EcoVis: Towards Energy and Connectivity Optimized Visual Surveillance

Manoj Kumar Lenka and Ayon Chakraborty
Sensing and Networked Systems Engineering (SeNSE) Lab, IIT Madras
{cs22s008, ayon}@cse.iitm.ac.in

***Abstract*—Visual Analytics Pipelines (VAPs) for real-time surveillance are resource-intensive, consuming high energy and bandwidth. We propose EcoVis, a novel approach that uses mmWave sensing to identify regions of interest (ROIs) for compressing surveillance video frames. This reduces the transmission of static background, optimizing resource usage while maintaining high fidelity of content relevant in traffic surveillance. Unlike conventional methods that rely solely on video frames to detect ROIs, our use of mmWave range-azimuth maps achieves a comparable reduction in network bandwidth while lowering energy consumption by approximately 40%. Moreover, our approach enhances energy efficiency by nearly another 25%, by dynamically controlling the sleep cycle of the camera. For simpler tasks such as vehicle detection or counting, EcoVis works with minimal reliance of the camera. Due to its lower dimensionality compared to video, it allows on-device processing, improving operational speed and network efficiency by roughly 20%. Finally, we introduce both uniform and non-uniform tiling algorithms, utilizing RoIs derived from mmWave analysis. These algorithms enable video encoding with tile-specific Quantization Parameters (QPs), optimizing the overall compression process.**

I. Introduction

With smart camera networks now pervasive, the demand for real-time *Video Analytics Pipelines* (VAPs) has grown significantly. For instance, city-wide traffic surveillance cameras continuously stream high-definition video feeds to edge computing infrastructure for real-time inference tasks. Such tasks range from basic applications like vehicle or pedestrian counting, or speed monitoring to more complex ones such as automatic number plate recognition, detecting unsafe driving behaviors and more. With the average number of camera installations reaching several hundreds per square kilometer, particularly in urban areas [1], the networking and compute infrastructure needed to support these deployments becomes extremely demanding. Even with state-of-the-art video codecs such as H.265 [2], we observe and validate that the *uplink* bandwidth requirement per camera typically exceeds 10 Mbps (e.g., 3 MP – 5 MP resolution at 15–30 fps) [3] leading to excessive broadcast traffic. In scenarios with relatively unstable 4G/5G uplinks, this traffic often contributes to a severe resource crunch for ISPs striving to maintain Quality of Service (QoS). Additionally, the VAPs hosted on the edge servers face scalability challenges with continuous and complex neural processing workloads on streaming video feeds.

Suboptimal Resource Usage. We identify several key challenges that hinder scaling of such camera networks without encountering significant resource constraints. First, the network streaming continues regardless of the scene context or any spatio-temporal redundancy present across the frames. Although the latest video codec standards (e.g., H.265) support intelligent compression mechanisms, such as spatial tiling with tile-specific quantization, leveraging such features, for instance, requires identifying regions of interest (ROIs) within a frame. This process involves keeping the camera active and using moderately heavy computation, e.g., deep neural networks (DNNs), to predict ROIs across frames. Second, running complex DNN inference models (e.g., Visual Transformers [4], [5] or Implicit Neural Representations [6]) involves considerable compute and energy requirements. This limits the feasibility of using battery-powered surveillance cameras, which would otherwise offer the flexibility for on-demand deployments, critical for various public safety scenarios.

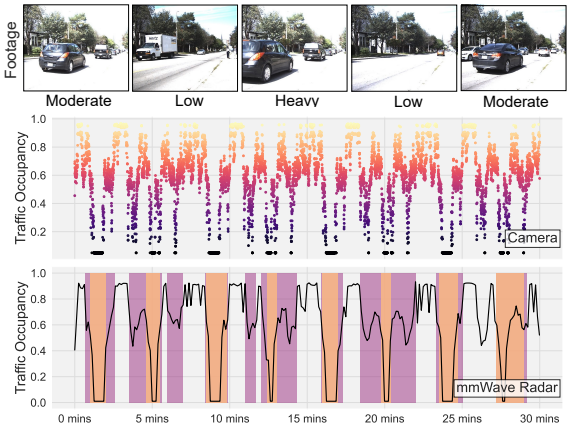


Fig. 1: Traffic occupancy estimation using visual feeds and mmWave sensing show comparable performance more than 75% of the time. The cumulative periods of inactivity (≈30%, *brown patches*, bottom figure) in the 30-minute timeline highlight opportunities to employ mmWave-triggered sleep cycles for the camera. Moreover, for moderate traffic conditions mmWave based sensing (*violet patches*, bottom figure) can be fused with camera to detect RoI in image frame and enable energy efficient compression

In this paper, we leverage wireless sensing to address the above challenges. Specifically, we design and implement EcoVis that uses millimeter waves (mmWave) as an out-of-the-band and lightweight sensing modality to infer potential ROIs, particularly targeting road traffic surveillance applications. In fig. 1, we present a 30-minute timeline of mid-day road traffic captured simultaneously with a surveillance camera along with a low-cost, low-power mmWave radar. A vehicular traffic occupancy metric ($\in [0, 1]$) is independently estimated on both the camera and radar frames. Note that the radar-based estimation is comparable to the camera-based estimation

for approximately estimating traffic occupancy. First, EcoVis facilitates automatic scheduling of sleep cycles for the camera sensor during periods of low activity. Fig. 1 (*brown patches*, bottom) highlights such periods (≈30%), where the occupancy metric is very low – indicating sparse or no traffic. Second, based on the application requirements, EcoVis performs intelligent fusion across the wireless and visual modalities, significantly improving resource consumption. For tasks like vehicle counting or traffic flow monitoring, mmWave-sensing delivers appreciable accuracy (≈95%), without camera involvement, even when the occupancy metric is high, as shown in fig. 1 (*violet patches*, bottom). Furthermore, for applications that require visual imagery (e.g., number plate recognition), our approach is highly effective in identifying regions of interest within the camera's viewport and preserving these areas while applying compression to the rest of the frame.

At the core of EcoVis is the mmWave radar's range-azimuth map, $\mathsf{RA}_{\mathsf{map}}$, that characterizes the approximate depth of objects/reflectors in the scene corresponding to its coverage area. By filtering static clutter, we improve the map's signal-to-noise ratio (SNR) and identify relevant targets, such as vehicles. These targets or Regions of Interest (ROIs) are then mapped from the radar's $\mathsf{RA}_{\mathsf{map}}$ to the camera's field of view, FOV (see §III-B). While the spatial mapping between the $\mathsf{RA}_{\mathsf{map}}$ and FOV can be analytically modeled, we observe that factors like lens aberrations, camera and radar's 3D pose, weather conditions, and radar-induced speckle noise complicate the process of accurately parameterizing the transformation. This limitation is coupled with the lower radar frame rate (≈ 5–10 Hz) compared to the camera's 30+ fps [7], [8]. Instead, we tackle this challenge using a lightweight neural network, a multilayer perceptron ($\mathsf{MLP}_{\mathsf{RoI}}$), trained on deployment-specific data. The training process completes in a few hundred seconds on an edge device along with a amortized real-time inference latency of ≈ 50 ms on a Raspberry Pi 4B, all without requiring GPU support. After inferring ROIs over a time window, EcoVis tiles each frame and compresses pixels within each tile using a corresponding quantization parameter (QP). EcoVis interfaces with the H.265 video encoder, passing such tiling and quantization data, reducing network load and computation. Such content-aware and lightweight compression simplifies both local processing and edge inference. EcoVis also excels in challenging conditions like heavy rain, fog, low light, and other visual impairments where traditional camera-based systems struggle, as the integration of mmWave sensing enables reliable operation even in these adverse environments [9]. Overall, we make the following contributions in this paper:

- We propose a lightweight hybrid surveillance system that integrates mmWave-based sensing with conventional camera setups, significantly reducing computational overhead compared to traditional vision-only approaches.
- We show that our approach reduces energy consumption by ≈ 50% compared to vision-only systems, while also reducing network bandwidth usage by ≈ 60% without compromising analytics accuracy, making it ideal for efficient, sustainable surveillance in areas with limited infrastructure or power.
- We design and implement a complete system prototype and deploy it on a busy street in *Chennai* (Indian metro city) to validate real-world performance. We also evaluate our algorithm on a benchmark dataset, showing its robustness across varied environments and traffic scenarios.

