.g., 4K resolution cameras) and the impressive accuracy gains achieved by DNNs (Deep Neural Networks) for tasks such as object detection [6] and facial recognition [7]. Vision-based sensing, however, continues to fairly energy intensive, especially in the wireless transmission of such high-resolution data and/or in the execution of DNN-based inference pipelines. Such overheads pose a major obstacle to greater in-the-wild deployment of vision sensing applications in spaces such as forests, parks and highways.

Given the increasing importance of developing a sensing infrastructure that is ultra-low power (and even battery-less) [8], a variety of approaches have explored the development of low-power vision sensing systems. Most such low-power vision systems adopt an offloading-based architecture, where the pervasive sensor platform simply *captures* and *wirelessly transmits* (possibly preceded by some lightweight encoding) to a more-resourced (e.g., GPU-equipped) edge node, where the actual DNN-based AI pipelines are executed. Even so, pervasive applications continue to suffer from the *fidelity-vs.-energy* tradeoff: reduction in energy consumption is achieved by sacrificing either resolution (spatial granularity) or frame rate (temporal granularity) of the transmitted image/video frames, which in turn affects the accuracy of DNN-based inference tasks.

In this work, we explore the use of a novel information-centric approach, ***M**ixed-Resolution **IM**aging* (**MRIM**), as a means of improving this fidelity-vs.-energy tradeoff. The *MRIM* approach (illustrated in Fig. 1(a)) seeks to increase the operational lifetime of the sensor platform by crafting a *lightweight* mechanism to reduce the volume of transferred data (and thus the dominant transmission energy cost) without affecting the subsequent DNN inference accuracy. Under this approach (as illustrated in Fig. 1(b), where 4 different sub-regions are processed at two distinct resolutions), the individual images captured by a camera sensor are broken up into multiple sub-regions, with the different sub-regions then down sampled at different resolutions prior to transmission. Our proposed approach, involving differential resolution *within* a single frame, is distinct from prior work on *inter-frame* dynamic camera resolution adaptation [9,10], which assume that any single frame is acquired, processed and/or transmitted at a uniform spatial resolution.

Intuitively, MRIM enables a more judicious use of system-resources on an energy-constrained vision sensor platform, adopting a lower resolution budget for the low-priority areas, while conserving the usage of higher resolution for the region of interest. The MRIM approach is motivated by two intuitive observations: (a) in most event-monitoring scenarios, objects or activities of interest are often not spread out uniformly over a camera's entire field-of-view (FoV) but localized or concentrated in certain (possibly multiple) *salient* sub-regions of the captured image; and (b) because the resolution reduction (and the corresponding loss in information fidelity) is applied disproportionately on the regions with lower saliency, the overall accuracy of DNN-based vision tasks remains largely unaffected. Building a practical MRIM-based vision-based sensing approach is, however, a non-trivial task and must address the following key research questions:

- How can the camera platform determine the saliency of different sub-regions within an image (a necessary prerequisite before applying the principle of mixed-resolution downsampling)? In particular, to ensure that any savings in transmission energy are not negated by a higher processing energy overhead, it is essential that this determination be computationally cheap and incur low latency.
- Can the relationship between the energy overheads vs. vision task accuracy (for varying resolution values) be accurately estimated across a diverse set of image/environmental context (e.g., for both images with a large number of small objects or a small number of large objects)?
- Given such saliency and energy-vs.-accuracy estimates, how can the camera platform use an image frame's object-level properties to determine, in a computationally lightweight manner, the right levels of resolution reduction (fidelity reduction) to be applied to each such sub-region?

Through our work, we show that it is indeed feasible to develop a practical MRIM approach that can overcome these challenges, focusing primarily on the commonplace *object detection* task (with additional studies on a license plate recognition task). In particular, we shall (a) propose and demonstrate the efficacy of mixed-resolution determination algorithms that significantly reduce transmission bandwidth/energy without adversely affecting the accuracy of state-of-the-art DNN-based object detectors, and (b) develop a lightweight, low-power and accurate saliency determination approach that executes on the camera platform. Using a combination of a real-world, non-optimized Raspberry Pi-based prototype and diverse, real-world image traces, we shall demonstrate that *MRIM* can provide either a total systems-level energy savings of ≈35% or task accuracy improvement of 8%over current approaches of uniform resolution reduction or image encoding. Furthermore, the energy reduction achieved, vis-a-vis uniform downsampling at equivalent accuracy, remains significant (60+%) for ESP32, a low-power micro-controller based vision board.

Key Contributions: We make the following key contributions:

- *Introduce the MRIM framework:* Through systematic studies, we establish both (a) the tradeoff between the visual resolution of objects and the resulting accuracy of DNN object detectors, and (b) the non-uniform spatial properties of such objects in the FoV of typical pervasive camera deployments. These insights help motivate the principle of mixed-resolution imaging, which preferentially preserves pixel resolution in sub-regions with a greater *predicted* number of objects, while degrading the resolution of less salient sub-regions.
- *Devise & Evaluate Mixed-Resolution Algorithms:* We formulate the MRIM problem as one of (i) either minimizing the total image transmission size subject to a mean task accuracy constraint, or conversely, (ii) maximizing the task accuracy subject to a maximum energy budget. We then present two novel algorithms: (a) *Max-Min*, which preferentially increases the resolution of higher saliency sub-regions until the image-level objectives are satisfied, and (b) *Water-Filling*, which incrementally increases the resolution of all individual sub-regions equitably, until the image-level objectives are achieved. Via experimental studies with two different tasks/datasets–(a) human detection using *WildTrack* [11] and (b) vehicle detection using *CityScapes* [12]—we show that our proposed algorithms can provide ∼10−20% improvement in object detection accuracy compared to currently-adopted approaches of either image encoding or uniform resolution adjustment. Fig. 1(c) summarizes the performance gains achieved by our *MRIM* strategies.
- *Demonstrate the Overall Effectiveness of an MRIM-based System:* We build and evaluate a working prototype of an *MRIM*-based camera, using the RPi (v3) board. The prototype integrates the mixed-resolution algorithms with a lightweight object detection technique (empirically shown to incur only 10 mJ/frame energy overhead) to determine the saliency of different sub-regions. Through careful experimental studies, we demonstrate that, in spite of many non-ideal system characteristics (e.g., high baseline power consumption), our *MRIM* approach provides energy savings/frame of 33−36% and 28%, respectively, over the uniform resolution and image encoding approaches across a range of compression values (with savings as high as 90% for high-accuracy settings. In addition, on a low power ESP32 vision board, *MRIM's* Max-Min technique achieves 60+% energy savings and a significantly lower energy consumption (< 40 mJ/frame), compared to uniform resolution reduction. Overall, *MRIM* allows the operational lifetime of such pervasive vision sensors to be almost doubled without loss in task accuracy.
- *Demonstrate the Automated Functioning and General Applicability of MRIM:* The performance of *MRIM* depends on the choice of N_r, the number of sub-regions into which the captured image is partitioned. Different deployments have different optimal N_r choices, based on the statistical distribution of object sizes and locations. We show how *MRIM* can perform auto-calibration of N_r, across diverse deployments, using an initial stabilization phase. We also show how the *MRIM* paradigm can provide significant energy-vs.-accuracy gains for an additional visual perception task, *license plate recognition*, where higher resolution is critical for maintaining the finer details needed to recognize characters.

The remainder of this paper is organized as follows. Section 2 discusses the related works close to this study. In Section 3, the motivation of this study is introduced by analyzing the energy-quality-accuracy tradeoff. The proposed water-filling and max–min algorithms are presented in Section 4. Section 5 describes the evaluation methodology and results. The discussion and conclusion remarks are given in Sections 6 and 7, respectively.

2. Related work

MRIM draws upon prior work in both (a) adaptive resolution in image capture, and (b) energy/power optimization in intelligent vision sensing systems.

2.1. Low power camera/vision sensing

LiKamWa et al. [10] demonstrated that image sensor energy consumption is ideally proportional to frame rate and resolution, and suggested multiple techniques (clock frequency control and low-power standby mode) to further reduce sensing power. To reduce the energy overhead of vision-related tasks, prior approaches utilize either on-board image processing (e.g., Cyclops [13]), a combination of low & high resolution cameras (e.g., SensEye [14]) or selective event-triggered activation of power-hungry vision sensors (e.g., Glimpse [15]). Several novel approaches have developed ultra-low power or battery-less camera sensors—for example, WISPCam [16] uses an RFID-powered harvester to trigger the capture of low-resolution, low frame-rate images, Naderiparizi et al. [17] utilizes analog backscatter communication to transfer HD-quality video transfer from an energy-harvesting vision sensor, while Elf [18] supports object counting by solar-powered cameras by adaptively adjusting the frame rate. In almost all cases, these approaches either require specialized hardware or additional infrastructure (e.g., RFID readers) and usually support low quality, infrequent (< 1 FPS) image capture. Collaborative sensing, across multiple cameras, has also been used to reduce the per-sensor energy overheads by opportunistically *deactivating* selected cameras-e.g., EECS [19] uses knowledge of (a) each camera's object detection accuracy, and (b) potential energy overhead to select a preferred set of {activated cameras, video processing parameters} to monitor a common region. Zam et al. [20] proposed an energy aware approach for collaborative object tracking, whereby specific battery-powered sensors are activated taking into consideration their residual energy and the object's trajectory. A camera-based facial identification framework is presented in [7] to reduce the latency and energy costs by using multi-DNN pipeline at both edge and cloud.

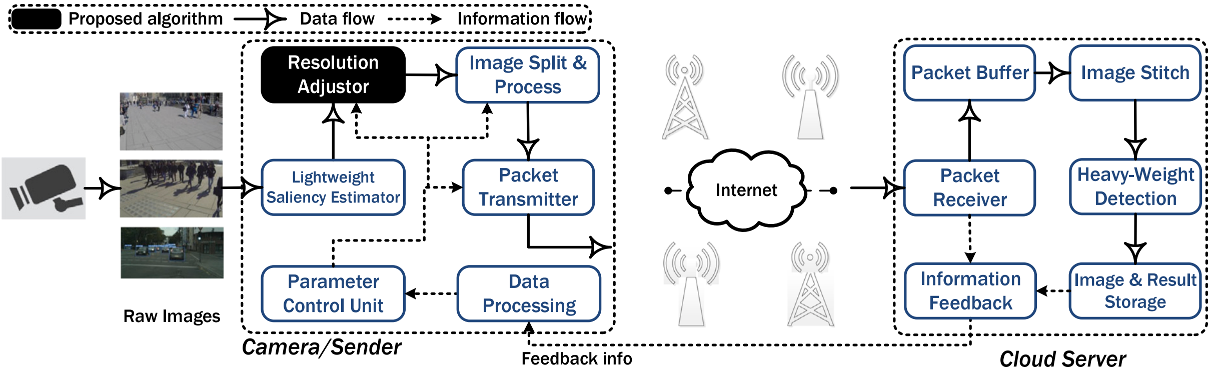


Fig. 2. System design of the proposed accuracy and energy aware mixed-resolution image transmission framework.

2.2. Mixed-resolution image processing

Past work has studied the broad relationship between image resolution and accuracy of vision-based tasks. The Banner prototype [9] demonstrated how dynamically reducing the *overall* image resolution (in contrast to *MRIM's* approach of utilizing differential resolution within a single image), based on the object's distance, can reduce camera sensing energy by 70%. The concept of *image resizing*, as a means of accelerating the computation of vision tasks, has also been recently explored in [21], where different regions of a single image are reduced, at an edge node, to different sizes (based on their priority), thereby increasing the overall inferencing throughput. Vision-based systems have also explored the processing of mixed-resolution *multiview* videos, where different cameras capture images at different spatial resolution and content from higher-resolution video streams is used to upscale the images captured by lower-resolution cameras (e.g., [22]). Richter et al. [23] propose a novel image super-resolution approach that is robust against both non-perfect calibration of spatially low-resolution depth sensors and inaccurate depth acquisition. However, all these mixed-resolution researches have not systematically studied the resolution-accuracy-energy relationship. Mallik et al. [24] present High Efficiency Video Codec based spatial resolution scaling type of mixed resolution coding model for frame-interleaved multiview videos. A flexible high-resolution object detection method is proposed in [25] on edge devices. This framework comes up with a image partition and model execution plan to maximize the overall detection accuracy. More recently, the MOSAIC system [26] performs spatial resizing and bin-packing of object-based tiles from multiple camera feeds into a single 'canvas' frame, thereby allowing edge devices to concurrently process a much larger number of image streams with negligible loss of accuracy.

