IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 November 2024, accepted 23 December 2024, date of publication 8 January 2025, date of current version 14 January 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3527134

RESEARCH ARTICLE

Onboard Person Retrieval System With Model Compression: A Case Study on Nvidia Jetson Orin AGX

JAY N. CHAUDHARI[1], HIREN GALIYAWALA[2], PAAWAN SHARMA[3], (Senior Member, IEEE), PANCHAM SHUKLA[4], AND MEHUL S. RAVAL[1], (Senior Member, IEEE)

[1]School of Engineering and Applied Science, Ahmedabad University, Ahmedabad 380009, India
[2]Rydot Infotech Pvt. Ltd., Ahmedabad 380027, India
[3]School of Technology, Pandit Deendayal Energy University, Gandhinagar 382007, India
[4]Imperial College London, SW7 2AZ London, U.K.

Corresponding author: Mehul S. Raval (mehul.raval@ahduni.edu.in)

This work was supported in part by the Imperial Open Access Fund (Imperial Fund); and in part by the Science, Technology, and Innovation (STI) Policy of Gujarat Council of Science and Technology, Department of Science and Technology, Government of the Gujarat State, India, under Grant GUJCOST/STI/2021-22/3858.

ABSTRACT A person retrieval system (PRS) in video surveillance identifies an individual based on descriptive attributes, a task that employs several computationally intensive deep learning models. We implement and analyse a PRS for pre-recorded videos on a graphics processing unit (GPU) and Nvidia Jetson Orin AGX. This paper presents a new Person Attribute Recognition (PAR) architecture, CorPAR, using three backbone networks, ConvNext, ResNet-50, and EfficientNet-B0. It enhances the F1-score by 4.1% with ConvNeXT-Base, 1.63% with the ResNet, and by 8.07% with EfficientNet-B0, surpassing the performance of the state-of-the-art Weighted-PAR method. The proposed method uses model compression techniques like quantisation and pruning with L1 regularisation to assess their impact on person retrieval. The study reveals that the PRS utilising EfficientNet-B0, with 32-bit quantisation, achieves the best performance, delivering a throughput of 22 frames per second and a True Positive Rate of 71% on Nvidia Jetson Orin AGX matching the performance of a model implemented using GPU.

INDEX TERMS Edge device, model compression, person attribute recognition, person retrieval, pruning, quantization, surveillance.

I. INTRODUCTION

Traditional video surveillance is crucial in maintaining public order and monitoring criminal activities. However, its effectiveness is compromised by labour-intensive and error-prone manual review processes. These methods are difficult to scale with the growing network of cameras and video databases and often do not meet real-time requirements. Artificial Intelligence (AI) based surveillance solutions are emerging with the potential to overcome these limitations. This development is driven by training deep learning models on Graphics Processing Units (GPU) with large amounts of data, enabling real-time insights and automating security and surveillance methods.

This paper addresses the challenge of person retrieval, particularly its deployment on edge devices. Person retrieval involves identifying an individual in surveillance footage based on discrete attributes extracted from a natural language description. This description typically consists of attributes such as gender, upper body clothing type and colour, and lower body clothing type, each of which may have multiple values. For example, a description like "A person with a white shirt and black pants" will have discrete attributes as *upper body clothing:* ***shirt****, upper body clothing colour:* ***white****, lower body clothing type:* ***pant****, lower body clothing colour:* ***black****.*

The associate editor coordinating the review of this manuscript and approving it for publication was Yongjie Li.

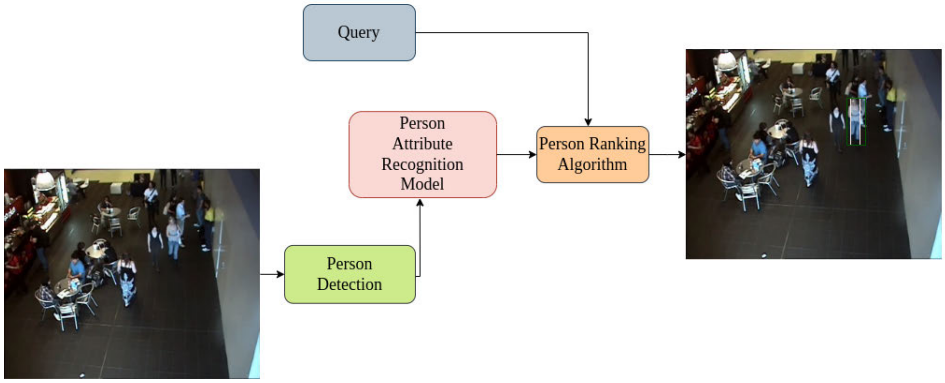


FIGURE 1. Overview of the person retrieval system. The input is a query with discrete attributes and a video frame. The output in a frame is a person(s) matching the query.

The Person Retrieval System (PRS), illustrated in Figure 1, integrates several vital components, including query processing, detection, person attribute recognition (PAR), and a person ranking algorithm (PRA). The input to the system is a video and a query specifying discrete attributes of the target individual. A detection module first detects all persons within the frames, after which the PAR model extracts specified attributes for each detected individual. The PRA then ranks the detected individuals by comparing their predicted attributes against the query attributes to retrieve the closest match.

The proposed system is primarily evaluated on two hardware platforms: the Nvidia Jetson Orin AGX, an edge device, and the NVIDIA RTX A5000 GPU. It is designed as a local inference system to reduce latency, enhance data privacy, and enable real-time decision-making. This makes the system well-suited for deployment in edge environments with limited or unreliable network connectivity. This study represents the first to investigate the deployment of PRS on edge devices.

A. MODEL COMPRESSION

Deep learning models trained on GPUs require massive datasets to improve their generalisation and consume significant energy despite their impressive abilities [1]. These disadvantages limit the accessibility and scalability of AI-powered surveillance on edge devices. However, Model Compression (MC) techniques like pruning, quantisation, knowledge distillation, and low-rank factorisation can overcome these challenges [2]. These techniques enable the execution of the models on edge devices [2].