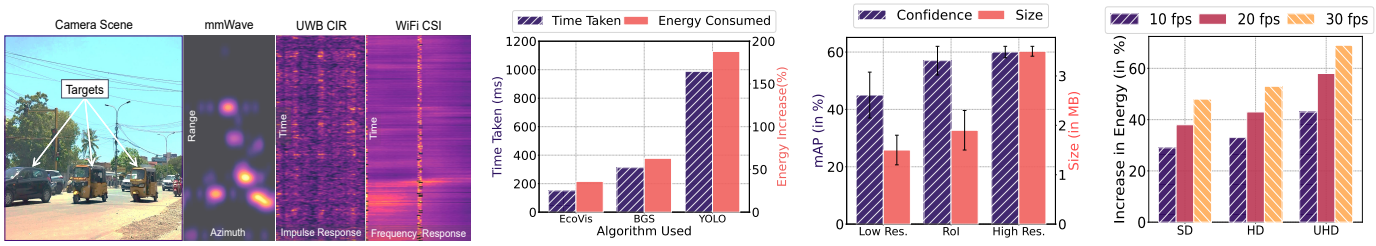
II. Background and Research Gaps

Unlike dynamic video content, surveillance footage typically features static backgrounds and predictable patterns, making it an ideal candidate for advanced compression techniques that can significantly reduce bandwidth usage and on-device processing needs improving energy usage.

A. Related Works on Bandwidth Aware Video Compression

Video Compression. The state-of-the-art video encoders, AVC (H.264) and HEVC (H.265), are widely used for efficient video compression. Both standards support tiling, which divides a frame into independent regions, and use a quantization parameter (QP) to adjust compression levels within each tile. This helps in striking a balance between file size and visual quality, but lacks awareness of the regions of interest (RoIs) within a frame. Neural compression techniques [10], [11] are emerging to fill this gap, aided by hardware accelerators [12], [13]. However, applying these methods to our problem would demand extensive site-specific training. Moreover, the inference models are computationally heavy with high latencies, making GPU support essential [14], [15].

ROI Identification. Without explicit ROI detection, quantization is applied uniformly, missing opportunities for intelligent compression that could further optimize video streams by focusing on the most relevant areas of the frame. Most existing ROI-based compression approaches depend on analyzing the video frames directly. Methods such as motion estimation [16], [17], background subtraction [18], [19], and deep learning-based techniques [20], [21], [22] including attention based mechanisms [23], [24] are widely used for ROI identification. While these approaches are effective in optimizing compression, reducing communication bandwidth, they are neither fast nor inherently energy-efficient. First, running computer vision algorithms continuously is computationally intensive and often too heavy to execute locally on the camera. Second, it requires the camera sensors to remain powered at all times, leading to unnecessary resource consumption. Along with software based methods, there are specialized hardware that detect the RoIs intrinsically like event cameras. Unlike traditional cameras that capture full frames at fixed intervals, event cameras track only pixels where brightness changes surpass a threshold, indicating movement [25]. This method records only dynamic regions, offering two key advantages: reduced data volume and energy consumption, along with high frame rates with microsecond precision. However, their high cost limits large-scale deployment, especially in fields like surveillance.



(a) WiFi CSI, UWB CIR, mmWave Range-Azimuths **(b)** Comparing Algorithms **(c)** Comparing Images **(d)** Comparing Resolution & FPS

Fig. 2: Fig. 2a shows the video frame, mmWave range-azimuth map, WiFi CSI, and UWB CIR for the same scene. Fig. 2b, compares the energy increase from idle conditions and the time taken for various algorithms on a RaspberryPi. The algorithms are used to detect RoI, using mmWave (EcoVis), the video itself (background subtraction) and deep learning (YOLO). Fig. 2c shows YOLO's performance across different compression levels. Fig. 2d, shows the percentage increase in energy consumption with rising video resolution and frame rates compared to idle consumption on the RaspberryPi without a video sensor

RF-augmented Video Analytics and Surveillance. Much of the existing research integrating RF-sensing with video has focused on improving inference accuracy and detection tasks, with relatively little emphasis on addressing video compression techniques. For instance, studies like [26], [27], [28] have enhanced human activity recognition using RF and video data, while others like [29] have explored gesture recognition. Wireless sensing has also been used to verify the integrity of surveillance videos, checking for tampering with WiFi [30] and mmWaves [31]. Despite these advancements in improving inference accuracy, less attention has been given to how RF-sensing can contribute to video compression or be integrated with encoder standards like HEVC. Recently, integrating RF-based sensing with video surveillance has gained traction, particularly for traffic monitoring in challenging conditions such as low-light environments, non-line-of-sight scenarios, or adverse weather [32], [33], [9]. This approach offers a significant advantage in terms of reduced processing requirements compared to the computational demands of neural networks used in computer vision tasks. RF-augmented visual surveillance has opened up significant opportunities, with recent commercial products [7], [8] actively exploring and incorporating this technology into real-world applications.

B. Research Gap and Motivation

Choice of RF technology. While WiFi offers greater range and better penetration through the environment, it suffers from poor ranging resolution due to its moderately lower bandwidth. In contrast, mmWave provides excellent range resolution, making it a viable replacement for visual modalities in relatively simple tasks. Fig. 2a illustrates the RF signatures of the three modalities—WiFi, mmWave, and UWB—captured in our outdoor testbed. Notably, mmWave radars produce a cleaner and more intuitive representation of the scene compared to UWB and WiFi. While UWB and WiFi are suitable for simple tasks like vehicle or pedestrian counting, complex tasks (e.g., detecting vehicle type [32], RoI prediction) or integrating them with visual data poses challenges due to their lower bandwidths and range resolution.

Resource Footprint. Figs. 2b, 2c and 2d illustrate the impact of executing common computer vision primitives on an embedded device, comparable to typical IP camera hardware. Notably, even running relatively simple algorithms such as edge detection (EDGE [34]) or background subtraction (BDS [35]) (or relatively involved tasks like object identification using YoLoV8 [36]) result in significantly higher energy consumption and computational latency (fig. 2b) – compared to performing vehicle tracking using a mmWave radar. In figs. 2c, we specifically demonstrate how YoLo's inference performance based on RoI filtered frames is close to that of an UHD frame, with almost 50-60% lesser network footprint. Additionally, in fig. 2d, we demonstrate the relative increment in energy consumption while using different streaming resolutions and frame rates.