All of these approaches focus on the differentiated *processing* of video/image streams, and are conceptually distinct from *MRIM*, which focuses on the low-overhead *capture and transfer* of images by a high-resolution camera.

2.3. Efficient on-board image processing

We shall see that *MRIM's* success lies partly in being able to determine the saliency of different image sub-regions in an ultra-lightweight manner. Light-weight neural detection models, such as Haar feature [27], LFFD [28] and libface [29]), have been proposed for on-board execution. FastMOT [30] implements a combined, lightweight object detector and tracking technique using either YoLo or SSD object detector models but requires GPU support on the sensing device. To support accurate object detection, approaches such as MobiSR [31] utilize a cheaper, low-resolution camera for image capture, followed by on-board upscaling on mobile devices to generate super-resolution images.

3. Motivating the *MRIM* Approach

Our overall approach for low-power pervasive vision tasks utilizes the system architecture illustrated in Fig. 2. In this architecture, the vision sensor platform performs the following key functions: (a) image capture–i.e., using the sensor to capture the raw image; (b) image pre-processing–i.e., performing any functions (such as compression and/or resolution reduction) locally prior to transmission; and (c) image transmission–i.e., using a suitable networking interface (e.g., WiFi/4G/5G) to transfer the processed image to an edge/cloud device. The edge/cloud platform then performs the vision task by executing the DNN pipeline. To keep the pervasive sensor cost and energy overheads low, we assume that the sensor platform does not have specialized hardware (e.g., GPUs) and cannot thus efficiently execute the complex state-of-the-art DNN models, such as YOLO v3 [32].

Our focus is purely on reducing the total power/energy consumption of the sensor platform (without compromising on the eventual accuracy of the vision task), such that this sensor platform has a longer operational lifetime. To achieve this, the sensor platform performs additional pre-processing via two conceptually distinct functional components: (a) **Saliency Estimator**: as a precursor to performing differential resolution downscaling, it determines the saliency of different regions in the image frame by estimating the likely general location and other relevant attributes of objects of interest. Note that, for high frame rate video, saliency

Table 1
Mathematical notations.

Symbol	Definition	Symbol	Definition
$\mathbb{N}_d$	No. of detections	$\mathbb{N}_r$	No. of sub-regions
mAP	Target mean average precision	$\mathcal{R}_i\vert_{1\leq i\leq N}$	Image regions
$\mathcal{S}_i\vert_{1\leq i\leq \mathbb{N}}$	Confidence scores	$\mathbb{E}$	Energy constraint
E	Total energy consumption	$\mathcal{S}$	Image file size
$\mathcal{V}_i\vert_{1\leq i\leq \mathbb{N}}$	Regional resolution values	Est_Eng	Est. energy consumption

determination need not be performed on each frame, but only intermittently (e.g., once every 1–2 s), as object attributes, such as their location and relative size, are unlikely to dramatically vary over O(msec) timescales; (b) **Resolution Adjuster**: this component, which lies at the rear of *MRIM*, modifies the resolution of each sub-region of the captured image, taking into account the region's saliency and the resulting accuracy-vs.-energy tradeoffs.

MRIM's requires careful consideration of the tradeoff between the Pre-processing and Transmission energy overheads: intuitively, the additional steps of saliency estimation and resolution adjustment will result in increased pre-processing energy, which should be offset by a greater reduction in transmission energy. In addition, we will need to show that our proposed approach offers a superior energy-vs.-accuracy profile compared to two established baselines for reducing transmission overheads: (i) Image compression/encoding, where standard codecs (often implemented in hardware) are used to perform lossy compression of the image/video content, and (ii) Uniform resolution adaption, where the entire image is uniformly downscaled (without consideration of the image content) to a specified size. Determining the right choices for *MRIM* thus first requires a careful understanding of both (i) the energy overheads of different pre-processing mechanisms and the subsequent transmission phase, and (ii) the resulting impact of different image sizes/resolutions on the DNN inference accuracy.

3.1. Modeling system energy consumption

We first model the energy consumption for processing and transmitting target image frames captured by an embedded camera platform (Table 1 lists the basic mathematical notations used). Specifically, the image processing (with our resolution adjustment algorithm or JPEG encoding) and data transmission (using 3G/4G/Wi-Fi chip) account for the main portion of power consumption of image application.

The total system energy consumption E including the idle (baseline) energy, as well as the energy spent in image capture, processing and subsequent transmission, is represented as:

$$\mathsf{E} = \mathsf{E}_{\text{idle}} + \mathsf{E}_{\text{cap}} + \mathsf{E}_{\text{proc}} + \mathsf{E}_{\text{tran}}, \tag{1}$$

where $\mathsf{E}_{\text{idle}}, \mathsf{E}_{\text{cap}}, \mathsf{E}_{\text{proc}}, \mathsf{E}_{\text{tran}}$ represent the idle state (i.e., baseline), image capture, processing and transmission energy, respectively. The idle baseline energy includes energy spent in powering the different system components (e.g., processor, SD card, etc.). The transmission energy depends on the data transmission power P_{tran} and duration (i.e., the amount of data transferred), and can be represented as $\mathsf{E}_{\text{trans}} = \mathsf{P}_{\text{tran}} \cdot d$; also, the capture energy E_{cap} is usually negligibly smaller (O(μW), compared to O(mW)) than the other components.

The processing energy incurred consists of both E_{sal}, the energy spent in saliency estimation (if this step is required), and $\mathsf{E}_{\text{image}}$, the subsequent energy spent in modifying the captured image–i.e.,:

$$\mathsf{E}_{\text{proc}} = \mathsf{E}_{\text{sal}} + \mathsf{E}_{\text{image}}. \tag{2}$$

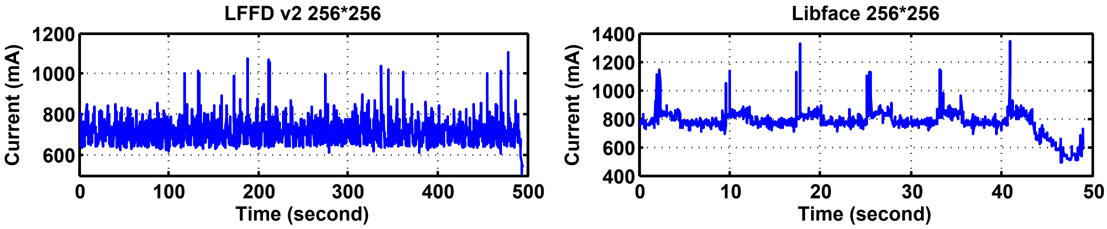
For *MRIM*, $\mathsf{E}_{\text{image}}$ includes the energy spent in *MRIM* in computing the modified resolution values and the subsequent downsampling; for a conventional compression-based approach, $\mathsf{E}_{\text{image}}$ would represent the energy spent in lossy compression, while $\mathsf{E}_{\text{image}} = 0$.

In Eq. (1), the transmission energy is dependent on the data transmission power P_{tran} and duration. This power can be estimating by measuring the difference before and after enabling the image data transfer. Therefore, we can have the equation for image application as follows.

$$\mathsf{E}_{\text{tran}} = \mathsf{P}_{\text{tran}} \cdot d, \tag{3}$$

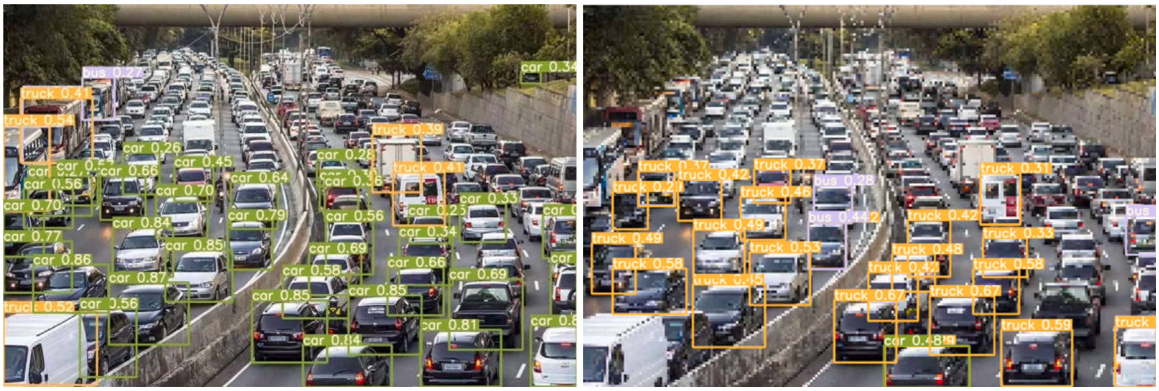
in which d represents the video session duration. In practical system measurements, the transmission energy value is obtained through the recorded power/current values and duration. The above method to measure transmission power can be adopted to measure the other power components as well.

Lightweight Saliency Estimator: *MRIM* relies on the use of a lightweight model to derive the approximate count/distribution of objects in each image sub-region and thereby derive each region's saliency. The energy consumption of the saliency estimator E_{sal} needs to be measure under different hardware/software environments. We evaluated multiple light-weight models for this purpose (e.g., the LFFD [28] and libface [29]). Fig. 3 presents the typical current values, where both detectors perform the saliency inference at the commonly-used resolution of 256 $*$ 256. We can observe that the power profiles of the two light-weight models are quite distinct. To further reduce this part of energy consumption, we can adopt an intermittent saliency estimation strategy, where the light-weight detector is executed only periodically, instead of on each image frame. Broadly speaking, the light-weight saliency estimator can be executed with lower frequency for images/environments where the objects move slowly, and vice versa.



(a) LFFD (b) Libface

Fig. 3. Power consumption profile for light-weight models for saliency estimation.



(a) 1350*900 (b) 225*75

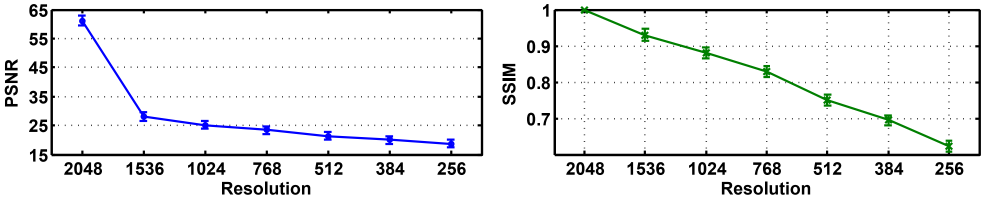
Fig. 4. DNN detection accuracy vs. image resolution.

To ensure *MRIM's* overall energy efficiency, it is important to characterize the energy profile of such candidate lightweight models. As measured via an implementation on the Raspberry Pi 3B platform, a Haar feature based detector [33] is able to achieve as low as 200 mJ for each iteration, and is shown to provide adequate indication of the likely presence of objects in individual sub-regions. In practice, this estimator can be run intermittently (e.g., once every 1–2 s or 20–30 frames), resulting in a very low normalized energy overhead (∼1–2 mJ/frame).

3.2. DNN accuracy vs. Resolution

The *MRIM* approach is premised on the observation that the accuracy of DNN-based object detectors depends on the resolution (reflecting the information fidelity) of the underlying images. To understand this phenomenon in detail, we consider a typical state-of-the-art DNN model, such as YOLO v5 [34]. The output of such an object detector includes the *class id* (e.g., a person or vehicle object), *bounding box coordinates* (the center point, width and height) and the *confidence score* (a value between (0,1) that represents the probability of the bounding box containing an object). To represent accuracy, we adopt the widely-used mean Average Precision (mAP) metric, which computes the mean AP over all classes and/or overall IoU (Intersection over Union) thresholds as follows: $mAP = \frac{\sum_{k=1}^{N_c} AP_k}{N_c}$, where N_c is the number of classes and AP_k indicates the average precision for the kth class (computed over a range of IoU values).