Quantisation compresses the original network by reducing the number of bits required to represent each weight. The process involves converting a neural network model's weights and activation values from high precision to low precision. It lowers memory overhead, bandwidth requirements, and power consumption and provides faster computation speed. Typically, parameter quantisation is carried out by either absolute max quantisation or zero point quantisation [1], [3].

Pruning is a technique used in deep learning to reduce the size of a neural network by removing redundant or less essential weights, thereby improving computational efficiency without significantly sacrificing performance. Structured pruning removes entire units, such as neurons, channels, or filters, resulting in a more organised and structured reduction of the network. This method is more accessible to implement on standard hardware, as it preserves the overall architecture of the model, making it more efficient for practical deployment [1]. Unstructured pruning eliminates individual redundant or non-contributory weights, streamlining the model and potentially increasing sparsity. L1 regularisation encourages sparsity in the model's parameters, further reducing model complexity and improving execution speed [1].

B. LITERATURE REVIEW

The PRS employs convolutional neural networks (CNNs) to extract features and create embeddings. Attributes are detected step by step using linear filtering [4], but this approach often suffers from inefficiency due to noise build-up. Improvements like adaptive torso patch extraction and bounding box regression enhance the detection and retrieval process [5]. Bekele et al. [6] proposed a ResNet backbone for PAR, showing that deeper networks improve classification accuracy and generalisation. In a study, Galiyawala et al. [7] introduced a single network model combining PAR with Mask R-CNN for person detection. This model achieved an average True Positive Rate (TPR) of 85.30% by optimising attribute weights.

In unified person attribute recognition (UPAR), the baseline method allows for developing large-scale, generalisable attribute-based person retrieval [8]. Approaches mentioned by Specker et al. [9] simultaneously improve single-attribute-based recognition and retrieval using spatial projection and normalisation modules. A novel recurrent neural network with a gated neural attention mechanism established a baseline performance over CUHK-PEDES, a large-scale person re-identification dataset containing

Natural Language Descriptions (NLDs) with images [10]. However, segmentation-based methods are computationally expensive and sensitive to variations.

The techniques described above primarily focus on GPU training models but often lack discussions on their deployment on edge devices with low resources. Understanding these methods' practical utility is essential to evaluate their performance under edge devices' constraints.

Parate et al. [11] emphasises optimal real-time performance for CPU-only edge devices in residential video surveillance, utilising a Long Range Radio network for privacy-preserving anomaly alerts. Gaikwad et al. [4] conducted person re-identification using the CEERI-VREID dataset, achieving real-time performance of 30 frames per second (FPS) on a distributed setup of NVIDIA Jetson Nano and TX2 with a cloud server. Suzen et al. [12] assess the performance of single-board computers, specifically NVIDIA Jetson Nano, Jetson TX2, and Raspberry Pi4, in executing a CNN algorithm for fashion product categorisation. Ullah et al. [13] conducted a comprehensive performance benchmarking of Jetson platforms, exploring the impact of algorithm optimisation and hardware acceleration across various data types, including dense, deep learning architectures and hyperspectral images.

These studies collectively contribute to understanding real-time intelligent surveillance and the efficiency of edge devices in diverse applications. However, existing edge device benchmarking studies predominantly focus on single modules, such as detection. More so, multi-model modules are not deployed fully on a single edge device, as they are partly implemented on a cloud-edge distributed network [4], [13], [14]. It is worth examining the deployment of a PRS with multiple modules wholly on edge device. The contributions are summarised as follows:

1) We introduce a new CorPAR architecture for attribute recognition.
2) We investigate and analyse the deployment of complete PRS on NVIDIA Jetson Orin AGX edge device.

The remainder of the paper is outlined as follows: Section II elaborates on the proposed system for person retrieval. Section III provides a detailed discussion of the results and findings, with the conclusion given in Section IV.

II. PROPOSED PERSON RETRIEVAL SYSTEM

The PRS involves three stages: person detection, PAR and person ranking, which are briefly discussed in each module below.

A. PERSON DETECTION MODULE

The ideal characteristics for the detection module should include robustness to occlusion, varying poses, illumination conditions, and portability for edge devices. Chaudhari et al. [15] showed that the detector You Look Only Once (YOLO) v8 performs better than YOLOv7 on the GPU. Hence, we picked the readily available YOLOv8n model [16] pre-trained on the Microsoft COCO dataset for preliminary investigations. It resulted in the mean average precision (mAP) of 37.3. Further on qualitative analysis, we observed that YOLOv8n failed to detect all the persons in the low lights or if the person was occluded by more than 50%. To improvise the detection, we selected a different model from the YOLOv8 suite and fine-tuned it on a benchmark person detection dataset. Considering the small size, we selected YOLOv8s.

We used the CrowdHuman dataset [17] for evaluating person detection in a crowd. It is a large, richly annotated dataset with 15,000, 4,370, and 5,000 images for training, validation, and testing. The dataset covers various scenarios with 470,000 human instances, averaging 22.6 persons per image, and includes multiple occlusions. Scenes range from streets, parks, and markets to stadiums, with individuals often partially obstructed by others, objects, or the background. YOLOv8s is fine-tuned for 100 epochs on the CrowdHuman dataset, with a split 90:10 between training and testing. Performance is evaluated using mAP, considering precision and recall across different confidence thresholds. When tested, it resulted in a better map of 45.6. On the qualitative comparison between YOLOv8n and YOLOv8s, we observed that YOLOv8s provided better detection with a higher confidence score for the same subject. With only 11.1 million parameters [15] on the CrowdHuman dataset [17], the YOLOv8s model has evolved to be robust and efficient. Therefore, YOLOv8s is employed in the proposed PRS.

B. PERSON ATTRIBUTE RECOGNITION MODULE

Weighted-PAR (W-PAR) [18] is inspired by the DeepMAR architecture [19]. However, the W-PAR model possesses the following limitations:

1) *Attribute correlation:* Attributes are often dependent; for example, certain types of clothing are more likely to be worn together, and models need to understand and leverage these correlations without becoming overly reliant on them.
2) *Inter-attribute interference:* The recognition of one attribute can interfere with the presence or absence of another. For instance, the presence of glasses might make it more challenging to recognise eye colour accurately.
3) *Large number of parameters:* Each attribute has an independent model, increasing the parameters and leading to overfitting.