The integration of RF sensing into visual surveillance applications is still in its early stages, with significant research gaps remaining. Purely video-driven approaches require continuous camera operation and rely heavily on neural compute loads that are too demanding to run on embedded devices like surveillance cameras with acceptable frame rates. While WiFi or UWB-based solutions have shown potential for indoor scenarios, they lack the range and resolution required for robust outdoor surveillance. In contrast, mmWave technology offers a promising alternative. This paper addresses the challenge of developing energy-efficient and network-friendly video compression at the edge, leveraging mmWave sensing for high accuracy.

III. Design of The EcoVis System

We propose EcoVis, a lightweight system that intelligently integrates mmWave sensing with visual data, seamlessly interfacing with state-of-the-art H.265 video encoder. mmWave sensing is leveraged either for performing simpler inference tasks solely or for identifying potential Region(s) of Interest (RoIs) within the camera's field of view (FoV). EcoVis calculates a *tiling matrix* based on the positions of the RoIs, where each tile represents an independent partition of the frame. The coverage of RoIs within each tile is then mapped to a corresponding quantization parameter, allowing for independent compression across the tiles of the frame. The mmWave sensing driven RoI prediction is significantly more energy-efficient compared to even basic computer vision primitives (see figs. 2b and 8). To further optimize resource usage and support high frame rates, EcoVis employs, SORT, a

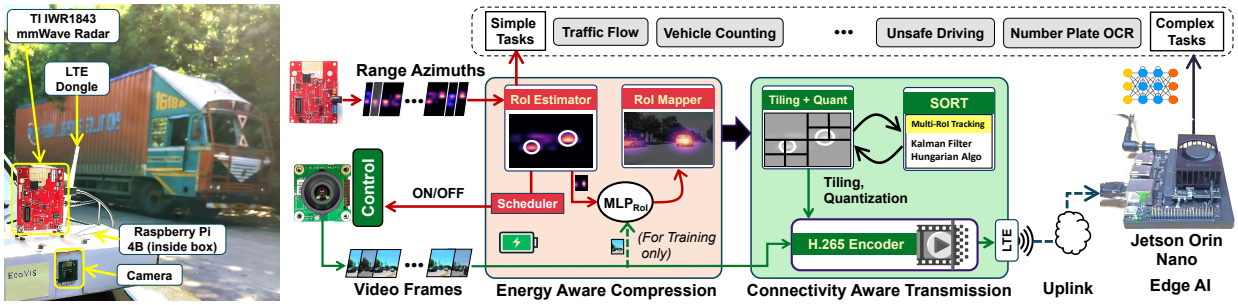


Fig. 3: Schematic diagram of the EcoVis system. The figure on the left presents a snapshot of our deployment prototype, featuring a Raspberry Pi 4B interfaced with an IWR1843BOOST mmWave radar and a 12 MP camera equipped with a SONY IMX500 sensor. Two important modules of the system are illustrated: (a) the RoI estimator and mapper, and, (b) the tiling and quantization module along with the H.265 encoder.We employ a Nvidia Jetson Orin Nano as the edge device to host the visual analytics processes, with the EcoVis prototype connected to it via LTE.

popular tracking algorithm, to track and predict multiple RoIs across frames. SORT uses a combination of the Kalman Filter combined with the Hungarian algorithm [37] (commonly used to solve *Assignment* problems) for the multi-RoI mapping. We present a detailed schematic of EcoVis in fig. 3 along with our prototype setup.

EcoVis consists of three functional components or phases. First, there is a *Bootstrapping and Calibration* phase, which initializes the various parameters associated with the internal models and algorithms to ensure optimal performance under specific conditions. The second component is a lightweight RoI *Estimator and Mapper* that identifies and maps potential RoIs within the frame, guiding the subsequent compression process. Finally, the *Communication-aware Video Encoding* phase adaptively tiles the frame and applies quantization to each tile, both spatially and temporally. The tiling matrix, along with the quantization information, is continuously passed to the H.265 encoder that generates the compressed video stream for network transmission.

A. Bootstrapping and Calibration

The mmWave radar captures range-azimuth maps ($\mathsf{RA_{map}}$) that register reflectors within its cross-sectional area. The RoI estimator then processes and enhances these spectrograms using signal processing algorithms, parameterized during the initial bootstrap and calibration phases (details discussed later). The RoI mapper utilizes a lightweight multilayer perceptron ($\mathsf{MLP_{RoI}}$) to map the RoI detected within the $\mathsf{RA_{map}}$ to the corresponding video frame. The $\mathsf{MLP_{RoI}}$ is deployed on the device and trained during the bootstrapping phase, where it learns weights and hyperparameters specific to the deployment site and the camera's 3D pose.

Dynamic parameter updates. Apart from the initial bootstrapping of the $\mathsf{MLP_{RoI}}$ model and setting calibration parameters, EcoVis autonomously updates them at specific time intervals, eliminating the need for human intervention. For example, a common issue with cameras is misalignment caused by strong winds or birds perching on them, which can severely impact calibration. While vision-based approaches often struggle to retrain after such disruptions, $\mathsf{MLP_{RoI}}$ is lightweight and can be efficiently retrained to recover from such issues.

B. Energy Aware RoI Estimator and Mapper

Predicting RoIs on the video frame from the range-azimuth ($\mathsf{RA_{map}}$) maps of mmWave radar involves two main steps: (a) *identifying* patches or RoI zones on the $\mathsf{RA_{map}}$ map that correspond to targets of interest and, (b) *mapping* such zones from $\mathsf{RA_{map}}$ to the video frame or the camera's FoV.

■ **Identifying Targets in Range-Azimuth Map.** The first step in target detection involves removing static clutter and random noise. EcoVis maintains a sliding time window buffer with N successive instances of $\mathsf{RA_{map}}$. Each such instance is filtered by removing $\overline{\mathsf{RA_{map}}}$ (the average map) that isolates the dynamic foreground components. Next, a Gaussian filter is applied to smoothen the foreground regions, which enhances target detection by attenuating low-intensity values and refining intensity edges. This further enhances the SNR of the resulting $\mathsf{RA_{map}}$. A critical parameter in this process is the variance of the Gaussian filter's kernel, which controls the level of smoothing applied to the $\mathsf{RA_{map}}$. The variance is influenced by the mmWave radar's configuration, specifically the maximum range and range resolution settings. Since it is independent of the scene, the filter variance can be predetermined and fixed during the calibration phase prior to deployment.

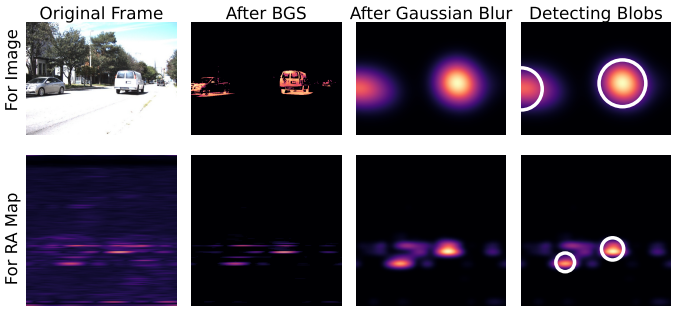


Fig. 4: The figure illustrates the steps involved in detecting targets of interest for both range-azimuth maps ($\mathsf{RA_{map}}$) and image frames, as discussed in §III-B

Next, to enumerate and register targets of interest, we detect independent blob-like structures in the $\mathsf{RA_{map}}$ (see fig. 4). While the *Laplacian of Gaussian* algorithm is effective for blob detection, its reliance on convolution operations introduces latency, particularly on an embedded device. Instead, we

employ the faster *Determinant of Hessian* (DoH), which uses box filters, providing sufficient accuracy for most blobs except very small ones. This trade-off is acceptable for our traffic surveillance use case. The DoH-based algorithm iteratively applies Gaussian kernels with different variances to compute the Hessian matrix, which is used to detect the blob centers and their radii. Identified blobs are then filtered according to overlap and intensity thresholds, producing a refined set of blobs, $\mathbf{B_{RA}}$. Each blob, $B^i_{RA} \in \mathbf{B_{RA}}$ is characterized by its center coordinates (X^i_{RA}, Y^i_{RA}) and radius R^i_{RA}.