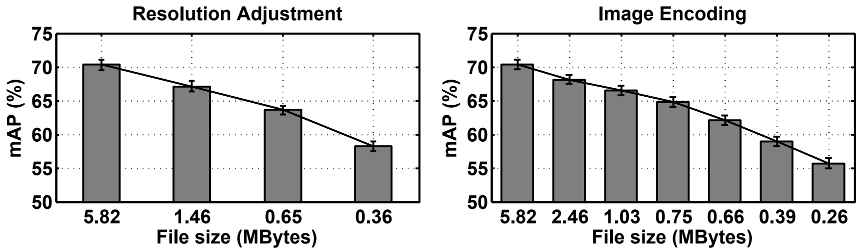
As an illustration of our underlying hypothesis, Fig. 4 plots the bounding boxes and confidence values identified by the YOLO v5 object detector [34] on two images of the same scene, but at different levels of resolution (original = $1350 * 900$ in Fig. 4a, reduced = $255 * 75$ in Fig. 4b. We can clearly observe both a decrease in the number of detected vehicles as well as an increase in the misclassification rate (e.g., observe several 'cars' in Fig. 4a being classified as 'trucks' in Fig. 4b). In addition, there is a substantial reduction in the confidence scores of the detected objects. In addition, we shall later see (Section 6) that such reduced resolution is even more harmful for tasks such as character recognition, leading to accuracy loss of 60+%. On closer inspection, we see that the "smaller size" vehicles appearing in the upper half part of the image suffer a higher accuracy loss than those "larger-sized" vehicles in the lower half. Although the mAP is reduced from 69.3% to 56.8% due to the resolution downgrade, the file size S of the underlying image exhibits a 12-fold reduction, from 707 to 57 kB, which should (as per Eq. (1)) lead to a reduction in the total energy consumption. Motivated by these observations, we now conduct a deeper study of the relationship between the system energy E, deep learning model accuracy mAP and image quality.



(a) PSNR

(b) SSIM

Fig. 5. Image quality vs. resolution.



(a) resolution

(b) encoding

Fig. 6. mAP versus file size.

3.3. Energy-Quality-Accuracy (EQA) tradeoff

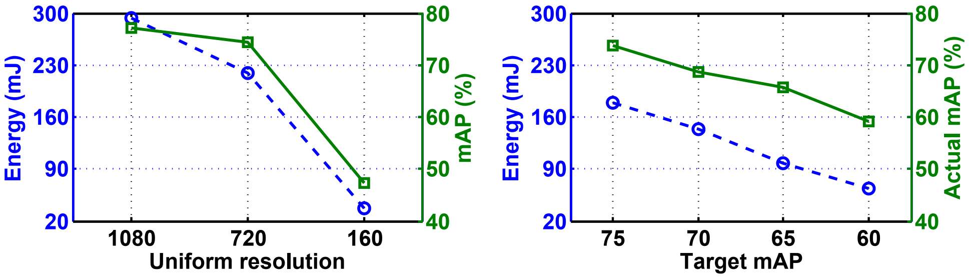
Generally speaking, the model accuracy is proportional to image *fidelity*, with the fidelity itself correlated to the resolution of the input image. To illustrate this point, we study how the fidelity/quality of images varies as the overall image resolution is progressively decreased. Fig. 5 plot the mean and confidence intervals (based on a corpus of 500 images curated from the WildTrack and CityScapes datasets) of two widely used measures of objective and subjective image quality, PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity), respectively, as the input image resolution varies from $2048 * 2048$ to $256 * 256$. We see that a reduced resolution leads to a progressive loss in quality, due to the degradation in the color and positional features of the underlying objects.

Similar to image quality, the accuracy of DNN-based vision tasks should also increase with the underlying image resolution. However, after a certain point, any increase in image resolution provides only a marginal improvement in vision task accuracy. As mentioned earlier, a reduction in image resolution has two effects: it decreases the mAP of task accuracy as well as the size of the underlying image files. To capture this relationship, Fig. 6 plots the mAP vs. file sizes (resolution) for two different baseline strategies: (a) uniform resolution downsampling and (b) compressive image encoding (using the principle component analysis compression technique). The figure plots the mean and 95% confidence intervals, computed over the 500 representative images mentioned earlier. In addition, we also use our Raspberry Pi implementation (Section 5) to empirically measure the resulting processing latency and energy consumption.

We observe that both approaches, uniform downscaling and encoding, exhibit an almost identical mAP-vs.-size tradeoff. In addition, the improvement in mAP values is more dramatic at low file sizes and becomes more muted as the image resolution increases from medium to high resolution–e.g., an ~4-fold increase in image size from 1.46 MB to 5.82 MB results in an mAP increase of ≤2%. On carefully analyzing the mAP performance for individual images, we further observe that:

- For images with predominantly larger-sized objects, the mAP degradation is not significant as the file size decreases. Conversely, the mAP values for images with predominantly small objects exhibit a much steeper drop as the file size (image resolution) decreases.
- While the mAP-vs.-file size variation is similar for both resolution downscaling and image encoding, the two strategies differ in their computational cost and latency. In particular, the image compression approach incurs much higher latency and energy (avg. = 261 ms and 73.8 mJ) than the resolution downscaling approach (avg. = 141 ms and 53.7 mJ). Intuitively, resolution downscaling involves very simple averaging operations, whereas encoding requires multiple steps to compute the statistical distribution of pixel intensities and the resulting codebook.

The EQA Tradeoff: Combining the individual experimental results allows us to now understand the energy-vs.-accuracy(quality) or *EQA* tradeoff generated by changing image quality (reflected by different image sizes). We conduct the experiments in two ways: (i) evaluate the energy and detection accuracy (precision) of different uniform resolutions (from 160p to 1080p) and (ii) later, gradually decrease the resolution values of the image regions of the 1080p image until the mAP reduces to {75%,70%,65%,60%}.



(a) uniform resolution

(b) mixed-resolution (MRIM)

Fig. 7. Energy versus mAP.

Fig. 7 plots the results, for both the uniform downscaling approach and, for comparison, the differential techniques (*MRIM*) that we shall detail in Section 4.

The results in Fig. 7a indicate the crux of the problem: under uniform resolution, *both* energy and mAP drop significantly when resolution drops from 720p to 160p. However, as shown in Fig. 7b, the mixed resolution approach permits a much more gentle decrease in mAP compared to the linear drop in energy. By deliberately and gradually changing the resolution of individual regions, we can achieve approximately the same level of accuracy (as shown in the green plot) with lower energy (blue plot) than that achieved via uniform resolution downscaling.

3.4. Problem formulation

MRIM's choice of the downscaled resolutions for each sub-region can then be formulated as one of two distinct objective functions:

- **P1**: Given a minimum mean average precision, adjust the resolution of each image sub-region so as to minimize the total energy consumption, i.e.,

$$\mathsf{P1}\,:\,\mathcal{V}_i|_{1\leq i\leq \mathbb{N}} = \underset{mAP\geq\overline{mAP}}{\arg\min}\,\{\mathsf{E}\}\,.$$

  For this objective (suited for scenarios where the vision task has a minimum required fidelity), the accuracy serves as a lower-bound constraint (e.g., $\overline{mAP} \geq 75\%$).
- **P2**: Given an energy constraint E, adjust the resolution values of each image sub-region so as to maximize the DNN task accuracy, i.e.,

$$\mathsf{P2}\,:\,\mathcal{V}_i|_{1\leq i\leq \mathbb{N}} = \underset{\mathsf{E}\leq\overline{\mathsf{E}}}{\arg\max}\,\{\mathrm{mAP}\}\,.$$

  In this case (suited for scenarios where the platform has a finite battery capacity and a target lifetime), E serves as an upper-bound constraint (e.g., $\overline{\mathsf{E}} \leq 50$ mJ/frame).

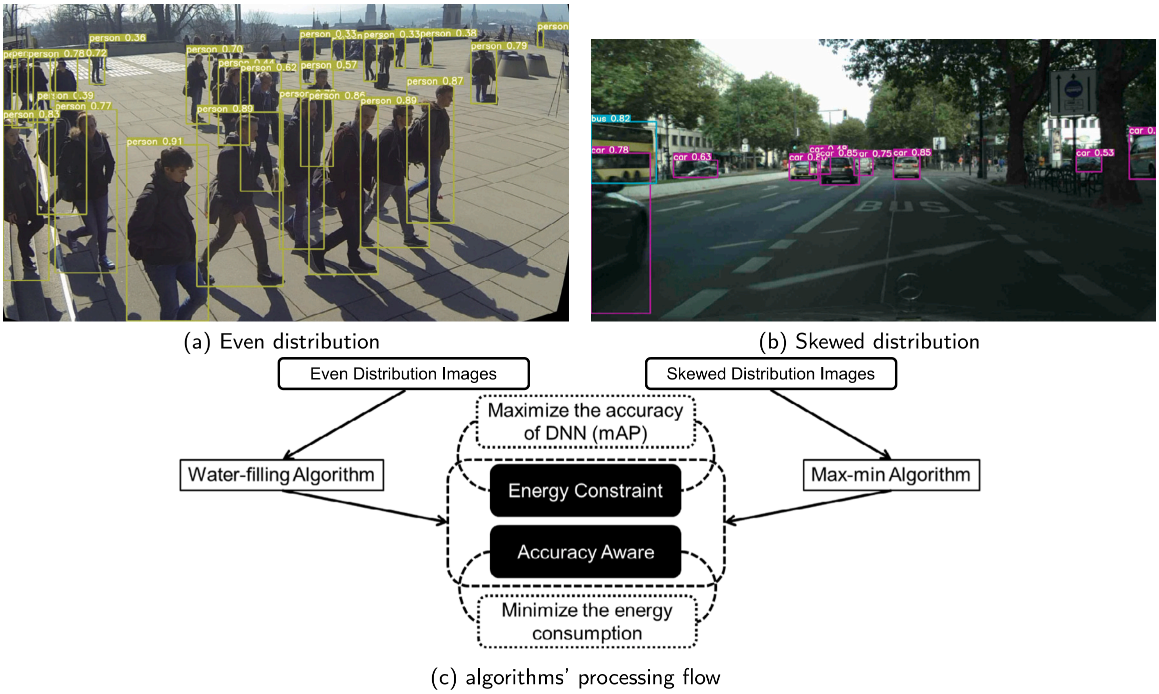
Given the many real-world non-ideal characteristics in both system energy consumption and DNN performance, developing a provably optimal solution to each problem is infeasible and impractical. Hence, we shall next focus on developing efficient (low-complexity) heuristic algorithms for P1 and P2.

4. Resolution adjustment algorithms

We now describe two different algorithms that the Resolution Adjuster can use to determine the different resolution choices for each of the sub-regions. The algorithm design is driven by our observation (using open-source image datasets corresponding to two representative tasks, human detection and vehicle detection, and illustrated in Fig. 8) that most captured images exhibit one of two spatial characteristics:

(i) *Uniform spatial distribution* (see Fig. 8a), where objects of interest are typically distributed across the entire image, even though the size of the objects vary depending on the observer-object distance.
(ii) *Skewed Distribution* (see Fig. 8b), where objects of interest are typically observed in selected salient sub-regions of the image–e.g., mostly confined to the upper or left portion of the camera's FoV.

In addition, as observed earlier in Fig. 4, reduced resolution impacts the detection accuracy for smaller-sized objects disproportionately, implying that the algorithms must also incorporate the different resolution-to-mAP relationship for different *object sizes*. Given these observations, (i) the Max–Min algorithm tends to allocate higher resolution values disproportionately to a smaller number of high saliency areas until the image (as a whole) satisfies a minimum predicted mAP value, and is thus better suited for images with skewed spatial distribution, whereas (ii) the Water-Filling algorithm, which conceptually attempts to equalize the predicted



(a) Even distribution

(b) Skewed distribution

(c) algorithms' processing flow

Fig. 8. Even & Skewed spatial distribution images and algorithms overview.

mAP values of all sub-regions, is better suited for images with uniform spatial distribution. Fig. 8c presents an overview of the two types of proposed algorithms, which are designed to address the distinctive properties of even and skewed distribution images, respectively. For each category of algorithms, either the energy constraint or the target accuracy is considered when optimizing the application accuracy/energy consumption.