As shown in Figure 2, we introduce CorPAR, a new architecture for PAR. It builds on the existing W-PAR model, addressing its limitations. The architecture uses a multitask learning approach with skip connections between attribute models. These connections transfer attribute model output as supplementary inputs for subsequent models, concatenated with image features. This cascading flow of information improves learning efficiency, reduces parameters and enhances the model's ability to infer attributes.

The major challenge for PAR is the imbalance distribution of the attributes. Figure 3 shows the imbalanced distribution of upper body clothing type attributes in AVSS+RAP dataset [15] used in the proposed work. For example, there are 32.8% samples of females whereas 67.15 % samples of males in the AVSS+RAP dataset. Similar The imbalanced distribution of the dataset is handled by introducing focal loss for each attribute [20]. The following subsection discusses the proposed CorPAR architecture and AVSS+RAP dataset.

1) CorPAR ARCHITECTURE

The CorPAR Architecture is trained on the AVSS+RAP dataset customised by Galiyawala et al. [18]. The AVSS+RAP dataset combines the AVSS 2018 Challenge II [21] dataset with 14,000 annotated images and the RAPv1 [22] dataset with 41,585 images. The attributes in the AVSS+RAP dataset include gender, age, upper body clothing type, Upper body clothing colour, lower body clothing type, and lower body clothing colour.

Given a dataset $D = \{(X_1, A_1), \ldots, (X_N, A_N)\}$, where each $X_i \in \mathbb{R}^{H\times W\times 3}$ represents an image, and $A_i = [a_{i1}, a_{i2}, \ldots, a_{iM}]$ denotes the corresponding attribute vector with M attributes, the goal is to train a model f such that for any input image X_i, the model predicts the attribute vector $\hat{A}_i = f(X_i)$, where $\hat{A}_i = [\hat{a}_{i1}, \hat{a}_{i2}, \ldots, \hat{a}_{iM}]$.

The image X_i is first passed through a pre-trained backbone network, such as EfficientNet-B0, which is represented as a function f_{base}. This function maps the input image to a 1000-dimensional feature vector:

$$x_{\text{base}} = f_{\text{base}}(X_i), \quad x_{\text{base}} \in \mathbb{R}^{1000} \tag{1}$$

The model includes sequential attribute-specific branches that predict each attribute. These branches are denoted by functions f_{branch_m} for each attribute m. The input to each branch m is the concatenation of the base feature vector x_{base} with the output of the previous branch, enhancing feature representation through intra-skip connections.

For the first attribute (e.g., gender):

$$y_{\text{gender}} = f_{\text{gender}}(x_{\text{base}}) \tag{2}$$

For the second attribute (e.g., upper body clothing type), the input includes the concatenated vector of x_{base} and y_{gender}:

$$y_{\text{ubody}} = f_{\text{ubody}}([x_{\text{base}}, y_{\text{gender}}]) \tag{3}$$

This process continues for each subsequent branch:

$$y_{\text{branch}_m} = f_{\text{branch}_m}([x_{\text{base}}, y_{\text{branch}_{m-1}}]) \tag{4}$$

where: $y_{\text{branch}_m} \in \mathbb{R}^{64}$ is the output of the m-th branch.

For each attribute m, the model uses a linear output layer to generate the final prediction $\hat{a}_{im}$. This is done by applying a linear transformation on the output of the corresponding branch:

$$\hat{a}_{im} = W_m \cdot y_{\text{branch}_m} + b_m \tag{5}$$

where: $W_m \in \mathbb{R}^{64\times 1}$ is the weight vector for the m-th attribute. $b_m \in \mathbb{R}$ is the bias term.

Thus, the predicted attribute vector $\hat{A}_i$ for the image X_i is:

$$\hat{A}_i = [\hat{a}_{i1}, \hat{a}_{i2}, \ldots, \hat{a}_{iM}] \tag{6}$$

The focal loss FL_i for each attribute m is defined as:

$$FL_{im} = \begin{cases} -\alpha_m(1-\hat{a}_{im})^{\gamma_m}\log(\hat{a}_{im}), & \text{if } a_{im}=1 \\ -\alpha_m\hat{a}_{im}^{\gamma_m}\log(1-\hat{a}_{im}), & \text{if } a_{im}=0 \end{cases} \tag{7}$$

where: α_m and γ_m are hyperparameters specific to the m-th attribute. a_{im} is the true label for the m-th attribute of the i-th image. $\hat{a}_{im}$ is the predicted probability for the m-th attribute.

The total focal loss for all attributes across the dataset is given by:

$$FL_{\text{total}} = \sum_{i=1}^{N}\sum_{m=1}^{M} FL_{im} \tag{8}$$

The experiments for the proposed CorPAR have been carried out using three backbone models - the ConvNeXT-Base [24], ResNet-50 [25], and EfficientNet-B0 [26] pre-trained on ImageNet [23]. ConvNeXT-Base [24], which builds on ConvNet architectures with enhancements inspired by Transformer models, typically has 89M parameters, 15.4G FLOPs and top1 accuracy of 83.8% on ImageNet 1K dataset [23]. ResNet-50 [25], a model known for balancing performance with computational efficiency, comprises around 25.6M parameters with 3.8G floating-point operations per second (FLOPs). In contrast, EfficientNet-B0 [26], designed with a focus on efficiency through a compound scaling approach, boasts a significantly lower parameter count of approximately 5.3M and requires about 0.39G FLOPs per inference with top-1 accuracy of 77.1%. Based on the comprehensive literature review, it is evident that these three backbone models have strong feature extraction capabilities, particularly in multi-attribute recognition. Therefore, we have selected these backbones for feature extraction in our system.

The training includes pre-processing, which involves resizing the images to 224 × 224. The data is normalised using the standard ImageNet mean (0.485, 0.456, 0.406) and standard deviation (0.229, 0.224, 0.225) values for the three channels. A batch size of 128 is used for training, with two worker threads to speed up the data-loading process.