■ **Mapping Targets from $\mathsf{RA_{map}}$ to Camera FoV.** As illustrated in fig. 5 the mapping between the range-azimuth coordinates of the $\mathsf{RA_{map}}$ and the width-height dimensions of the camera's FoV frame depends on both the FoV and the orientation/relative placement of the radar with respect to the camera. Additionally, the depth of the scene within the FoV may exhibit either azimuthal symmetry or asymmetry, which further influences this relationship. For asymmetric FoVs (fig. 5, top two rows), both azimuth-width and range-height exhibit positive correlations, though the rate of height increase diminishes with range due to perspective effects. In contrast, symmetric FoVs (fig. 5, bottom row) show minimal variation in range and height, leading to clustering around a central location. Although fig. 5 demonstrates a straightforward mapping of RoIs from the $\mathsf{RA_{map}}$ to the camera's FoV, factors like weather, sensor pose, and speckle noise introduce uncertainty. To address this, we use a simple multilayer perceptron, $\mathsf{MLP_{RoI}}$. The $\mathsf{MLP_{RoI}}$ maps each blob in the $\mathsf{RA_{map}}$, B^i_{RA}, to a corresponding blob on the camera frame, B^j_{IM}. The input layer has two neurons for the center coordinates of B^i_{RA} (i.e., (X^i_{RA}, Y^i_{RA})), while the output layer predicts the center coordinates and radius of B^j_{IM} with three neurons. The blob radius, R^i_{RA} is unused due to little or no correlation with the FoV image. We use two hidden layers with eight neurons each to form the network. $\mathsf{MLP_{RoI}}$ can be trained fairly accurately and occupies only a few tens of kilobytes of flash memory, making it lightweight and energy-efficient. Its real-time suitability is demonstrated by a median inference latency of approximately 100 ms on a Raspberry Pi 4B (including the RoI prediction), ensuring efficient operations in resource-constrained environments. Additionally, the model is periodically retrained, typically every few hours, and requires approximately 50 seconds on an Nvidia Jetson Orin Nano [38](edge device) to train $\mathsf{MLP_{RoI}}$ reliably – around 200 seconds without GPU support.

C. Communication Aware Video Encoding

After the RoIs (B^j_{IM}) are mapped to the camera's FoV, EcoVis partitions the video frame into tiles, allowing each tile to be compressed independently using a quantization parameter. To maintain high framerate processing with multiple RoIs, EcoVis employs SORT, a widely-used multi-target tracking algorithm, to predict intermediate RoI positions.

■ **Frame Tiling**. EcoVis can choose between *uniform tiling*, where all tiles in the video frame have the same dimensions, or *non-uniform tiling* that adapts to the distribution of RoIs

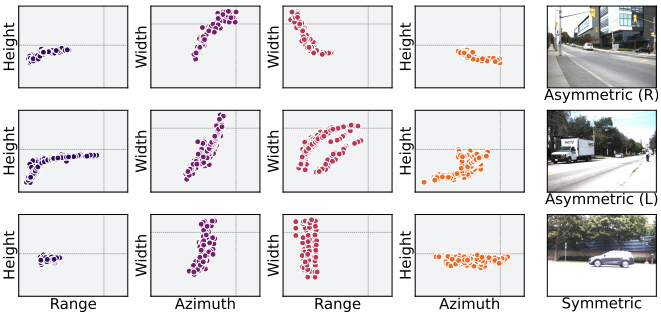


Fig. 5: This figure illustrates the relationship between the range-azimuth (RA) map and the image dimensions (height and width) for different scenes. Each point represents the coordinates of the centers of corresponding blobs. The range and height values are provided by y-coordinates of the centers of blobs for RA map and image respectively. Similarly, azimuth and width values are provided by x-coordinates

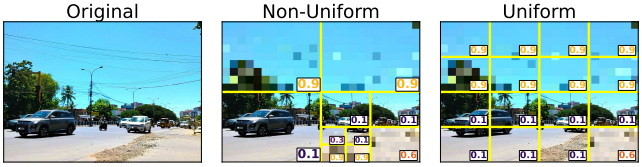


Fig. 6: Comparison of uniform versus non-uniform tiling. Note, how non-uniform tiling provides a more accurate level of compression, but with lesser number of tiles. The values mentioned inside each tile is the QP values normalized between 0 and 1. Note that lower QP is higher fidelity and vice versa

across the frame. For uniform tiling, the grid size is determined by the radius of the smallest blob, with a tile dimension of $\approx 2r$, where r is the radius of the smallest blob. Non-uniform tiling aims to partition the frame such that each tile maximally covers the points within a particular RoI blob. To achieve this, representative points are first generated for each blob by performing Gaussian sampling around the blob's center with variance proportional to the radius of the blob and their intensity values normalized to the range [0,1]. At this point, we define a metric ρ_T (aka, RoI coverage), which is calculated as the ratio of sum of intensity values of the representative points within a given tile T, to its area (hence, $\rho_T \in [0, 1]$). Non-uniform tiling uses a quad-tree based approach [39] to recursively partition the frame offering greater flexibility (fig. 6) compared to its uniform counterpart. The frame is initially divided into four tiles, and RoI coverage is calculated for each. For a tile T, if the coverage ρ_T greater than ρ_{high} or less than ρ_{low}, T is not subdivided any further. ρ_{low} and ρ_{high} are application specific parameters (discussed in the next paragraph). Tiles corresponding to $\rho_{\text{low}} \leq \rho_T \leq \rho_{\text{high}}$ are recursively subdivided until either a maximum number of iterations is reached or all tiles meet the coverage thresholds (typically 2–4 iterations, see Algo. 1 for further details).

■ **Assigning Quantization Parameter**. For each tile T, its coverage, ρ_T is mapped to a quantization parameter (QP), where higher values of ρ_T correspond to lower QP values, resulting in better fidelity. Conversely, lower values of ρ_T lead to higher QP values, facilitating greater compression at the expense of fidelity. While most real-time surveillance video streams target machine perception, reduced fidelity can in-