4.1. Estimating resolution-to-mAP values

To enable the *MRIM* algorithms to determine the right resolution choices, we first need to build a predictive estimate of how the final DNN task accuracy will be affected by different possible downscaled resolution candidates. To estimate this, we first compute the *weighted confidence* of the objects detected by the Saliency Estimator as: $WConf = \frac{\sum_{i=1}^{\mathbb{N}} S_i \cdot bbox_i}{\sum_{i=1}^{\mathbb{N}} bbox_i}$, where $bbox$ represents the bounding box area and S represents the class confidence of each detected object. Subsequently, we utilize a look-up table (which has been populated by extensive offline empirical studies) that maps this confidence value to an overall image mAP predicted to be achieved, for different image resolution/size values, when the heavyweight DNN (YOLOv5 [34] in our case) is executed on the edge node. For example, a sample entry in the lookup table might be of the form ($WConf = 57.2, Res = 300, mAP = 67.6$), implying that a Saliency Estimator WConf value of 57.2% is estimated to eventually result in a YOLO mAP = ≈67.6%, if the image was downsized to a 300 × 300 resolution.

4.2. Max-min algorithm

The Max-Min algorithms operate in greedy fashion, preferentially adjusting the resolution of individual sub-regions in order of descending priority, until the overall image mAP reaches the specified threshold. The priority order is determined by the saliency of individual sub-regions (which is pre-computed based on the number of objects predicted by the Lightweight Saliency Estimator) as well as an evaluation of whether the initial overall image mAP is lower or higher than the target mAP. At a high-level, if the current overall image mAP is lower than the target mAP, we seek to preferentially increase the resolution of the regions with highest saliency; conversely, if the current overall image AP is higher than the target mAP, we preferentially decrease the resolution of the regions with the lowest saliency.

Algorithm 1 provides the high-level pseudocode of the Max-Min algorithm for objective P1. The algorithm inputs include the number of sub-regions $\mathbb{N}_r$, number of objects $\mathbb{N}_d$ and their bounding boxes (detected by the Lightweight Saliency Estimator), and target mean average precision $\overline{mAP}$. The algorithm starts with a nominal (low) resolution allocation to each of the sub-regions, and uses the afore-mentioned lookup table to estimate (line 1) each sub-region's anticipated mAP score, as well as the image's overall

Algorithm 1: Accuracy-Aware Energy Minimization (Max-Min Optimization)

```
   Input: $\mathbb{N}_d$, $\mathbb{N}_r$, bounding boxes, $\mathcal{S}_i|_{1\le i\le\mathbb{N}}$, $\overline{mAP}$, $\mathcal{V}_i|_{1\le i\le\mathbb{N}}$;
   Output: $\mathcal{V}_i|_{1\le i\le N}$, $\mathcal{Q}_i|_{1\le i\le N}$, $mAP$;
 1 Calculate $mAP$ based on confidence scores for each class;
 2 if $mAP < \overline{mAP}$ then
 3     Rank the regions $\mathcal{R}_i|_{1\le i\le\mathbb{N}_r}$ based on included bounding box areas in descending order;
 4     for each region $\mathcal{R}_i|_{1\le i\le\mathbb{N}_r}$ do
 5         for $\Delta\mathcal{V}$ from small to large resolution changes do
 6             $\mathcal{V}_i = \mathcal{V}_i + \Delta\mathcal{V}$;
 7             $\mathcal{S}_j = \mathcal{S}_j + \overline{\mathcal{S}}, \forall$ object $j \in \mathcal{R}_i$;
 8             $\mathcal{Q}_j = \mathcal{Q}_j - \Delta\mathcal{V}$;
 9             $mAP = \frac{\sum_{i=1}^{\mathbb{N}} \mathcal{S}_i}{\mathbb{N}}$;
10             if $mAP \ge \overline{mAP}$ then
11                 break;
12             end
13         end
14     end
15 end
16 else
17     Rank the regions $\mathcal{R}_i|_{1\le i\le\mathbb{N}_r}$ based on included bounding box areas in ascending order;
18     for each region $\mathcal{R}_i|_{1\le i\le\mathbb{N}_r}$ do
19         for $\Delta\mathcal{V}$ from small to large resolution changes do
20             $\mathcal{V}_i = \mathcal{V}_i - \Delta\mathcal{V}$;
21             $\mathcal{S}_j = \mathcal{S}_j + \overline{\mathcal{S}}, \forall$ object $j \in \mathcal{R}_i$;
22             $mAP = \frac{\sum_{i=1}^{\mathbb{N}} \mathcal{S}_i}{\mathbb{N}}$;
23             if $mAP < \overline{mAP}$ then
24                 $\mathcal{V}_i = \mathcal{V}_i + \Delta\mathcal{V}$;
25                 break;
26             end
27         end
28     end
29 end
30 return $\mathcal{V}_i|_{1\le i\le N}$, $mAP$;
```

mAP. In case the estimated mAP is below the target mAP, the algorithm proceeds to progressively increase the resolution $\mathcal{V}_i$ of individual sub-regions individually, starting with the most salient sub-region (the one with the largest number of predicted objects). At each step of such resolution adjustment, it recomputes the overall mAP (lines 6–9) and stops whenever this overall mAP has exceeded the target value (lines 10–12). However, if the overall image mAP has not reached the target value even after the first sub-region's resolution has been maximally increased, the algorithm then greedily proceeds to the region with the next highest saliency. This process is repeated until the overall mAP target has been achieved or all possible regions have been expanded to the maximum permissible resolution. Conversely, if the initial estimated overall mAP is higher than the target mAP, the algorithm assumes that the current resolution choices are too generous and seeks to iteratively decrease the resolution of sub-regions, starting with the lowest saliency region, until it 'just' exceeds the target mAP. To execute this, the algorithm now prioritizes the sub-regions in the reverse order of saliency (line 7), starting with the region with the lowest number of predicted objects. It then iteratively reduces the resolution of each such region, until the point that any further decrease will cause the overall image mAP to fall below the specified threshold.

The complexity of Algorithm 1 is $O(\mathbb{N}_r \cdot \mathbb{N}_d \cdot \frac{V}{\Delta V})$. We shall show in Section 5 that, under reasonable values of $\mathbb{N}_r$ (=4, 6, 8, ...), the complexity of this algorithm is low enough to permit low-latency, low-energy execution on embedded platforms. We now present a detailed explanation of the steps in Algorithm 1.

- Line 1: Estimate the current mAP (mean average precision) value based on the light-weight saliency detection results for all classes and input algorithm values. Lines 2–15 deal with the where mAP is less than the target precision value, implying the need for increasing the resolution, whereas lines 16–29 deal with the converse case where we can decrease resolution in selected sub-regions.

- Line 3: Rank the regions $\mathcal{R}_i|_{1\leq i\leq N}$ based on included bounding box areas in descending order. Through our experimental studies, this metric provides a better estimate of each region's importance, compared to alternative metric such as the number of detected objects or average confidence scores.
- Line 4–14: We loop through the regions, in descending order of priority, to determine the modifications to resolution values needed to minimally meet the overall mAP target.
- Line 5–13: We change the resolution values incrementally from smaller to larger values (e.g., 32, 64, 128), so as to try and *just* exceed the target mAP value. To perform this task, we also estimate the detection accuracy at each iteration and update the average mAP estimate, based on the updated resolution values. Once this overall estimated average mAP exceeds the target value, we abort any further resolution adjustment.
- Line 17: Rank the regions, based on bounding box areas, but now in *ascending* order.
- Line 18–28: Similar operations to the previous for loop, but here we gradually decrease the per-region resolution values. We abort further iterations and revert to the most-recent resolution value choices if mAP falls below the target value.

The overall greedy approach of Max-Min can also be suitably adapted to tackle the optimization problem P2 with the energy constraint $\overline{\mathcal{E}}$, as detailed in Algorithm 2. As before, the algorithm operates in greedy fashion, prioritizing individual sub-regions on the basis of their estimated saliency. In this case, however, during each iteration of resolution adjustment, the total energy consumption for that specific region (and thus the overall image) is re-estimated (using Eq. (1)), with the adaptation process continuing until the estimated total energy is 'just' below the permitted energy budget.

Algorithm 2: Energy-Constrained Accuracy Maximization

```
Input: $\mathbb{N}_d$, $\mathbb{N}_r$, bounding boxes, $\mathcal{S}_i|_{1\leq i\leq \mathbb{N}}$, $\mathbb{E}$, $\mathcal{V}_i|_{1\leq i\leq \mathbb{N}}$;
Output: $\mathcal{V}_i|_{1\leq i\leq N}$, $\mathcal{R}_i|_{1\leq i\leq \mathbb{N}_r}$, Est_Eng;
$\mathcal{R}_i = \mathcal{R}_i^{\min}, \forall 1 \leq i \leq \mathbb{N}_r$;
Rank the regions $\mathcal{R}_i|_{1\leq i\leq N}$ based on included bounding box areas in descending order;
for each region $\mathcal{R}_i|_{1\leq i\leq N}$ do
    for $\Delta\mathcal{V}$ from small to large resolution changes do
        $\mathcal{V}_i = \mathcal{V}_i + \Delta\mathcal{V}$;
        $\mathcal{S}_j = \mathcal{S}_j + \overline{\mathcal{S}}, \forall$ object $j \in \mathcal{R}_i$;
        $\mathcal{Q}_j = \mathcal{Q}_{\min}$;
        $E_i = E_i^{\text{idle}} + E_i^{\text{enc}} + E_i^{\text{tran}}$;
        Est_Eng $= \sum_{i=1}^{N} E_i$;
        if $Est\_Eng \geq \overline{\mathbb{E}}$ then
            $\mathcal{V}_i = \mathcal{V}_i - \Delta\mathcal{V}$;
            break;
        end
    end
end
return $\mathcal{V}_i|_{1\leq i\leq N}$, Est_Eng;
```

4.3. Water-filling algorithm

The Water-Filling Algorithms are inspired by prior work on equalizing channel performance in communication systems, and are based on the observation that water height effectively equalizes across multiple connected reservoirs independent of their individual heights. At a high-level, the algorithm views the resolution (pixel count) as a fluid resource that is effectively distributed among the different sub-regions so as to equalize their water *height*, where the height is defined by each region's predicted mAP value.

Algorithm 3 provides the high-level pseudocode for this approach for objective P1 (minimizing energy under an accuracy constraint). The algorithm starts by assuming each sub-region to be associated with the lowest permitted spatial resolution (pixel count). The resolution level is then incrementally increased across the board, and the resulting average accuracy is computed. Subsequently, each sub-region's individual predicted mAP score (height) is compared against this image-wide average, and the resolution for that region is iteratively increased until it is no longer below the current image-wide average value. This iterative approach effectively causes regions with higher saliency (larger number of objects or smaller-sized objects) to benefit from an increased allocation of pixel account, thereby assuring that such regions do not suffer poor accuracy. The Water-Filling approach can also be similarly adapted to objective P2 (maximizing accuracy under an energy constraint), although the pseudocode is omitted due to space constraints. This entire process is repeated multiple times, until the entire image's accuracy value meets the specified constraint.