The optimiser used for training this network is stochastic gradient descent (SGD) with momentum (0.9), and weight decay (0.0005) is utilised. A two-tier learning rate schedule is applied, where new parameters have a higher learning rate of 0.005 and fine-tuned parameters (pre-trained layers) use a lower rate of 0.001. A staircase decay schedule is introduced, which reduces the learning rate by a factor of 0.1 at epoch 51, facilitating fine-tuning towards the later stages of training. The model is trained over 150 epochs with focal loss.

C. PERSON RANKING ALGORITHM MODULE

Various distance metrics assess person rankings in the AVSS 2018 Challenge-II dataset [21]. These metrics include cosine similarity, Hamming distance, their combination,

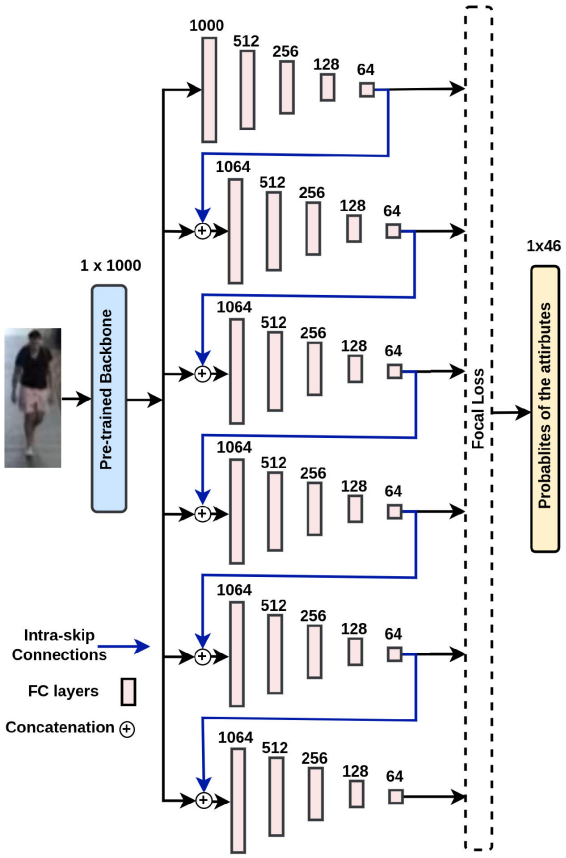


FIGURE 2. CorPAR model architecture. It uses transfer learning with backbones - ConvNeXT-Base, ResNet-50, and EfficientNet-B0 trained on ImageNet [23].

and shallow neural networks. These distance-based methods failed to establish a non-linear relationship between the two vectors. We employ a shallow neural network for PRA as it performs better than the distance-based metrics [15], [18]. The shallow network considers a single colour for upper and lower clothing, and therefore, in the case of multi-colour clothing, the colour with the highest probability given by the CorPAR is considered.

III. RESULTS AND DISCUSSION

The GPU used for the development of the system is Nvidia Quadro P5000 [27]. It is a unit with 16 GB of GDDR5X memory on a 256-bit interface, delivering a memory bandwidth of up to 288 GB/s. It has 2560 NVIDIA CUDA cores and interfaces through PCI Express 3.0 × 16. It has an active thermal solution with a maximum power consumption of 180 W.

The edge device used to infer the models is the Nvidia Jetson Orin AGX Developer Kit [28]. The 32GB variant offers 200 trillion operations per second (TOPS) (INT8) performance and has NVIDIA Ampere architecture GPU with 2048 cores and 64 tensor cores. Its GPU frequency reaches up to 1.3 GHz. The CPU consists of a 12-core Arm Cortex-A78AE 64-bit processor with 3MB L2 + 6MB L3

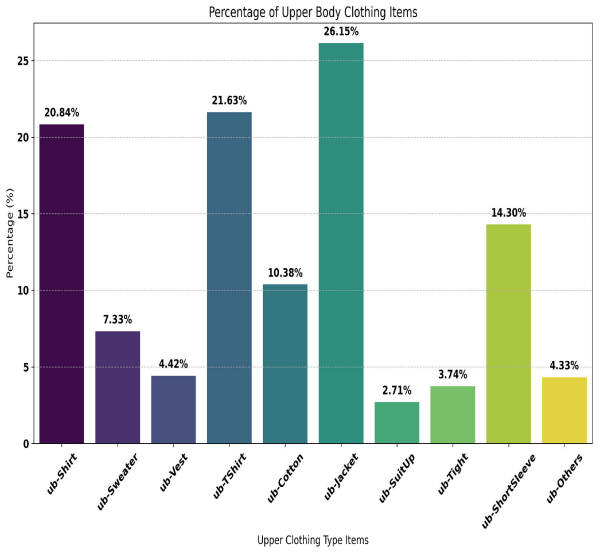


FIGURE 3. Imbalance of the attributes of upper body clothing type in AVSS+RAP dataset [15].

cache and a maximum frequency of 2.2 GHz. Deep Learning acceleration is provided by 2x NVDLA v2.0 with a maximum frequency of 1.4 GHz, and a PVA v2.0 is included for vision acceleration.

TABLE 1. Libraries and framework used in the developing PRS.

Library/Framework	Version
Python	3.8.10
Nvidia Jetpack	R35.1
PyTorch	2.1
Torchvision	0.16.1
OpenCV	4.9.0.80
ONNX	1.15.0
TensorRT	8.5.2.2
NumPy	1.24.2
Ultralytics	8.1.5
Matplotlib	3.7.4
CUDA	11.4

All experiments for each subsystem were conducted on the Nvidia Jetson AGX Orin AGX under the 15 W power mode configuration, and it can use a maximum of 2K cores. This power mode was chosen to evaluate the system's performance under realistic, resource-constrained scenarios, reflecting typical deployment conditions for edge devices. The results reported include metrics that account for both computational efficiency and accuracy, ensuring a comprehensive assessment of each subsystem's capabilities in this constrained power environment.