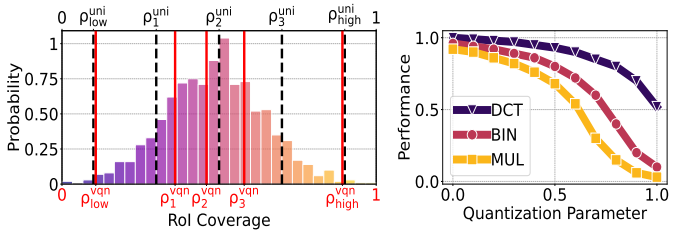


Fig. 7: Left: Distribution of the coverage of RoI in tiles. Please note how vector quantization differentiates the level non-uniformly based on the distribution. **Right**: Shows the impact of quantization on the performance for different application (discussed in §IV-C). The more complex the task higher is the impact of quantization

crease the risk of inference errors. We define four application-specific parameters: ρ_{low} and ρ_{high}, representing the upper and lower bound on RoI coverage respectively, and Q_{low} and Q_{high} that define the minimum and maximum QP values. As illustrated in fig. 7 *(right)*, first, lowering QP below Q_{low} results in diminishing returns, whereas arbitrarily high values of Q_{high} will degrade the performance of the inference model. Second, the choices for Q_{low} and Q_{high} are dependent on specific application requirements. Additionally, we introduce Q_{num}, representing the number of unique QP values, which is constrained by both the hardware capabilities and the real-time processing requirements. As shown in fig. 13 *(left)*, increasing the number of unique QP values can lead to higher processing demands. Tiles with a coverage ratio above ρ_{high} are assigned Q_{low}, while those below ρ_{low} receive Q_{high}. In cases where $\text{Q}_{\text{num}} \leq 3$, the QP value for tiles with coverage between ρ_{low} and ρ_{high} is set to the average value, i.e., $(\text{Q}_{\text{low}} + \text{Q}_{\text{high}})/2$. For scenarios where $\text{Q}_{\text{num}} > 3$ and $\rho_{\text{low}} \leq \rho_T \leq \rho_{\text{high}}$, the QP values are determined using *vector quantization* [40] based on the distribution of the tile coverage $\rho_T \forall T$. As shown in Fig.7 *(right)*, vector quantization allows for a non-uniform partitioning of the distribution (denoted as ρ_i^{vqn}, *solid red lines*), adapting to the frequency of the data rather than relying on uniform divisions (denoted as ρ_i^{uni}, *dashed black lines*).

■ **Multi-RoI Tracking and RoI Prediction**. The bounding box, once generated for two consecutive frames, is tracked in future frames. Such tracking relies on a *Kalman filter* to predict the position, scale, velocity and trajectory of the RoI targets across frames, smoothing out noisy detections. This allows for efficient prediction of RoIs between frames at a granular timescale without continuous re-detection. For association among the detected objects and predicted positions, the *Hungarian algorithm* is used. The algorithm minimizes the cost metric defined by the *Intersection over Union* (IoU) of bounding boxes, ensuring robust matching of detections to object tracks, especially when objects overlap or move between frames. The frequency of RoI generation can be user defined and is set during calibration. This is modulated by factors such as activity levels in the scene, environmental conditions and hardware constraints on the device. The resulting energy savings and their impact on the accuracy of RoI detection are analyzed in detail in §V-D.

Algorithm 1 QUAD-TREE based Non-Uniform Tiling

```
Tiling (Frame, B_IM):
    Tiles, Q ← {}
    Mask ← GaussianSamp(Frame, B_IM)
    NonUniTile(Mask, Tiles, Q, 0)
    return Tiles, Q

NonUniTile (Mask, Tiles, Q, i):
    ρ ← Sum(Mask)/Area(Mask)
    if ρ > ρ_high then:
        Q.append(Q_low)
        Tiles.append(Mask)
    else if ρ < ρ_min then:
        Q.append(Q_high)
        Tiles.append(Mask)
    else if i == MaxIter then:
        Q.append(AssignQP(ρ))
        Tiles.append(Mask)
    else
        h, w ← Height(Mask), Width(Mask)
        NonUniTile(Mask[0 : h/2, 0 : w/2], Tiles, Q, i + 1)
        NonUniTile(Mask[0 : h/2, w/2 : w], Tiles, Q, i + 1)
        NonUniTile(Mask[h/2 : h, 0 : w/2], Tiles, Q, i + 1)
        NonUniTile(Mask[h/2 : h, w/2 : w], Tiles, Q, i + 1)
    end if
```

IV. Testbed, Dataset and Baselines

We prototype the EcoVis system and deploy it end-to-end under real road traffic conditions on a busy street in Chennai, India. Additionally, to validate our results using established datasets, we leverage the RADDet dataset [32], which contains traffic surveillance videos paired with mmWave traces. We also evaluate a suite of benchmark applications on both our system and the RADDet dataset to further demonstrate the robustness of our approach.

A. EcoVis *Prototype and Testbed Setup*

The testbed employs a portable setup consisting of a Raspberry Pi 4B (RPi [41]) interfaced with a 12 MP SONY IMX500 sensor camera [42] and a IWR1843BOOST mmWave radar [43] from Texas Instruments. The setup is mounted on a car parked at the roadside, providing a clear view of oncoming traffic. Video frames are captured in UHD at 30 fps, while the radar's range-azimuth data is recorded at 5–8 fps.

Video Encoder. To integrate EcoVis's algorithms, we utilize Kvazaar [44], an open-source software-based implementation of the H.265 video encoder that runs on the RPi. While slower than hardware-based solutions, it is sufficient for validating our algorithms. Although hardware accelerators for H.265 are widely available, they currently lack the flexibility for runtime optimizations (as provided by Kvazaar), which EcoVis requires. Specifically, the tiling and quantization module (see fig. 3) continuously interacts with Kvazaar to dynamically update the RoI. Kvazaar uses the QP values between 0 and 51, where 0 represents no quantization and 51 is the maximum quantization. A typical value of 27 is used for QP, that balances compression and size. The QP values used by EcoVis

are normalized between zero and one, and does not depend on any specific software implementation.

Power Usage. The radar sensor consumes ≈1.4W when all antennas (4 RX and 3 TX) are active. With EcoVis's software components running, the Raspberry Pi's power consumption ranges between 3.5W and 4W. The camera adds an additional 3W for capturing UHD video at 30 fps, as shown in fig. 2d. The RPI is connected via cellular data, streaming the compressed video to an edge device – an NVIDIA Jetson Orin Nano [38], featuring a 1024-core GPU with 32 tensor cores.

B. Benchmark Datasets for Evaluation

We utilize two distinct datasets in this study – a publicly available dataset, RADDet [32], and a custom dataset that we collect using on our EcoVis testbed.

■ **RADDet Dataset.** The RADDet dataset consists of 10K+ standard definition (SD, 640×480) frames of road traffic imagery and their corresponding RAD (3D Range-Azimuth-Doppler) map. From the RAD map we only consider the range and azimuth data to get a 256×256 range-azimuth map or $\mathsf{RA}_{\mathsf{map}}$. The dataset spans 14 different scenes at different roads, times of the day and road conditions.

■ **EcoVis Dataset.** Our dataset has around 50K+ UHD video frames and their corresponding $\mathsf{RA}_{\mathsf{maps}}$. Similar to RADDet the $\mathsf{RA}_{\mathsf{maps}}$ are of size 256×256. We also capture data in adverse condition like low light (night time), fog (mist) and rain. The normal, low light, fog and raining conditions belong to four different scenes totally spanning $\approx 40+$ minutes.

C. Video Analytics Applications on Edge Device

To effectively benchmark EcoVis's video compression, we use a range of video analytics applications with increasing complexity listed in the following. Tasks at higher levels demand more computational resources and higher-resolution images, particularly in regions of interest (ROIs). These applications are run on our Jetson Orin Nano Edge device.

■ **Detection (**DCT**) and Counting (**CNT**)**. The detection task distinguishes between *busy* (with moving cars or pedestrians) and *empty* frames, evaluated by accuracy, while the counting task determines the exact number of objects per frame, with performance measured by the error between predicted and actual counts.