As mentioned earlier, the water-filling approach is especially suited for images characterized by greater uniformity in the spatial distribution of objects. Our main idea on this solution can be described as: the resolution adjustment starts from the region with the lowest resolution. This region is considered as the bucket with the lowest predicted mAP. The objective is to maximize the average water level, i.e., try to equalize the water level of all the buckets (regions). Ideally, we want to achieve approximately the same level of detection accuracy with lower energy consumption for the whole image. The proposed water-filling algorithm for resolution

Algorithm 3: Water-Filling Resolution Adjustment With Accuracy Requirement

```
Input: $\mathbb{N}_d$, $\mathbb{N}_r$, bounding boxes, $\mathcal{S}_i|_{1\leq i\leq \mathbb{N}}$, $\overline{\text{mAP}}$, $\mathcal{V}_i|_{1\leq i\leq \mathbb{N}}$;
Output: $\mathcal{V}_i|_{1\leq i\leq N}$, $\mathcal{R}_i|_{1\leq i\leq N}$, Est_Eng;
Cur_ACC = $\frac{\sum_{i=1}^{\mathbb{N}} \mathcal{S}_i}{\mathbb{N}}$;
if $Cur\_ACC < \overline{mAP}$ then
    for each region $\mathcal{R}_i$ do
        Cur_ACC$_{\mathcal{R}_i}$ = $\frac{\sum_{j=1}^{\mathbb{N}} \mathcal{S}_{\mathcal{R}_i}}{\mathbb{N}_{\mathcal{R}_i}}$;
        while $Cur\_ACC_{\mathcal{R}_i} < \mathbb{mAP}$ do
            $\mathcal{V}_i = \mathcal{V}_i + \Delta\mathcal{V}$;
            Cur_ACC = $\frac{\sum_{i=1}^{\mathbb{N}} \mathcal{S}_i}{\mathbb{N}}$;
            $E_i(\mathcal{V}_i) = E_i^{\text{idle}}(\mathcal{V}_i) + E_i^{\text{tran}}(\mathcal{V}_i)$;
        end
    end
end
Est_Eng = $\sum_{i=1}^{N} E_i(\mathcal{V}_i)$;
return $\mathcal{V}_i|_{1\leq i\leq N}$, Est_Eng;
```

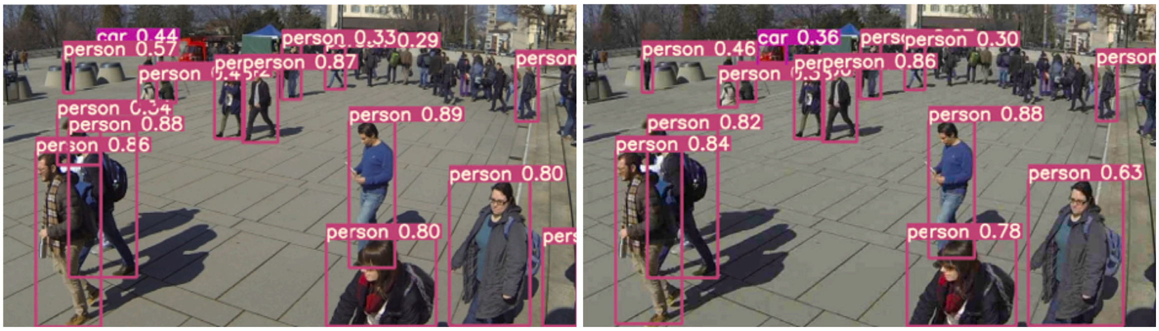
adjustment with regard to accuracy requirement is presented in Algorithm 4. We start by assuming the available resolution values as zero and then gradually increasing these values under the energy constraint.

Algorithm 4: Water-Filling Resolution Adjustment With Energy Constraint

```
   Input: $\mathbb{N}_d$, $\mathbb{N}_r$, bounding boxes, $\mathcal{S}_i|_{1\leq i\leq \mathbb{N}}$, $\mathbb{E}$, $\mathcal{V}_i|_{1\leq i\leq \mathbb{N}}$;
   Output: $\mathcal{V}_i|_{1\leq i\leq N}$, $\mathcal{R}_i|_{1\leq i\leq N}$, Est_Eng;
1  $E_i(\mathcal{V}_i) = E_i^{\text{idle}}(\mathcal{V}_i) + E_i^{\text{process}}(\mathcal{V}_i) + E_i^{\text{tran}}(\mathcal{V}_i)$;
2  Est_Eng = $\sum_{i=1}^{\mathbb{N}_r} E_i(\mathcal{V}_i)$;
3  $\mathbb{V} = 0$;
4  while $Est\_Eng < \mathbb{E}$ do
5      $\mathcal{V}_i = \mathcal{V}_i + \Delta\mathcal{V}$;
6      Est_Eng = $\sum_{i=1}^{N} E_i(\mathcal{V}_i)$;
7      $\mathbb{V} = \mathbb{V} + \Delta\mathcal{V}$;
8  end
9  Cur_ACC = mAP = $\frac{\sum_{i=1}^{\mathbb{N}} \mathcal{S}_i}{\mathbb{N}}$;
10 for each region $\mathcal{R}_i$ do
11     Cur_ACC$_{\mathcal{R}_i}$ = $\frac{\sum_{j=1}^{\mathbb{N}} \mathcal{S}_{\mathcal{R}_i}}{\mathbb{N}_{\mathcal{R}_i}}$;
12     while $Cur\_ACC_{\mathcal{R}_i} < Cur\_ACC$ do
13         $\mathcal{V}_i = \mathcal{V}_i + \Delta\mathcal{V}$;
14         Cur_ACC = $\frac{\sum_{i=1}^{\mathbb{N}} \mathcal{S}_i}{\mathbb{N}}$;
15         $\mathbb{V} = \mathbb{V} - \Delta\mathcal{V}$;
16     end
17 end
18 if $\mathbb{V} > 0$ then
19     $\mathcal{V}_i = \mathcal{V}_i + \frac{\mathbb{V}}{\mathbb{N}}$;
20 end
21 return $\mathcal{V}_i|_{1\leq i\leq N}$, Est_Eng;
```

We now provide a more elaborate explanation of the steps in Algorithm 4.

- Line 1–2: Estimate the current total energy consumption by adding per-region energy values, i.e., the cumulative sum of baseline, processing and transmission energy values.
- Line 4–8: Calculate the available resolution values (the fluid in the water-filling process) that can be assigned to different regions based on the current estimated energy and energy constraint.
- Line 9: Estimate the current mean average precision.
- Line 10–17: The for loop to adjust the regional resolution values under energy constraint.



(a) high energy constraint

(b) low energy constraint

Fig. 9. Impact of energy constraint on person detection accuracy in Algorithm 4.

- Line 12–16: The while loop to gradually increase a region's resolution using fixed step size until the regional mAP reaches the image-level mAP.
- Line 18–20: Assign the remnants of the available resolution values to the whole image.

Fig. 9 plots the mixed-resolution images resulting from the use of Algorithm 4, under high and low energy constraints respectively. We observe that the tighter constraint on image size in Fig. 9b results in a lower resolution image, which in turn decreases the confidence values reported by the object detector. Consequently, the mAP in Fig. 9a is 0.76, while Fig. 9b results in a lower mAP of 0.68.

5. Performance evaluation

We evaluate the performance of our proposed algorithms, both in terms of its (a) the resulting object detection task accuracy (mAP), and (b) the actual energy and latency overheads on real-world embedded vision sensor platforms. Accuracy performance is evaluated by replacing images from multiple benchmark public datasets. For the energy and latency metrics, we evaluate *MRIM* using the *Raspberry Pi (RPi) 3B*. RPi 3B is a popular embedded platform, equipped with Imx219 image sensor, BCM2837, a quad-core 1.2 GHz ARM-Cortex processor, 1 GB of LPDDR2 SDRAM and an onboard BCM43438 chip supporting an 802.11ac radio. Power measurements, both system-level and for individual functional components, are obtained via the use of the Monsoon power monitor. We have also conducted the energy profiling using the low-power ESP32 CAM development board, which is equipped with two 32-bit LX6 CPUs, 512 kB SRAM, 4 MB PSRAM and OV2640 image sensor. The power measurements are performed using the Nordic power profiler Kit II, which supports a current measurement range of 200 nA–1 A, at 100 ksps sampling rate.

5.1. Alternative baselines

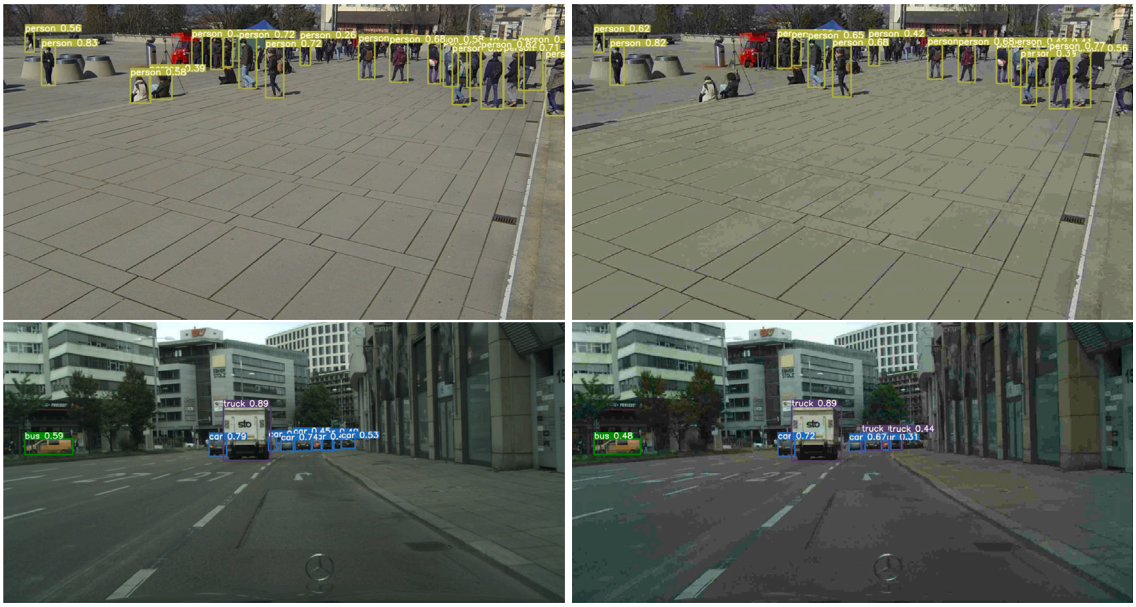
We compare *MRIM's* differential downscaling approach (via either Max-Min or Water-filling algorithms) with baselines of:

- *Compressive Image Coding:* We utilize the JPEG encoder of Python 3.8 OpenCV libraries, and modify the compression quality values (higher value → higher quality) within the range $(100, \dots, 20)$ to execute different levels of compression. Specifically, the image encoding function used in the experiments is the cv2.imencode in OpenCV 4.7.2 [35]. This compression module is able to encode input image data into a memory buffer. The compression quality range for this function is $[0, 100]$. In general, higher encoding quality value represents better reconstructed image quality. Fig. 10 illustrates the image quality (and object detection accuracy) resulting from different encoding values; as expected, higher compression loss leads to significant degradation in task accuracy. we also evaluate the performance of an alternative encoding technique – Pillow (PIL) encoding – under varying levels of encoder quality. The use of two distinct encoders allow us to confirm that our key findings are generalizable and independent of implementation-specific artifacts.
- *Uniform Resolution Downscaling:* In this approach, the entire image is uniformly downscaled to the target resolution, without considering the saliency of different sub-regions.

We compare with image encoding method because this is commonly used in image transmission and storage systems. For *MRIM*, we adopt the LFFD as the sailency estimator, retraining the baseline model for the person and vehicle detection tasks.

5.2. Datasets

To evaluate *MRIM's* effectiveness in preserving task accuracy under different energy/accuracy constraints, we utilize two different public benchmark datasets that are representative of typical pervasive vision applications:



(a) high quality

(b) low quality

Fig. 10. Impact of image quality on person and vehicle detection accuracy.

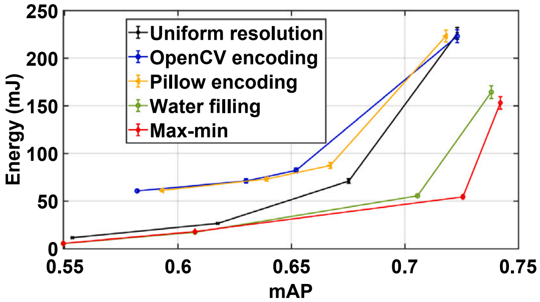


Fig. 11. Energy vs. Accuracy: *MRIM* vs. Alternative baselines (RPi Board).

- *WildTrack:* The WildTrack dataset [11] involves the use of multiple HD 1920 × 1080 cameras (with an average ~55% FoV overlap across cameras) to capture a very crowded public area on a university campus for the purposes of *human object detection*. The images are captured with HD-quality (1920 × 1080) cameras, under conditions of very high crowd density with significant occlusion of individuals.
- *CityScapes:* The CityScapes dataset [12] includes 5000 images, across 27 cities, representing a wide variety of urban environments, and annotated with objects corresponding to 30 classes (e.g., different types of vehicles, construction, human). In our studies, we utilize a subset of 2000 images most suited for the *vehicle object detection* task.