A. PERSON DETECTION

The PRS is built using the Pytorch Library. The PyTorch weights are converted into the TensorRT engine format using Nvidia's TensorRT library [29]. It optimises and accelerates the inference performance of deep learning models. To further enhance system efficiency, threading

is introduced, bifurcating the workload into three distinct threads. One thread is dedicated to person detection, the second handles PAR, and the third handles person ranking and image saving for storage. Table 1 shows the versions of the libraries used in the present work.

We quantise the YOLOV8s model's weight to different precision levels: FP32, FP16, and INT8. Table 2 shows a quantitative evaluation of the test set on the CrowdHuman dataset on mAP@50 and mAP@50-95 metrics on the person detection task. The results in the first two rows are for unquantised models deployed on GPU and edge devices with a precision of FP64, and the next three are weight quantisation with a precision of 32, 16, and 8 bits deployed on the Nvidia Jetson AGX Orin.

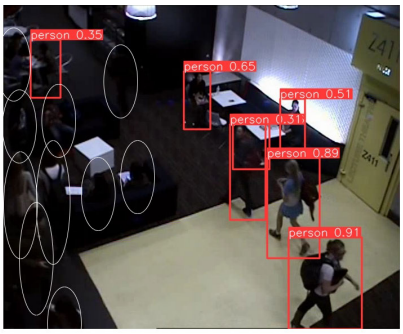
The performance metrics of the quantised models reveal a difference between FP32 and FP16 precision implementation. Further decline in performance is observed when the model is quantised to INT8, with the mAP@50-95 dropping from 0.51 to 0.48, as mentioned in Table 2. This reduction is attributed to the loss of information when model weights are quantised from floating-point to integer representation. Quantisation from FP32 to FP16 and INT8 yields higher FPS (from 40 in FP32 to 43 in FP16 and 46 in INT8), indicating that lower bit representation enhances processing speed. However, the mAP@50 and mAP@50-95 metrics decline with lower precision, showing a loss in detection accuracy. For instance, the mAP@50-95 metric falls from 0.51 in FP32 to 0.48 in INT8, showcasing that INT8 quantization sacrifices detailed detection capability for faster processing. The qualitative analysis of the numbers shown in Table 2 is observed in Figure 4a and Figure 4b.

TABLE 2. Performance metrics for the test set of the CrowdHuman dataset for person detection by weight quantisation method [17] using YOLOv8s.

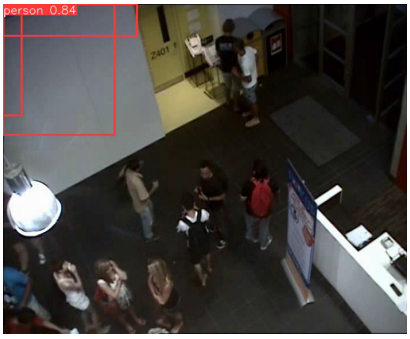
Platform	mAP@50	mAP@ 50-95	FPS
Pytorch on GPU FP64	0.78	0.51	40
Pytorch on Jetson FP64	0.78	0.51	40
TensorRT FP32	0.78	0.51	40
TensorRT FP16	0.76	0.49	43
TensorRT INT8	0.73	0.48	46

The person detection module in the PRS is one of the most essential parts, as the person detected in the video will be propagated to subsequent modules. A false positive or negative leads to noise propagation and system failure. As illustrated in Figure 4, limitations can arise from MC, particularly regarding precision and the ability to handle low-quality images. Figure 4a demonstrates that when using INT8 precision, the YOLOv8s model does not successfully detect all individuals in the scene. In Figure 4b, the YOLOv8s model, under the INT8 precision, produces false positives, incorrectly identifying objects as persons.

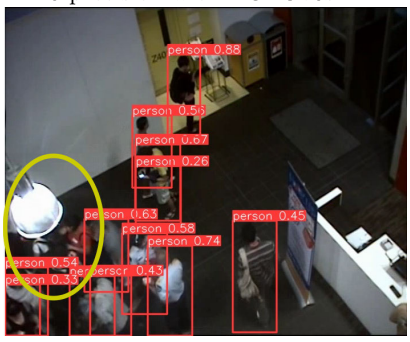
Additionally, Figure 4c reveals that even while operating in FP32 precision mode, the detection model struggles to recognise a person obscured by a lamp, with excessive lighting in the scene further compounding detection difficulty. Further



(a) Yolov8s unable to detect all the persons with INT8 precision. White circles highlight the persons missed by the detector.



(b) Generation of false positive during INT8 precision with YOLOv8s.



(c) Generation of false negative due to occlusion and excessive lighting in FP32 precision inference with model.

FIGURE 4. Issues of YOLOv8s with MC - missing detection, false positive and false negative.

analysis during inference highlighted an issue in the bounding boxes with the FP16 model. It is noted in Figure 4c that the bounding boxes are consistently smaller than expected, failing to cover the entire person, leading to significant information loss. Additionally, the model's performance is adversely affected in scenarios involving persons occluded by more than 50%

We apply only weight quantisation to YOLOv8s and have not applied layer pruning, as when the redundant weights are reduced to zero, introducing sparsity made the model error-prone and led to a deterioration in performance.

B. PERSON ATTRIBUTE RECOGNITION

The PAR model extracts attributes from the detected images. Considering the computational complexities, such as the number of parameters and FLOPs, is essential. The PAR module has them due to two factors: 1. a pre-trained backbone model. 2. fine-tuning it on the AVSS+RAP dataset. Therefore, understanding the computational complexities and accuracy is crucial for assessing PAR's efficiency and resource requirements.

1) ATTRIBUTE RECOGNITION USING CorPAR

Table 3 presents a comparative analysis between CorPAR and the existing state-of-the-art W-PAR across several backbone architectures on GPU, revealing notable enhancements CorPAR achieves. In terms of detection quality, CorPAR consistently improves the F1 score across all backbones, with ConvNeXT-Base showing a 4.1% increase to 80.64%, ResNet-50 achieving a 1.63% rise to 77.56%, and EfficientNet-B0 benefiting the most with an 8.07% improvement, reaching 79.04%. These F1 score enhancements underscore CorPAR's superior attribute recognition capability across diverse model architectures.