■ **Binary (**BIN**) and Multi-class (**MUL**) Classification**. The binary classification task differentiates objects as vehicle or pedestrian, with performance measured by *mean average precision* (mAP), accounting for both detection accuracy and prediction confidence. The multi-class classification task expands such class categories to include cars, trucks, buses, motorcycles, bicycles, and pedestrians, focusing on the challenge of distinguishing components of the road traffic, with performance also evaluated using mAP.

■ **Automatic number plate recognition (**ANPR**)**. This task involves detecting and classifying vehicles while recognizing license plates, necessitating high fidelity in ROIs. Performance is assessed by the ratio of correctly detected number plates to the total number in the ground truth. We do not present results for performing OCR on the number plate since, once it is correctly identified as an ROI, the OCR software on the edge can be fine-tuned to optimize performance.

It is important to emphasize that the ground truth for the aforementioned algorithms is established by performing the analysis on non-compressed video data. In the subsequent results (§V), we demonstrate the degradation in performance when videos undergo compression. This approach highlights the relative impact of video compression on the accuracy and effectiveness of video analytics. A additional advantage is that, by using the non-compressed video as a baseline, we can effectively manage and compare performance metrics across different tasks that may not be directly comparable. Further, it highlights the excess compute overhead introduced by the compression process itself, providing a holistic view of its trade-offs in practical deployments.

D. SOTA *RoI Detection Techniques (Baselines)*

Contemporary methods for identifying ROI zones in surveillance video frames mainly rely on analyzing the video content itself to detect motion. Since surveillance cameras are often fixed, with a largely static background, regions with movement become the focus.

Many state-of-the-art (SOTA) techniques [45], [46], [47], [48], [22], known for their performance and energy efficiency, enhance or combine four key video processing primitives for ROI detection: **BGS:** background subtraction or frame differencing [45], [47], **ED:** edge detection [45], [46], [47], and, **SAD:** sum of absolute difference [46], [48]. In the following sections, we compare our ROI detection algorithm, based on mmWave $\mathsf{RA}_{\mathsf{maps}}$, with these four key primitives.

Although deep learning methods for ROI detection have made significant strides [49], [50], [51], [52], they are not fast enough to achieve real-time inference speeds and are often too demanding in terms of memory and compute requirements (e.g., GPU support) for deployment on modest edge devices. Additionally, running such models directly on a local camera is generally impractical due to their resource intensity.

V. EVALUATION RESULTS

We evaluate EcoVis in terms of energy, network bandwidth and the end application performance. We provide provide detailed empirical results and evidences that motivate algorithmic decisions taken in our approach.

A. Compressing video using mmWave Range-Azimuth Map

■ **Implications on Energy and Network Bandwidth.** The primary objective of EcoVis is to minimize energy consumption required for video compression and the encoding processes, while simultaneously maximizing the reduction in network bandwidth usage. As demonstrated empirically in fig. 8, EcoVis achieves comparable reductions in network bandwidth usage but with substantially lower energy consumption when compared to existing SOTA methods: **SAD**, **ED** and **BGS**. To ensure a fair comparison across the different baselines we use the same values of $\mathrm{Q}_{\mathrm{high}}$, $\mathrm{Q}_{\mathrm{low}}$ and

Q_{num}, while the tiling depends on the RoI detected by the different methods. EcoVis improves energy consumption by 30% – 40%. This reduction in energy consumption extends the operational time, an encouraging step towards battery-powered camera operations. For instance, while running on a RPi powered by a 10,000 mAh battery, this can extend battery life by almost two hours (a 20% increase). Furthermore, we perform a comparative analysis between our testbed dataset and RADDet, highlighting that the decrease in energy consumption is more pronounced in our case. This is primarily due to the use of higher resolution images (UHD), compared to the SD resolution employed in RADDet. This also shows that higher the resolution of camera the more beneficial it is to use EcoVis, rather than existing methods.

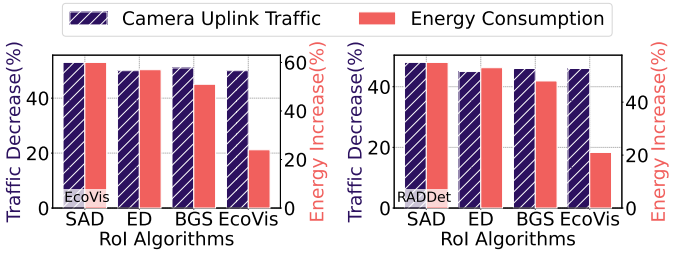


Fig. 8: The figure illustrates the percentage increase in energy consumption during video compression and encoding, compared to the percentage reduction in network bandwidth usage resulting from the compression. The energy increase is measured relative to the system's idle energy consumption, while the bandwidth reduction is calculated relative to the bandwidth used when transmitting uncompressed video. The left plot corresponds to the EcoVis dataset, and the right plot corresponds to the RADDet dataset.

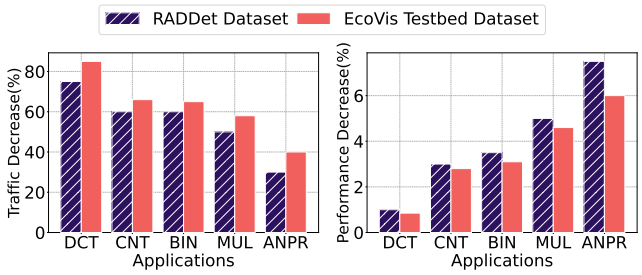


Fig. 9: The edge application performance and network bandwidth usage for non-compressed versus compressed video using EcoVis are showcased. Depending on the application, Q_{low} and Q_{high} values were varied during the encoding process to achieve different levels of compression.

■ **Implications on Edge Application Performance.** Fig. 9 highlights the effectiveness of ECOVIS in leveraging video compression to optimize network traffic while preserving analytics accuracy. For simpler tasks, such as vehicle or pedestrian detection, ECOVIS achieves a substantial reduction in network traffic – exceeding 75% – with negligible accuracy loss, typically under 1%. While the benefits of compression are less pronounced for more complex tasks, such as ANPR, ECOVIS still demonstrates its utility by maintaining accuracy degradation below 10% with a modest reduction in network traffic ranging between 30% and 40%. These results underscore the adaptability of ECOVIS to varying levels of task complexity. It is important to note that the primary role of mmWave radar in ECOVIS is to guide the compression process by identifying the RoI, while the analytics are performed using standard computer vision pipelines. The potential of using only mmWave data for analytics, independent of video, is discussed in §V-B.

Our dataset shows a slight performance improvement (≈ 1%), but a more significant reduction in bandwidth (≈ 5–10%) compared to RADDet. This is due to the higher image resolution in our dataset, where compression benefits are more pronounced with higher resolution inputs. Note that reduction in network bandwidth also improves energy consumption.

■ **Effect of Environmental or Weather Conditions.** Environmental factors like low light (e.g., nighttime) and weather conditions (rain, fog) impair visibility and degrade the performance of visual-based techniques such as **BGS**, **ED**, and **SAD**. While advanced image deblurring methods exist, they require extensive scenario-specific training and are often too latency-intensive for real-time use [53]. In contrast, mmWave sensing is much less impacted by weather conditions. As demonstrated in fig. 10, EcoVis proves advantageous for detecting RoI in environments where video quality is compromised. While all techniques experience similar performance degradation under normal conditions, EcoVis significantly outperforms others in adverse conditions, particularly in low-light scenarios (by over 5%). For rainfall and fog, there is a slight degradation in mmWave range (both maximum distance and accuracy) and azimuth accuracy due to scattering [9], but such differences remain marginal (2% to 5%). Notably, most of the performance decline in EcoVis under adverse conditions arises from challenges in video analytics (e.g., multi-class detection and classification) on degraded images, rather than inaccuracies in RoI detection.