5.3. Energy-accuracy tradeoff

We first study how the total energy of the vision platform, as well as the task accuracy (YOLOv5 based object detection) is affected by different resolution choices, expressed by file size constraints. Fig. 11 plots the average energy consumption vs. mAP, across both WildTrack and CityScapes, as measured using the RPi platform. In addition, Fig. 1c (presented earlier in Section 1) demonstrates the tradeoff between mAP and transmission file size for the different approaches. Overall, we see that both Max-Min and Water-filling outperform the other baselines—in particular, the *MRIM* approaches are able to achieve ≈20% increase in accuracy under identical energy overheads. We also note that, as expected, the mAP of Uniform Downscaling degrades rapidly as the image is downscaled. While Image Encoding can achieve, on average, higher mAP compared to Uniform Downscaling, its energy overhead

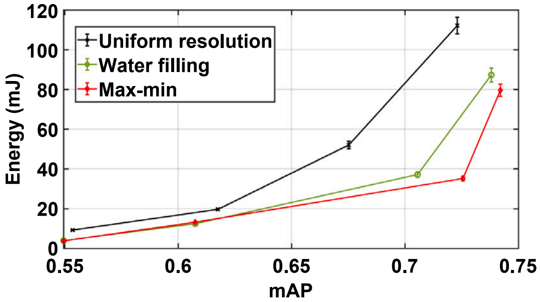


Fig. 12. Energy vs. Accuracy: *MRIM* vs. Alternative baselines (ESP32 Low-Power Board).

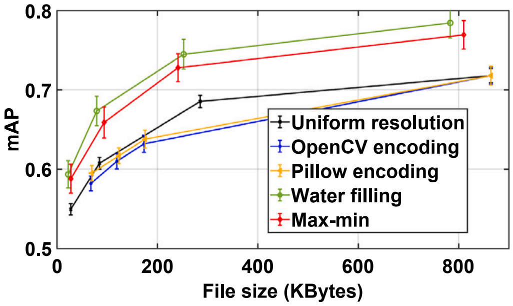


Fig. 13. mAP vs. File size: MRIM vs. Alternative baselines.

is significantly higher compared to all other approaches. The energy vs. accuracy performance of the OpenCV and Pillow encoding algorithms are close to each other.

Low-Power Hardware: While Fig. 11 demonstrates energy savings of ~35% on the RPi device, we note that the per-frame energy is still relatively high (~50 mJ) even for *MRIM*. This is due to the unoptimized RPi platform, which has a fairly high baseline power of 230 mW. To study the potential advantage of *MRIM* on more power-efficient, optimized hardware, we also compute the energy vs. accuracy performance using the low-power ESP32 development board. Fig. 12 illustrates the energy vs. accuracy performance (as the processing logic remains unchanged, the accuracy numbers are identical to Fig. 11). We see that *MRIM* achieves not only in appreciable *relative* energy savings (more than 3x energy reduction compared to a uniform resolution baseline), it also results in much lower *absolute* energy values under high-accuracy regimes (<40 mJ for mAP > 0.7).

Fig. 13 presents the mAP versus image file size for all the evaluation methods. As *MRIM's* performance depends on the choice of N_r (number of regions into which the original image is partitioned), the results presented here denote the mean values obtained as N_r is varied between $(2, \dots, 16)$. The largest image file size represents the original image (no reduction in resolution). Across all image sizes, our proposed algorithms outperform both the baseline schemes. As the results indicate, our proposed mixed-resolution framework is more flexible than the fixed full or small resolution in terms of reducing energy or increasing accuracy.

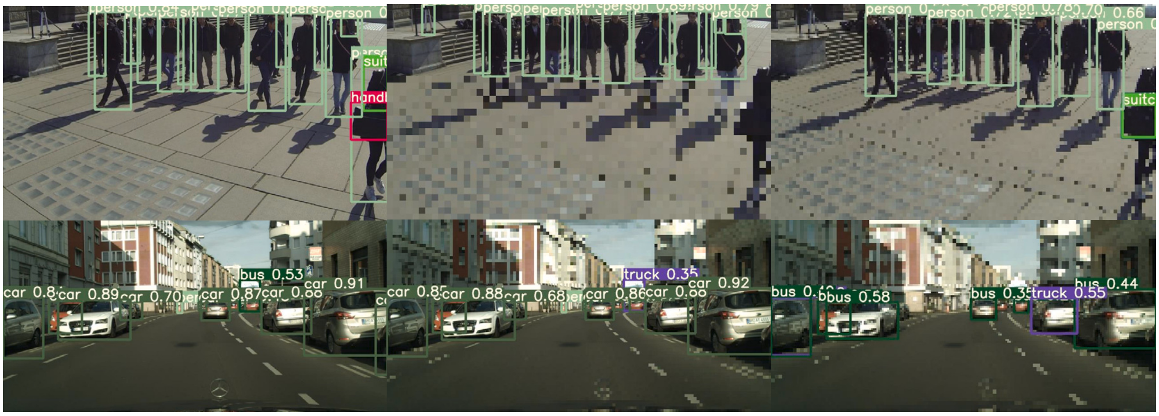
5.4. Max-min vs. Water-filling

To further study the differences between the two algorithms presented in Section 4, Table 2 plots the energy-vs.-mAP variation for two different scenarios: (i) Water-Filling applied to objective **P1** and (ii) Max-Min applied to objective **P2**, when the raw camera image has a resolution of $1350 * 900$. We can see that both approaches are able to dramatically reduce the overall file size (by ≈80%–65%, compared to an original image size of ~1.2 MB) with only a modest 3.6% loss of accuracy (to ~0.68 from the maximum value of 0.73, achieved when the raw image is input to the DNN). *MRIM* can thus achieve a significant (almost 3-fold) reduction in system energy, even though the RPI's non-negligible baseline power of 230 mW presents a challenge to true energy-proportionality.

Additionally, Fig. 14 illustrates the resulting compressed images (and the object detection output) under both schemes, for one representative image from each dataset. We can see that Max-Min is more aggressive in reducing the resolution for less salient regions (notice the increased blockiness of the foreground areas for WildTrack), whereas Water-filling tends to preserve greater detail for such lower-saliency regions. Accordingly, we believe that the choice of the executed algorithm will depend on the deployment-specific object spatial distribution characteristics.

5.5. Delay & energy characteristics

To further illustrate the appeal of *MRIM's* computationally-efficient, yet effective, technique of image downsampling, Table 3 plots the variation in file size/energy consumption and processing latency (on the RPi) as the degree of compression is varied.



(a) Original Images (b) Max-Min Algorithm (c) Water-Filling Algorithm

Fig. 14. Original vs. *MRIM*-based images.

Table 2
Water-filling vs. Max-Min performance.

Water filling (accuracy-aware)			
Index	File size (kB)	Total (mJ)	Target mAP
1	472.8	107.52	0.673
2	119.5	27.62	0.642
3	52.09	10.83	0.568
4	17.6	3.472	0.526
Max–min (energy constraint)			
Index	File size (kB)	Target energy (mJ)	mAP
1	715.3	156.878	0.687
2	183.5	43.836	0.663
3	69.07	13.93	0.593
4	18.79	3.854	0.539

Table 3
MRIM vs. Image encoding.

Resolution adjustment			
Resolution	File size (kB)	Total (mJ)	Latency (ms)
1350 ∗ 900	736	156.9	196
775 ∗ 450	219	43.8	50.8
338 ∗ 225	69.07	13.9	14.3
225 ∗ 75	19	3.9	3.56
Image encoding			
Jpeg quality	File size (kB)	Total (mJ)	Latency (ms)
100	736	156.9	196
80	143.4	54.4	145
60	96.43	46.2	141
20	51.87	39.6	134

We see that while compressive encoding can indeed reduce the file size significantly, the additional on-board processing overhead dampens the reduction in overall latency and energy, relative to *MRIM*. For example, under compressive encoding, a file size of 96 kB incurs a processing latency of 141 ms; in contrast, with *MRIM*, a reduced file size of 69 kB incurs a processing latency of only 3.5 ms. (implying a maximum possible throughput of ~300 fps). Overall, our results indicate that *MRIM* can achieve ~3–40x reduction in per-frame processing latency compared to image encoding, thereby providing a compelling, computationally-efficient mechanism to reduce the transmission bandwidth and energy overhead.

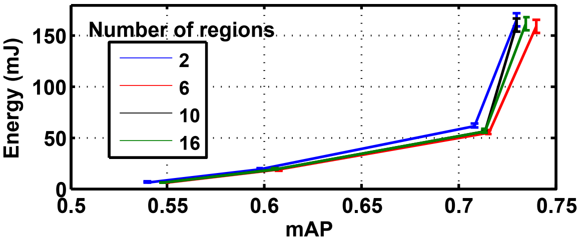


Fig. 15. Energy vs. Accuracy (Varying $\mathbb{N}_r$).

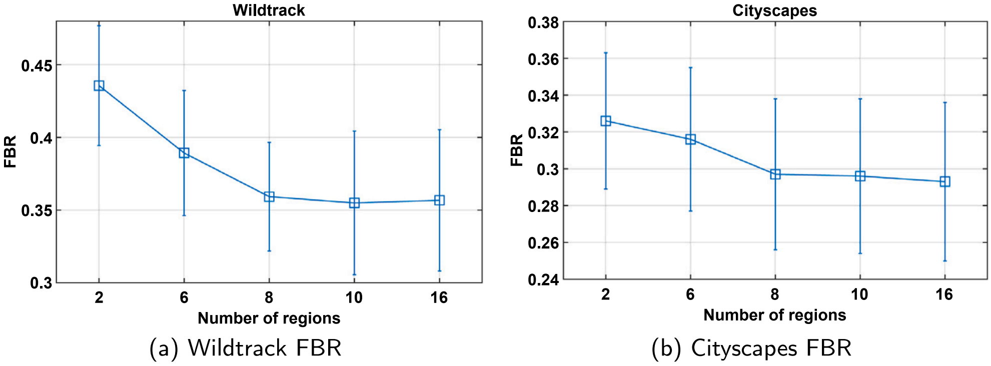


Fig. 16. Foreground to background ratio (FBR) vs. The number of regions (N_r).

6. Extensions and discussion

While our work attests to the power of *MRIM*, there are a few additional enhancements to, and applications of, *MRIM* that we now investigate.

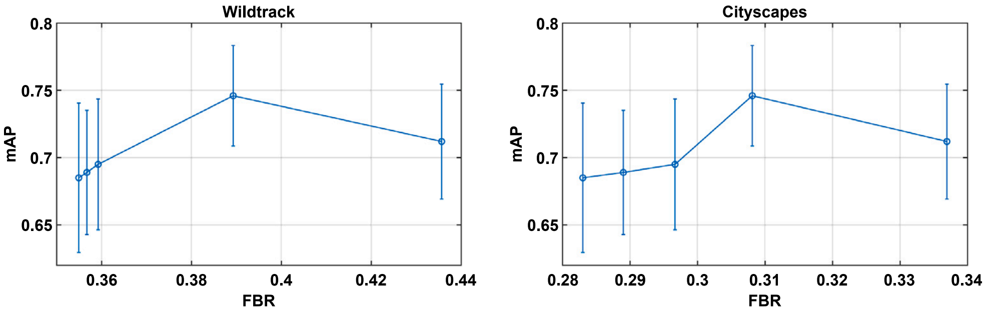
Automated Choice of $\mathbb{N}_r$: The core *MRIM* algorithm takes the number of partitioned sub-regions, N_r, as an input. We experimentally observe that the most suitable values for the number of distinct sub-regions N_r are in the range of $[2, 6, 10, 16]$. Fig. 15 plots the energy vs. mAP variation for different values of $\mathbb{N}_r$. Intuitively, a larger value of $\mathbb{N}_r$ permits more fine-grained resolution adjustment (and thus improved mAP); however, the increased algorithm complexity leads to a larger value of E_{proc}. Indeed, we can see that a choice of $\mathbb{N}_r = 6$ provides higher accuracy at lower energy overheads, compared to $\mathbb{N}_r = 10$ or 16. As the optimal value of $\mathbb{N}_r$ depends on the deployment-specific spatial properties of objects, a fully autonomous version of *MRIM* should empower a specific sensor instance to automatically determine its close-to-optimal choice of $\mathbb{N}_r$.