CorPAR also demonstrates substantial reductions in model parameters, reflecting its efficiency in minimizing redundant weights. For ConvNeXT-Base, CorPAR reduces parameters by 1.69 million, bringing the total to 92.15M. ResNet-50, the reduction is even more pronounced, with a decrease of 11.02M, resulting in streamlined 29.11M parameters. EfficientNet-B0 also benefits by reducing 2.4M parameters down to 8.84M. This efficiency is largely due to the intra-skip connections within CorPAR, which facilitate feature reuse among attribute models without redundant relearning, effectively streamlining model complexity.

Moreover, CorPAR significantly reduces the model weight-file size (MWS), emphasizing its compactness and efficiency. ConvNeXT-Base experiences a 28.8MB decrease, lowering the MWS from 728.8MB in W-PAR to 700MB in CorPAR. Similarly, ResNet-50's MWS decreases by 9.6MB to 233.4MB, and EfficientNet-B0 sees a 1.7MB reduction, reaching 70.1 MB. This storage efficiency makes CorPAR particularly suitable for deployment in resource-constrained environments. Additionally, the intra-skip connections that allow feature reuse reduce the training time by 10% compared to W-PAR, making CorPAR both computationally and time-efficient.

Table 4 illustrates the attribute-wise performance comparison between CorPAR and W-PAR across different backbones. For the ConvNeXT-Base backbone, CorPAR achieves an overall average accuracy of 94.5%, surpassing W-PAR's 93.28% by 1.22%. Similarly, with the EfficientNet backbone, CorPAR records a higher average accuracy of 94.65%, outperforming W-PAR's 92.08% by 2.57%. While W-PAR shows a marginal advantage over CorPAR with the ResNet backbone, with a difference of only 0.12%. CorPAR demonstrates overall accuracy superiority, mainly when used with ConvNeXT-Base and EfficientNet-B0 models.

2) MODEL COMPRESSION EVALUATION

MC strategies like quantisation and pruning are implemented to deploy models on edge devices. Specifically, we consider the application of unstructured pruning techniques with L1 regularisation.

A comprehensive evaluation of these MC techniques demonstrates their impact on the model's performance, as shown in Table 5. Across all three backbones, there is no significant difference between the performance of NVIDIA Quadro P5000 or Jetson Orin AGX when running uncompressed models with FP64 trained with native PyTorch. Applying quantisation decreases accuracies compared to their corresponding full-precision uncompressed models. In the ConvNeXT-Base model, the conversion from uncompressed FP64 to quantised FP32 resulted in an accuracy change of 1.27%. Similarly, for the ResNet-50 model, the change observed from an uncompressed format to a quantised FP32 is 2.05%. In the case of EfficientNet-B0, the observed shift after quantisation to FP32 was 1.4%. The Nvidia Jetson platforms use FP32 as the default precision setting in Nvidia TensorRT [29] to accelerate model inference and deployment on its hardware. So, converting FP64 to FP32 leads to some loss of precision, affecting the accuracy.

The reduction in accuracy with quantisation varies among backbones. For instance, with a change of precision from FP32 to FP16, the accuracy in CorPAR based on ConvNeXT-Base [24] drops by 2.76%, while it drops by 1.73% in ResNet-50, and by 1.7% in EfficientNet-B0 backbones. As expected, while quantising from FP32 to INT8, the drop in accuracy of CorPAR is even more: 5.78% (ConvNeXT-Base), 4.45% (ResNet-50), and 4.8% (EfficientNet-B0). This shows that the impact of quantisation is highest on the ConvNeXT-Base backbone and lower on ResNet-50 and EfficientNet-B0 models.

The results in Table 5 reveal that quantisation to FP32 and FP16 levels impacts the model's performance less. However, a significant reduction in the accuracy and F1 score is observed when the model is quantised to INT8.

We observed that CorPAR's backbones exhibit a high proportion of weight sharing - 95.48% are shared in ConvNeXT-Base [24], 83.96% for ResNet-50 [25], and 67.34% for EfficientNet-B0 [26]. Therefore, the model can be further compressed by removing redundant layers for better computational performance. We employed two successive compressions - L1 pruning with quantisation (L1PQ) to compress the model more. The results are shown in Table 5. We observe that at FP32 precision with higher compression, the accuracy of CorPAR (compared to the uncompressed value and FP64 precision on GPU) drops across all backbones - 6.33% (ConvNeXT-Base), 4.03%(ResNet-50), 1.54% (EfficientNet-B0). The change of accuracy from 32-bit (FP32) to 8-bit (INT8) results in a further drop for the model by 4% (ConvNeXT-Base), 5.71%

TABLE 3. Comparison of F1 score, parameters, and model weight-file size (MWS) between W-PAR [15] and CorPAR using AVSS+RAP dataset on GPU. Refer to p as parameters and. The boldface indicates superior results. ↑ shows increase in quantity and ↓ shows decrease in quantity.

Backbone	CorPAR			W-PAR [15]		
	F1	p (M)	MWS (MB)	F1	p (M)	MWS (MB)
ConvNeXT-Base [25]	**80.64** ↑	**92.15** ↓	**700** ↓	76.54	93.84	728.8
ResNet-50 [26]	**77.56** ↑	**29.11** ↓	**233.4** ↓	75.93	40.13	243
EfficientNet-B0 [27]	**79.04** ↑	**08.84** ↓	**70.1** ↓	70.97	11.24	71.8

(ResNet-50), 14.46% for EfficientNet-B0. We observe that accuracy drops most for the model with an EfficientNet-B0 backbone and least for ConvNeXT-Base. This is because the model with EfficientNet-B0 has the lowest weight sharing of 67.34%, and further compression causes information loss. On the other hand, CorPAR with ConvNeXT-Base has 95.48% of weight sharing, so it can tolerate additional compression without significant loss in accuracy. However, one must note that accuracy and F1-score are not the only metrics used to evaluate PAR; one should also consider memory usage, FPS, and TPR.

C. PERSON RETRIEVAL

Subsections III-A and III-B show a trade-off for individual modules when MC techniques are implemented on Jetson Orin AGX.