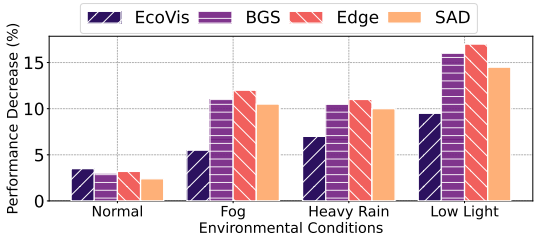


Fig. 10: This figure compares the accuracy of multi-class classification when videos are compressed using various techniques, across different environmental conditions. The percentage decrease in accuracy is measured relative to the non-compressed videos, highlighting how different compression methods impact performance under varying conditions

B. EcoVis-*sans-Video for Simple Applications.*

For simpler tasks like vehicle and pedestrian detection, counting, and binary classification, mmWave range-azimuth maps alone suffice. This eliminates the need to compress video and transmit it to the edge for analysis. This approach offers substantial savings in network bandwidth and reduces inference latency (by an order of magnitude) by avoiding network overhead. Although it may slightly increase energy consumption, the amount of energy used is comparable or less (e.g., using DL accelerators or TPUs [54]) to that required for video compression and encoding. Fig.11 illustrates the trade-off between performance and inference latency for three

simple tasks. While the decrease in accuracy compared to compressed video is modest (less than 5%), the reduction in inference latency is more pronounced, with improvements of approximately 25%. For tasks like detection only the range information (without azimuth data) is enough and gives a similar performance with even more significant decrease in inference latency (by about 60%).

C. Duty-Cycling Camera using Range-Azimuth Maps

A key advantage of EcoVis is to identify RoIs and the ability to turn off camera sensors in absence of targets, leading to substantial energy savings. As shown in fig. 12, the energy savings are more pronounced for higher resolution cameras that require more power to operate. Under low traffic conditions — when there are extended periods with no significant activity, the camera remains off for longer durations, further increasing energy efficiency. For battery-powered systems, this results in extended operational time. When combined with the energy savings discussed in §V-A, we observe an approximate 4 to 5 hour increase in battery life (≈ 50% increment) under average traffic conditions.

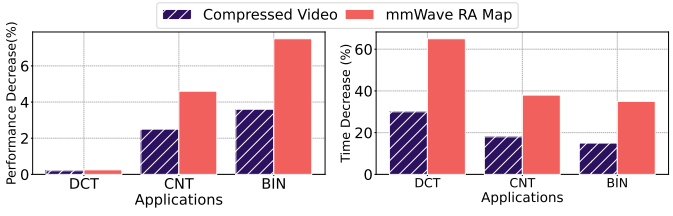


Fig. 11: EcoVis-sans-Video. Effectiveness of using *only* range-azimuth maps compared to compressed video for simple tasks. **Left:** reduction in accuracy, **Right:** reduction in inference latency when compared to non-compressed videos.

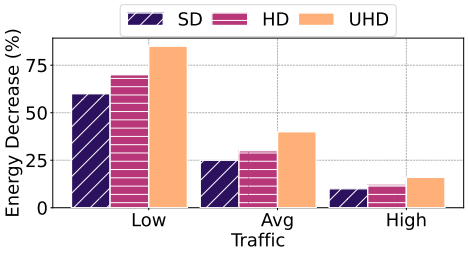


Fig. 12: Aggregate reduction in energy consumption achieved by turning off the camera using EcoVis across three traffic scenarios: light, medium, and heavy. The results are shown for different video resolutions — 480p, 1080p, and 4K. The percentage decrease in energy consumption is measured relative to the baseline where the camera remains continuously active.

D. Discussion on Parameter Choices Made for EcoVis.

In this section, we provide the rationale behind several key design choices for EcoVis, as discussed in §III.

■ **Tile Count and Unique QP Values.** As shown in fig. 13 (*left*), both increasing the number of tiles and the number of unique Quantization Parameter (QP) values lead to higher energy consumption during encoding. This is due to the increased computational complexity of the encoding process. Although the figure presents results for uniform tiling, similar trends are observed with non-uniform tiling. These findings highlights the importance of limiting both the tile count and the number of unique QP values used in the system.

■ **Uniform *vs.* Non-Uniform Tiling.** Although uniform tiling typically results in a greater number of tiles compared to non-uniform tiling, it is simpler to implement. Consequently, despite the expected increase in energy consumption due to the higher tile count, the simplicity of uniform tiling often results in comparable or, in some cases, slightly lower energy consumption than non-uniform tiling.

■ **SORT for Interpolating RoIs.** We utilize SORT to predict RoIs for the next frame based on the previous frame, rather than generating the RoI with EcoVis for every frame, reducing computational load and energy consumption. The critical question is how frequently the RoI should be estimated from the sensing data — specifically, at what time intervals should SORT be applied. As illustrated in fig. 13 (*right*), the energy savings exhibit diminishing returns, although the initial reduction is substantial. Meanwhile, compression accuracy, measured by the *Intersection over Union* (IoU) between SORT predicted and actual RoIs, decreases exponentially as the interval increases, becoming significant at larger intervals. Based on empirical evidence, we recommend an interval of 1–2 seconds between each RoI prediction.

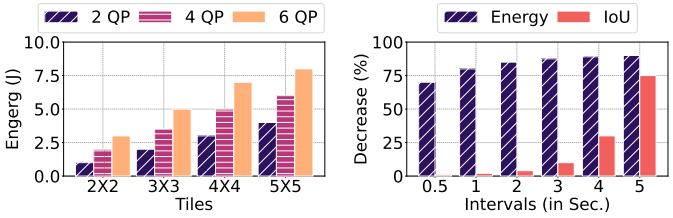


Fig. 13: Left: Increase in energy consumption as the number of tiles and distinct Quantization Parameter (QP) values increase during encoding. **Right**: Percentage reduction in energy consumption versus percentage decrease in accuracy (measured by IoU) when using SORT, evaluated across different time intervals for the next detection. The percentage decrease is relative to the case where SORT is not applied.

VI. Conclusions

In this paper, we propose a lightweight hybrid surveillance system EcoVis that integrates mmWave-based sensing with conventional camera setups, significantly reducing computational overhead compared to traditional vision-only approaches. By utilizing mmWave range-azimuth maps to identify RoIs in camera frames, our system optimizes video compression, leading to a reduction in energy consumption by approximately 50% and a 60% decrease in network bandwidth usage, without compromising analytics accuracy. We validated our system with a real-world deployment in an Indian metro city and demonstrated its robustness across various traffic scenarios, showcasing its suitability for efficient, sustainable surveillance in areas with limited infrastructure or power. We also showcase the resilience of EcoVis to adverse environment conditions. Our approach further leverages mmWave's out-of-band sensing capability to control camera operation, turning it off when there is no object around, further enhancing both energy efficiency and network usage.

REFERENCES

[1] M. Aasif, A. Beohar, D. Kumar *et al.*, "Status of policing in india report 2023: Surveillance and the question of privacy," *Common Cause*, 2023. [Online]. Available: https://www.commoncause.in/wotadmin/upload/REPORT_2023.pdf

[2] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on circuits and systems for video technology*, 2012.