We now present one approach for automated selection of N_r. Under this approach, a newly deployed image sensor first transmits all captured images at full resolution (i.e., without executing *MRIM*) during an initial, short ‘stabilization’ phase. Such full-resolution, deployment-specific, representative images are then analyzed to obtain attributes related to the locations and density of objects. More specifically, the output of the DNN detector at the edge is used to first observe the distribution of the detected objects’ locations and density.

Next, these object locations and sizes are used to compute the average (across all regions) “Foreground-to-Background” (FBR) ratio, under varying values of N_r. For any region, FBR is computed as the ratio of pixels corresponding to identified objects to the total pixels in that region. In general, a larger value of N_r provides a finer-grained partitioning of the image, as a result of which some regions are likely to have no objects (and thus very low FBR), while other regions consist almost entirely of objects (and thus have high FBR). Consequently, a larger N_r is likely to translate into a smaller mean FBR but a larger variance in FBR values. Fig. 16 plots the variation in FBR vs. N_r, over the training images in Wildtrack and Cityscapes datasets. Consistent with our expectation, we observe that the average FBR generally decreases while the variance (std. deviation) increases with increasing N_r.

Changes in the FBR values are also likely to affect the accuracy of the object detection task. Generally speaking, *MRIM* will perform best at *intermediate* FBR values: at very high FBR (low value of N_r), almost all regions contain a large set of objects and *MRIM* will degenerate to uniform downsampling; conversely, at very low FBR (very high value of N_r), *MRIM* will end up conserving pixels in multiple low-saliency regions. Fig. 17 plots the mAP vs. FBR results for the WildTrack and CityScapes datasets. The figures confirm our hypothesis: broadly, mAP is highest when FBR is ∼0.39 for WildTrack and ∼0.31 for CityScapes, and decreases for both higher and lower FBR values.

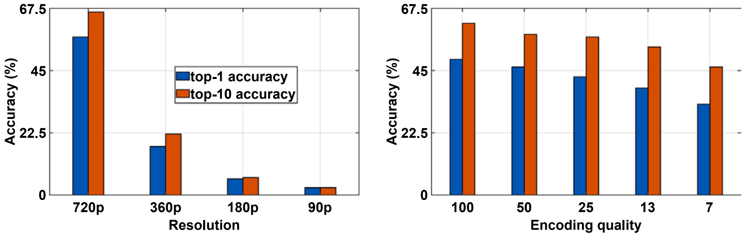
Finally, the combination of these two analyses can be used to determine an appropriate choice of N_r. At a high level, our approach uses the sample images captured during the stabilization phase to empirically determine both the optimal value for FBR (in terms of highest mAP) and, subsequently, the choice of N_r that results in such FBR values. Applying this approach to our representative



(a) Wildtrack FBR

(b) Cityscapes FBR

Fig. 17. mAP vs. Foreground to background ratio (FBR).



(a) count vs. resolutions

(b) count vs. encoding qualities

Fig. 18. Impact of resolution adjustment and image encoding on LPR performance.

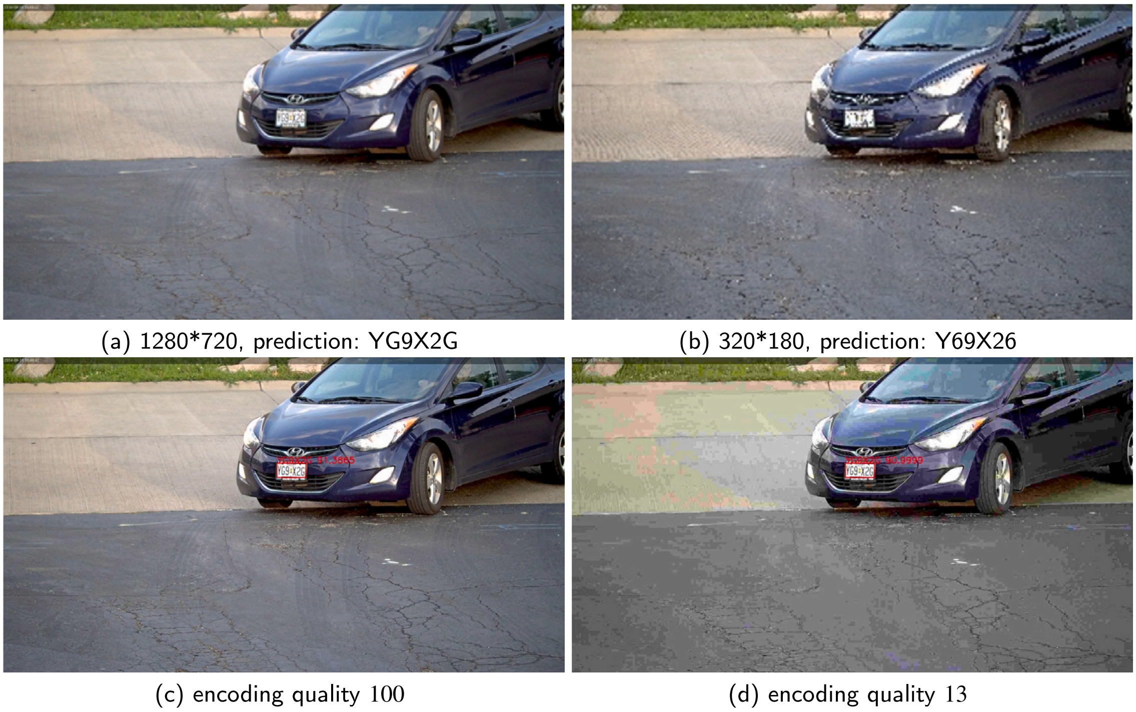
datasets, we observe that $N_r = 6$ and $N_r = 8$ represent the best choices for WildTrack and CityScape, respectively. An automated version of *MRIM* can then initiate the execution of mixed-resolution downsampling, using these choices for N_r, at the end of the stabilization phase.

Extension to Other Vision Tasks: While our primary exposition has considered only the object detection task, we believe that *MRIM's* approach of *preserving resolution differentially for salient regions* has broader applicability across a variety of vision tasks, such as scene classification, action recognition, etc. However, the accuracy-vs.-resolution tradeoff can be discontinuous for different tasks–e.g., it is likely that person identification requires a minimum resolution to operate and any additional reduction in resolution will cause the task accuracy to drop precipitously.

To demonstrate this generalizability, we consider License Plate Recognition (LPR) as an alternative exemplar task. LPR is representative of a broader class of ‘recognition’ tasks, which intuitively will be more suspectible to the loss of high-frequency spatial information that arise from resolution downsampling. We first evaluate the openalpr [36] implementation on 222 images of the openalpr end-to-end benchmark dataset (US category), while first varying either the image resolution (uniformly) or the encoding quality. Openalpr returns 10 predictions with highest confidence values. We compute both the *top-1* accuracy (i.e., the accuracy of the highest confidence output), as well as the *top-10* accuracy (i.e., the probability that any one of the 10 highest confidence predictions is correct). Fig. 18(a) and (b) plot the average accuracy for different values of resolution and encoding quality, respectively. In line with earlier results on YOLOv5-based object detection, lower resolution or lower quality results in a significant drop in LPR accuracy, with LPR accuracy falling below 5% at 180p resolution. To provide a visual feel for these results, Fig. 19 illustrates the visual quality and recognition results for a representative image under different levels of resolution and encoding quality.

To demonstrate the benefit of *MRIM*, we then applied the *MRIM* algorithm on selected images from the dataset, preserving the original resolution around the license plate region while differentially downsizing the non-salient region. We choose resolution reduction scales RS={1,2,4,16}, where a value of RS=rs implies that the corresponding pixels are downsampled by a factor of rs^2 (both width and height reduced rs-fold). Fig. 20 plots the LPR accuracy and file size variations as a function of the *MRIM* scale values. We observe that the LPR accuracy degrades much more slowly as the image is selectively downsampled; the top-1 accuracy remains >40% even when $rs = 4$, corresponding to an overall image file size reduction of ≈50%. To provide a visual feel of the result, Fig. 21 illustrates the corresponding *MRIM* images. Collectively, these results demonstrate that *MRIM* indeed offers a superior accuracy-vs.-file size tradeoff for the LPR task, compared to both saliency-agnostic uniform downsampling or image encoding.

Tradeoff for Low Power Vision Platforms: We chose the RPi platform for our experiments primarily for expediency (easily available, open source drivers, etc.). However, the RPi's baseline/idle power of 230 mW is significantly higher than other specialized

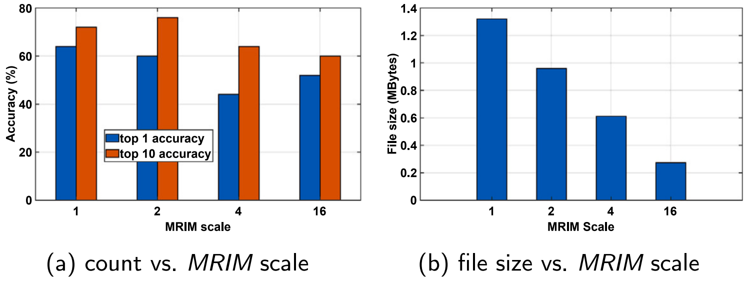

(a) 1280*720, prediction: YG9X2G

(b) 320*180, prediction: Y69X26

(c) encoding quality 100

(d) encoding quality 13

Fig. 19. Visual quality differences on the LPR images.



(a) count vs. *MRIM* scale

(b) file size vs. *MRIM* scale

Fig. 20. Impact of *MRIM* scale on LPR performance.

ultra-low power platforms (such as the Pixy2[1]), making the system far from energy-proportional [10] and thus limiting the gains exhibited by *MRIM*. As evidenced by our evaluation of the ESP32 vision board, *MRIM* can continue to achieve substantial energy reduction (∼60%–70%), especially when attempting to preserve higher accuracy (mAP > 0.7). Of course, on such ultra-low power micro-controller based hardware, the per-frame processing time also increases, leading to a greater proportion of the energy budget being consumed by on-board image processing (as opposed to data transmission), as well as a lower frame throughput.

Video Encoding and Mixed-Resolution Adjustment: *MRIM* currently performs per-image differential resolution reduction, before transmitting each such mixed resolution image (without any further encoding) wireless to the edge inferencing platform. As future work, it will be interesting to evaluate mechanisms that integrate *video* compression techniques with the proposed *MRIM* mechanisms. There at least two distinct approaches for such integrated, *saliency-aware video processing*:

- Combine *MRIM's* per-region saliency estimator, together with motion-based compression techniques, to perform saliency estimation along both spatial and temporal axes. In this approach, the video encoding quality can be dynamically adjusted, separately for each sub-region, to account for differences in spatiotemporal saliency values across regions.
- Pipeline *MRIM's* saliency-aware blurring (downsampling) mechanism together with an off-the-shelf, state-of-the-art video encoder, so as to further reduce the video transmission bandwidth with negligible impact on video quality. To perform an

[1] CMUCam5 (Pixy-2 platform) URL- http://www.cmucam.org/projects/cmucam5.