A comparison between CorPAR and W-PAR [15] systems, showing that CorPAR consistently outperforms W-PAR across two backbone architectures, ConvNeXT and ResNet-50, in terms of True Positive Rate (TPR) and Intersection over Union (IoU). For ConvNeXT, CorPAR achieves a TPR of 71 and an IoU of 72, surpassing W-PAR's 68 TPR and 70 IoU. Similarly, with ResNet-50, CorPAR records a TPR of 70, significantly higher than W-PAR's 55, although W-PAR slightly edges out CorPAR in IoU, with 73 compared to CorPAR's 71. This comparison highlights CorPAR's superior performance in attribute recognition tasks across backbone configurations.

Applying MC techniques to the PRS to implement over-edge devices brings various challenges. The following points elaborate on their effects, and quantitative performance is enumerated by the performance data in Table 6.

1) **Model's TPR:** The data shows that MC affects PRS models differently. For example, the ConvNeXT-Base PRS model shows a TPR of 71% without compression. However, post-compression, a 7% decline to 65% TPR is observed for model INT8 with L1PQ TensorRT on Jetson Orin AGX. Similarly, ResNet-50 PRS models start with a TPR of 71% but experience an 8% drop to 63% with L1PQ and INT8 TensorRT on Jetson Orin AGX. In the EfficientNet-B0-based PRS, designed for high efficiency, an uncompressed PRS has a TPR of 72% but sees a reduction (of 12%) to 60% with INT8 and L1PQ, highlighting the balance between performance and efficiency, albeit with a challenge in maintaining accuracy under heavy compression. Again, the PRS with ConvNeXT-Base (Highest weight sharing) backbone has the lowest drop in TPR, while EfficientNet-B0 (Lowest weight sharing) has the highest TPR drop.
2) **Resource Constraints:** Deploying models on edge devices is impacted by resource limitations, as shown by the effects on different models. The PRS with ConvNeXT-Base, for instance, reduces GPU memory usage significantly from 1.5GB (GPU and uncompressed) to 700MB (L1PQ and INT8 TensorRT Jetson Orin AGX) through compression, but at a cost to real-time performance, with FPS rates dropping from 22 to 12. The PRS with ResNet-50 also sees memory reduction from 990MB (GPU, uncompressed) to 300MB (L1PQ and INT8 TensorRT Jetson Orin AGX), with FPS dropping from 30 to 10, underscoring the complex trade-off between the model size, computational efficiency, and real-time performance. EfficientNet-B0 sees a minor reduction in memory use (558MB to 380MB) and maintains a higher FPS (26), demonstrating its suitability for edge devices.
3) **Dependency Management:** The dependency on specific deep learning libraries tailored for edge devices, like Nvidia Jetpack [30], poses challenges in versioning and compatibility, potentially hampering the deployment of new model innovations and optimisations on edge devices. For example, the current Nvidia Jetpack version is 6.0, which is supported by the Jetson Orin AGX but is not supported by Jetson Nano.
4) **Data Pipeline Bottlenecks:** In the PRS system, which employs multiple models, ensuring a seamless data flow across various hardware components, including model synchronisation, RAM, and CUDA, is critical. Lack of synchronisation among the models results in bottlenecks. It is observed that the PRS based on ResNet-50, after L1PQ to INT8, led to an uneven performance, resulting in a 1.1-millisecond asynchrony between the detection and the PAR model. This is reflected in a reduced frame rate of only 10 FPS.
5) **Thermal Management:** The continuous operation of larger models leads to thermal issues, as inferred from GPU memory and FPS in the PRS with ConvNeXT-Base model. High GPU utilisation exacerbates thermal conditions, resulting in performance throttling and reduced system performance over time. For example, when running the ConvNeXT-Based system on a Jetson Orin AGX device using FP64 in a Pytorch environment, the temperature rose to 61°C. Compared to the limited

TABLE 4. Attribute-wise comparison of W-PAR and CorPAR accuracy. Here, ub refers to upper body, lb refers to lower body, and Avg refers to Average. The boldface represents superior values. The averages are rounded to the nearest decimals.