[3] Reolink, *IP Camera Bandwidth Calculation: Easy Formula and Quick Tips to Reduce Bandwidth Usage*. [Online]. Available: https://reolink.com/blog/ip-camera-bandwidth-calculation/

[4] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[5] N. Carion, F. Massa, G. Synnaeve *et al.*, "End-to-end object detection with transformers," in *ECCV 2020*. Springer.

[6] B. Mildenhall, P. P. Srinivasan, M. Tancik *et al.*, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, 2021.

[7] D. Engineering, *D3-3P-DESIGNCORE-RADAR*. [Online]. Available: https://www.ti.com/tool/D3-3P-DESIGNCORE-RADAR

[8] Milesight, *X5-Sensing Camera*. [Online]. Available: https://www.milesight.com/product/x-infinity/x5-sensing-camera

[9] S. Zang, M. Ding, D. Smith *et al.*, "The impact of adverse weather conditions on autonomous vehicles: How rain, snow, fog, and hail affect the performance of a self-driving car," *IEEE Vehicular Technology Magazine*, 2019.

[10] L. Theis, W. Shi, A. Cunningham, and F. Huszár, "Lossy image compression with compressive autoencoders," in *ICLR 2017*.

[11] K. Gregor, F. Besse, D. Jimenez Rezende *et al.*, "Towards conceptual compression," in *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2016.

[12] S. Ma, X. Zhang, C. Jia *et al.*, "Image and video compression with neural networks: A review," *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

[13] Ambrella, *Ambrella H22*. [Online]. Available: https://www.ambarella.com/wp-content/uploads/Ambarella_H22_Product_Brief.pdf

[14] NVIDIA, *NVIDIA Video Codec SDK*. [Online]. Available: https://developer.nvidia.com/video-codec-sdk

[15] ——, *NVIDIA TensorRT*. [Online]. Available: https://developer.nvidia.com/tensorrt

[16] X. Guo, S. Li, and X. Cao, "Motion matters: A novel framework for compressing surveillance videos," in *ACM MM 2013*.

[17] A. D. Bagdanov, M. Bertini, A. Del Bimbo, and L. Seidenari, "Adaptive video compression for video surveillance applications," in *IEEE ISM 2011*.

[18] F. Yonga, C. Bobda, and A. Zarazadeh, "Improving video communication in distributed smart camera systems through roi-based video analysis and compression," in *IEEE ICDSC 2012*.

[19] H. Xue, Y. Zhang, and Y. Wei, "Fast roi-based hevc coding for surveillance videos," in *19th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, 2016.

[20] L. Zhao, S. Wang, S. Wang *et al.*, "Enhanced surveillance video compression with dual reference frames generation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

[21] R. R. Selvaraju, M. Cogswell, A. Das *et al.*, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *IEEE ICCV 2017*.

[22] J.-Y. Wu, V. Subasharan, T. Tran, and A. Misra, "Mrim: Enabling mixed-resolution imaging for low-power pervasive vision tasks," in *IEEE PerCom 2022*.

[23] A. de Santana Correia and E. L. Colombini, "Attention, please! a survey of neural attention models in deep learning," *Artificial Intelligence Review*, 2022.

[24] K. Xu, J. Ba, R. Kiros *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *ICML 2015*.

[25] G. Gallego, T. Delbrück, G. Orchard, and Bothers, "Event-based vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[26] J. Chen, K. Yang, X. Zheng *et al.*, "Wimix: A lightweight multimodal human activity recognition system based on wifi and vision," in *IEEE MASS 2023*.

[27] C. Zhu, Z. Zhao, Z. Shan *et al.*, "Robust target detection of intelligent integrated optical camera and mmwave radar system," *Digital Signal Processing*, 2023.

[28] H. Chen, S. Munir, and S. Lin, "Rfcam: Uncertainty-aware fusion of camera and wi-fi for real-time human identification with mobile devices," *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2022.

[29] X. Liu, S. Tang, B. Zhang *et al.*, "Wivi-gr: Wireless-visual joint representation based accurate gesture recognition," *IEEE Internet of Things Journal*, 2023.

[30] Y. Huang, X. Li, W. Wang *et al.*, "Forgery attack detection in surveillance video streams using wi-fi channel state information," *IEEE Transactions on Wireless Communications*, 2021.

[31] M. Han, H. Yang, M. Jia *et al.*, "Seeing the invisible: Recovering surveillance video with cots mmwave radar," *IEEE Transactions on Mobile Computing*, 2024.

[32] A. Zhang, F. E. Nowruzi, and R. Laganiere, "Raddet: Range-azimuth-doppler based radar object detection for dynamic road users," in *Conference on Robots and Vision 2021*.

[33] S. Yao, R. Guan, X. Huang *et al.*, "Radar-camera fusion for object detection and semantic segmentation in autonomous driving: A comprehensive review," *IEEE Transactions on Intelligent Vehicles*, 2023.

[34] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, 1986.

[35] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *IEEE ICPR 2004*.

[36] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE CVPR 2016*.

[37] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[38] NVIDIA, *NVIDIA Jetson Orin Nano Developer Kit*. [Online]. Available: https://developer.nvidia.com/embedded/learn/get-started-jetson-orin-nano-devkit

[39] D. P. Mehta and S. Sahni, *Handbook of data structures and applications*. Chapman and Hall/CRC, 2004.

[40] P. C. Cosman, K. L. Oehler, E. A. Riskin, and R. M. Gray, "Using vector quantization for image processing," *Proceedings of the IEEE*, 1993.

[41] Raspberry Pi, *Raspberry Pi 4 Model B*. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/

[42] ——, *Raspberry Pi AI Camera with SONY IMX500 Sensor*. [Online]. Available: https://www.raspberrypi.com/products/ai-camera/

[43] Texas Instruments, *IWR1843BOOST: User's Guide*. [Online]. Available: https://www.ti.com/lit/ug/spruim4b/spruim4b.pdf

[44] M. Viitanen, A. Koivula, A. Lemmetti *et al.*, "Kvazaar: Open-source hevc/h.265 encoder," in *ACM MM 2016*.

[45] J. H. Ko, B. A. Mudassar, and S. Mukhopadhyay, "An energy-efficient wireless video sensor node for moving object surveillance," *IEEE Transactions on Multi-Scale Computing Systems*, 2015.

[46] A. Aliouat, N. Kouadria, M. Maimour, and S. Harize, "Region-of-interest based video coding strategy for low bitrate surveillance systems," in *IEEE SSD 2022*.

[47] A. Aliouat, N. Kouadria, M. Maimour *et al.*, "Region-of-interest based video coding strategy for rate/energy-constrained smart surveillance systems using wmsns," *Ad Hoc Networks*, 2023.

[48] J. H. Ko, T. Na, and S. Mukhopadhyay, "An energy-quality scalable wireless image sensor node for object-based video surveillance," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2018.

[49] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE CVPR 2014*.

[50] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[51] J. Redmon, "You only look once: Unified, real-time object detection," in *IEEE CVPR 2016*.

[52] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *IEEE/CVF CVPR 2023*.

[53] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "Deblurgan: Blind motion deblurring using conditional adversarial networks," in *IEEE CVPR 2018*.

[54] Coral, *USB TPU Accelerator*. [Online]. Available: https://coral.ai/static/files/Coral-USB-Accelerator-datasheet.pdf