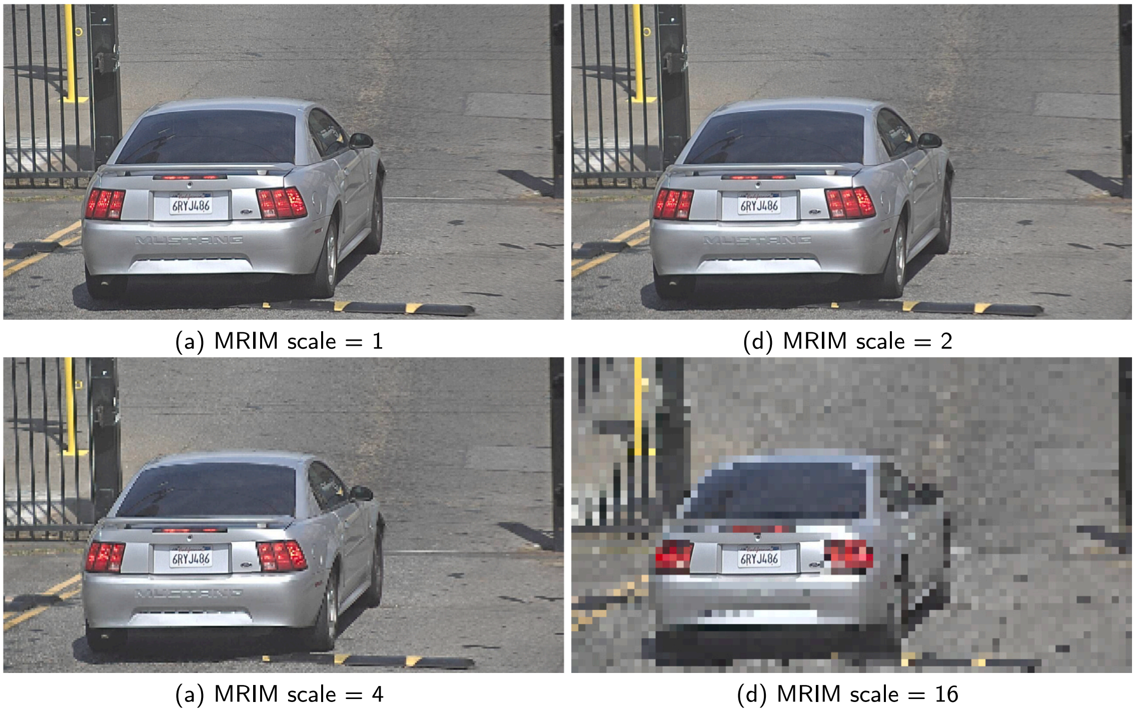


Fig. 21. Visual quality differences on the MRIM images.

Table 4
Original and compressed file size comparison.

mAP constraint	75%	80%	85%	N.A.
Original file size (MB)	27.7	28.7	29.5	33.5
Compressed file size (MB)	1.39	1.93	2.3	2.57

initial feasibility test of this concept, we have performed tests to evaluate the relative bandwidth (file size) requirements between (a) *MRIM's* image-based compression strategy, and (b) an alternative approach where the *stream* of *MRIM*-compressed images are encoded into H.264 format using default CRF=23 values. Table 4 plots the file size variations, for both approaches, using representative original video samples selected from the WildTrack data-set. The software video codec utilized is FFMPEG 5.1.3 [37]. The mixed-resolution images are generating by setting different target mAP constraints for *MRIM's* Water-Filling algorithm. We observe that joint *MRIM* +H.264 encoding can almost halve transmission file size (~46% reduction from 2.57 MB to 1.39 MB). In contrast, image-level encoding provides a much more modest 17% file size reduction (33.5 MB to 27.7 MB). This result suggests that *MRIM's* mixed-resolution processing can indeed be combined with video encoding, at least for streaming vision applications, to achieve very significant bandwidth and transmission energy savings.

However, additional studies will be needed to quantify the overall quality-vs.-energy tradeoffs, especially as video encoding typically involves significant additional on-board processing and the most optimal strategy may depend on the specific hardware resources on the vision sensor platform.

7. Conclusion

MRIM introduces an approach for performing differential resolution downscaling on different regions of a single image, so as to reduce the overall energy overheads of pervasive vision sensing without compromising the accuracy of DNN-based vision tasks. We have described simple, but effective and computationally efficient, techniques for both dynamically estimating the saliency of different sub-regions of a single captured image, and subsequently determining the resolution values for each sub-region. Via the use of multiple benchmark urban monitoring datasets and an RPi-based implementation, we have demonstrated that *MRIM* can consume ~30% less energy than even a hardware-optimized image encoder and achieve ~20% improvement in object detection accuracy over a comparable uniform resolution downscaling approach—the savings are even more significant (~80%) at high mAP values. On a lower-power, ESP32 vision board, *MRIM* continues to provide significant (60+%) energy reduction per frame, compared to a baseline uniform downsampling technique. Overall, *MRIM* offers a new approach that uses lightweight computation on an embedded sensor platform as a means of flattening the energy-vs.-accuracy curve. Future work includes the judicious integration of this mixed-resolution approach with temporal adaptation concepts from video coding, to further minimize the energy overheads on pervasive vision sensor platforms.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Research Foundation, Singapore under its NRF Investigatorship grant (NRFNRFI05-2019-0007). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

References

[1] J.-Y. Wu, V. Subasharan, T. Tran, A. Misra, MRIM: Enabling mixed-resolution imaging for low-power pervasive vision tasks, in: 2022 IEEE International Conference on Pervasive Computing and Communications (PerCom), IEEE, 2022, pp. 44–53.

[2] N. Bicocchi, M. Lasagni, F. Zambonelli, Bridging vision and commonsense for multimodal situation recognition in pervasive systems, in: 2012 IEEE International Conference on Pervasive Computing and Communications, IEEE, 2012, pp. 48–56.

[3] T. Kumrai, J. Korpela, T. Maekawa, Y. Yu, R. Kanai, Human activity recognition with deep reinforcement learning using the camera of a mobile robot, in: 2020 IEEE International Conference on Pervasive Computing and Communications (PerCom), IEEE, 2020, pp. 1–10.

[4] C. Bermejo, D. Chatzopoulos, P. Hui, EyeShopper: Estimating shoppers' gaze using CCTV cameras, in: Proceedings of the 28th ACM International Conference on Multimedia, MM '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 2765–2774.

[5] A. Skordylis, N. Trigoni, Efficient data propagation in traffic-monitoring vehicular networks, IEEE Trans. Intell. Transp. Syst. 12 (3) (2011) 680–694.

[6] L. Liu, H. Li, M. Gruteser, Edge assisted real-time object detection for mobile augmented reality, MobiCom '19, Association for Computing Machinery, New York, NY, USA, 2019.

[7] J. Yi, S. Choi, Y. Lee, EagleEye: Wearable camera-based person identification in crowded urban spaces, in: Proceedings of the 26th Annual International Conference on Mobile Computing and Networking, MobiCom '20, Association for Computing Machinery, New York, NY, USA, 2020.

[8] J. Hester, J. Sorber, The future of sensing is batteryless, intermittent, and awesome, in: Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems, SenSys '17, Association for Computing Machinery, New York, NY, USA, 2017.

[9] J. Hu, A. Shearer, S. Rajagopalan, R. LiKamWa, Banner: An image sensor reconfiguration framework for seamless resolution-based tradeoffs, in: Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services, 2019, pp. 236–248.

[10] R. LiKamWa, B. Priyantha, M. Philipose, L. Zhong, P. Bahl, Energy characterization and optimization of image sensing toward continuous mobile vision, in: Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services, 2013, pp. 69–82.

[11] T. Chavdarova, P. Baqué, S. Bouquet, A. Maksai, C. Jose, T. Bagautdinov, L. Lettry, P. Fua, L. Van Gool, F. Fleuret, Wildtrack: A multi-camera hd dataset for dense unscripted pedestrian detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5030–5039.

[12] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3213–3223.

[13] M. Rahimi, R. Baer, O.I. Iroezi, J.C. Garcia, J. Warrior, D. Estrin, M. Srivastava, Cyclops: In situ image sensing and interpretation in wireless sensor networks, in: Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems, SenSys '05, ACM, 2005.

[14] P. Kulkarni, D. Ganesan, P. Shenoy, Q. Lu, SensEye: A multi-tier camera sensor network, in: Proceedings of the 13th Annual ACM International Conference on Multimedia, MULTIMEDIA '05, ACM, 2005.

[15] S. Naderiparizi, P. Zhang, M. Philipose, B. Priyantha, J. Liu, D. Ganesan, Glimpse: A programmable early-discard camera architecture for continuous mobile vision, in: Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys '17, Association for Computing Machinery, 2017, pp. 292–305.

[16] S. Naderiparizi, A.N. Parks, Z. Kapetanovic, B. Ransford, J.R. Smith, WISPCam: A battery-free RFID camera, in: 2015 IEEE International Conference on RFID (RFID), 2015, pp. 166–173.

[17] S. Naderiparizi, M. Hessar, V. Talla, S. Gollakota, J.R. Smith, Towards battery-free HD video streaming, in: Proceedings of the 15th USENIX Conference on Networked Systems Design and Implementation, NSDI '18, USENIX Association, 2018, pp. 233–247.

[18] M. Xu, X. Zhang, Y. Liu, G. Huang, X. Liu, F.X. Lin, Approximate query service on autonomous IoT cameras, in: Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services, Association for Computing Machinery, 2020, pp. 191–205.

[19] T. Dao, K. Khalil, A.K. Roy-Chowdhury, S.V. Krishnamurthy, L. Kaplan, Energy efficient object detection in camera sensor networks, in: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), 2017, pp. 1208–1218, http://dx.doi.org/10.1109/ICDCS.2017.152.

[20] A. Zam, M.R. Khayyambashi, A. Bohlooli, Energy-aware strategy for collaborative target-detection in wireless multimedia sensor network, Multimedia Tools Appl. 78 (13) (2019) 18921–18941, http://dx.doi.org/10.1007/s11042-019-7204-5.

[21] Y. Hu, S. Liu, T. Abdelzaher, M. Wigness, P. David, On exploring image resizing for optimizing criticality-based machine perception, in: 2021 IEEE 27th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), 2021, pp. 169–178.

[22] S.-P. Lu, S.-M. Li, R. Wang, G. Lafruit, M.-M. Cheng, A. Munteanu, Low-rank constrained super-resolution for mixed-resolution multiview video, IEEE Trans. Image Process. 30 (2020) 1072–1085.

[23] T. Richter, J. Seiler, W. Schnurrer, A. Kaup, Robust super-resolution for mixed-resolution multiview image plus depth data, IEEE Trans. Circuits Syst. Video Technol. 26 (5) (2015) 814–828.

[24] B. Mallik, A. Sheikh-Akbari, A.-L. Kor, Mixed-resolution HEVC based multiview video codec for low bitrate transmission, Multimedia Tools Appl. 78 (6) (2019) 6701–6720.

[25] S. Jiang, Z. Lin, Y. Li, Y. Shu, Y. Liu, Flexible high-resolution object detection on edge devices with tunable latency, in: Proceedings of the 27th Annual International Conference on Mobile Computing and Networking, 2021, pp. 559–572.

[26] I. Gokarn, H. Sabbella, Y. Hu, T. Abdelzaher, A. Misra, MOSAIC: Spatially-multiplexed edge AI optimization over multiple concurrent video sensing streams, in: Proceedings of the 14th Conference on ACM Multimedia Systems, MMSys '23, Association for Computing Machinery, 2023, pp. 278–288, http://dx.doi.org/10.1145/3587819.3590986.

[27] A. Bahadur, Haar feature-based light-weight detector, 2017, https://github.com/akshaybahadur21/FaceDetection.

[28] Y. He, D. Xu, L. Wu, M. Jian, S. Xiang, C. Pan, LFFD: A light and fast face detector for edge devices, 2019, arXiv preprint arXiv:1904.10633.

[29] H. Peng, S. Yu, A systematic iou-related method: Beyond simplified regression for better localization, IEEE Trans. Image Process. 30 (2021) 5032–5044.

[30] Y. Yang, FastMOT: High-performance multiple object tracking based on deep SORT and KLT, 2020, http://dx.doi.org/10.5281/zenodo.4294717.

[31] R. Lee, S.I. Venieris, L. Dudziak, S. Bhattacharya, N.D. Lane, MobiSR: Efficient on-device super-resolution through heterogeneous mobile processors, in: The 25th Annual International Conference on Mobile Computing and Networking, MobiCom '19, Association for Computing Machinery, 2019.

[32] J. Redmon, A. Farhadi, YOLOv3: An incremental improvement, 2018, CoRR abs/1804.02767. URL: http://arxiv.org/abs/1804.02767.

[33] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Vol. 1, 2001.

[34] G. Jocher, ultralytics/yolov5: v3.1 - bug fixes and performance improvements, 2020, http://dx.doi.org/10.5281/zenodo.4154370, https://github.com/ultralytics/yolov5.

[35] G. Bradski, The OpenCV Library, Dr. Dobb's J. Softw. Tools.

[36] A.K. Matthew Hill, Open automatic license plate recognition, 2019, https://github.com/openalpr.

[37] S. Tomar, Converting video formats with ffmpeg, Linux J. 2006 (146) (2006) 10.