Architecture	CorPAR			W-PAR		
Attribute	ConvNeXT-Base	ResNet-50	EfficientNet-B0	ConvNeXT-Base	ResNet-50	EfficientNet-B0
Female	**95.02**	92.37	93.51	93.49	90.39	85.68
Male	**95.01**	92.40	93.51	93.52	90.39	85.82
Avg Gender Accuracy	**95.01**	92.39	93.51	93.51	90.39	85.75
ub-Shirt	**91.18**	90.47	89.86	90.42	79.57	79.57
ub-Sweater	91.64	**92.38**	90.58	92.00	92.38	92.38
ub-Vest	**98.10**	95.20	97.55	97.81	95.20	95.20
ub-TShirt	**84.60**	83.38	82.73	84.10	78.38	78.38
ub-Cotton	**93.12**	92.03	92.26	92.81	90.04	90.04
ub-Jacket	**85.27**	83.42	83.90	84.25	73.23	74.58
ub-SuitUp	**98.02**	89.20	97.80	97.91	97.20	97.20
ub-Tight	**96.52**	80.97	95.87	95.95	95.97	95.97
ub-ShortSleeve	**93.86**	93.30	93.53	93.42	85.40	87.55
ub-Others	94.16	80.06	96.36	96.20	**97.06**	**97.06**
Avg ub Clothing Accuracy	**92.65**	88.04	92.04	92.49	88.44	88.79
ub-ColourBlack	**89.32**	87.90	88.42	87.41	53.36	88.07
ub-ColourWhite	**89.20**	87.48	87.40	85.50	74.67	74.67
ub-ColourGray	**88.10**	87.74	87.13	86.10	77.39	77.39
ub-ColourRed	**95.75**	80.68	95.57	94.81	88.89	88.89
ub-ColourGreen	95.76	95.39	**97.23**	96.36	94.07	94.07
ub-ColourBlue	**94.65**	90.66	93.78	92.84	86.40	86.40
ub-ColourSilver	**98.68**	60.72	99.72	98.72	95.42	97.75
ub-ColourYellow	**97.43**	80.65	97.13	96.47	95.65	95.65
ub-ColourBrown	**97.44**	84.30	96.80	97.30	95.30	97.10
ub-ColourPurple	**98.60**	50.49	98.11	97.32	97.49	97.49
ub-ColourPink	**98.11**	86.67	97.92	97.82	96.67	96.67
ub-ColourOrange	**98.80**	88.79	**98.80**	96.20	98.47	98.47
ub-ColourMixture	**93.16**	**92.48**	92.02	91.71	87.85	87.85
ub-ColourOther	98.34	**98.56**	97.91	96.56	98.16	97.16
Avg ub Clothing Colour Accuracy	**95.24**	84.23	94.85	94.15	89.04	91.26
lb-LongTrousers	85.68	85.93	85.30	**86.32**	46.43	83.07
lb-Skirt	**96.14**	85.16	95.65	95.43	94.89	94.89
lb-ShortSkirt	96.45	90.89	97.63	**97.99**	97.52	97.52
lb-Dress	96.10	86.71	**96.93**	96.74	96.71	96.71
lb-Jeans	**89.84**	87.03	89.6	89.59	71.98	87.96
lb-TightTrousers	**94.48**	87.22	94.21	94.13	88.88	88.88
Avg lb Clothing Type Accuracy	93.12	87.16	93.22	**93.37**	82.74	91.51
lb-ColourBlack	**89.71**	88.80	89.22	88.88	42.90	88.90
lb-ColourWhite	92.33	85.00	92.85	**97.58**	97.16	97.16
lb-ColourGray	91.99	91.25	91.10	**92.24**	90.10	90.10
lb-ColourRed	**99.03**	88.29	98.95	98.33	98.29	98.29
lb-ColourGreen	**98.67**	94.57	98.66	98.60	98.57	98.57
lb-ColourBlue	**91.11**	89.69	90.93	89.77	72.74	86.29
lb-ColourSilver	**99.97**	82.1	98.65	97.25	97.41	98.27
lb-ColourYellow	**98.44**	91.06	98.34	98.08	98.06	98.06
lb-ColourBrown	**98.2**	92.14	97.25	98.18	98.14	98.14
lb-ColourPurple	**99.77**	90.70	99.71	99.51	99.70	99.70
lb-ColourPink	95.45	87.00	**99.46**	99.21	99.20	99.20
lb-ColourOrange	95.77	99.78	**99.8**	99.78	99.78	99.78
lb-ColourMixture	**98.74**	88.37	98.65	98.50	98.37	98.37
lb-ColourOther	96.45	99.02	98.63	92.75	98.53	98.54
Avg lb Clothing Colour Accuracy	96.12	87.70	**96.99**	96.98	92.29	96.38
Avg Overall Accuracy	94.50	87.90	**94.65**	93.97	88.58	90.66

available power and resources, the high computational demand leads to performance degradation.

Based on the highlights in Table 6, we observe that with FP32 and FP16 quantisation, TPR on Jetson Orin AGX remains consistent (68%-72%) through all the backbones with the lowest GPU memory usage (540MB) and highest FPS (22-24) in EfficientNet-B0. Considering the performance on train and test sets, the impact of quantisation, smaller memory footprint and real-time implementation, we found that the PRS using CorPAR with EfficientNet-B0 is most suitable for implementation on Nvidia Jetson Orin AGX.

Figure 5 highlights the robustness of the PRS with EfficientNet-B0 backbone, L1PQ (two-level compression) at

TABLE 5. Quantitative performance evaluation of PAR model using MC techniques. Refer to Quantisation as Quant and L1 layer Pruning + Quantisation as L1PQ. in the Table.

PAR Model Backbone	MC Technique	Model / Platform	Testing Accuracy (%)	F1-Score
ConvNeXT-Base	-	Pytorch GPU	94.50	80.64
		Pytorch Jetson	94.50	80.64
	Quant.	FP32 System	93.23	80.12
		FP16 System	90.47	71.26
		INT8 System	87.45	68.24
	L1PQ	FP32 System	88.17	77.56
		FP16 System	85.47	71.26
		INT8 System	84.17	67.56
ResNet-50	-	Pytorch GPU	87.9	77.56
		Pytorch Jetson	87.9	77.56
	Quant.	FP32 System	85.12	76.32
		FP16 System	83.39	74.83
		INT8 System	80.67	71.29
	L1PQ	FP32 System	83.14	74.53
		FP16 System	82.64	72.52
		INT8 System	77.43	70.22
EfficientNet-B0	-	Pytorch GPU	94.65	79.04
		Pytorch Jetson	94.65	79.04
	Quant.	FP32 System	93.25	78.87
		FP16 System	91.55	77.45
		INT8 System	88.45	68.25
	L1PQ	FP32 System	93.11	78.54
		FP16 System	90.78	76.59
		INT8 System	78.65	66.91

TABLE 6. Quantitative performance evaluation of PRS using MC techniques. The person detection module is quantised to the precision of FP32. The values of metrics are rounded to the nearest integers.

Backbone	MC	Model & Platform	TPR	IOU	FPS	GPU Memory
ConvNeXT-Base		PyTorch GPU	71	72	22	1.5GB
	-	PyTorch Jetson	71	69	14	1.4GB
	Quant.	FP32 TensorRT Jetson	72	70	14	1.1GB
		FP16 TensorRT Jetson	69	68	15	980 MB
		INT8 TensorRT Jetson	67	65	21	902 MB
	L1PQ	FP32 TensorRT Jetson	69	48	14	850 MB
		FP16 TensorRT Jetson	68	67	12	810 MB
		INT8 TensorRT Jetson	65	61	12	700 MB
ResNet-50		PyTorch GPU	70	71	30	990 MB
		PyTorch Jetson	71	69	10	950 MB
	Quant.	FP32 TensorRT Jetson	71	70	14	600MB
		FP16 TensorRT Jetson	68	67	12	540 MB
		INT8 TensorRT Jetson	66	64	15	500 MB
	L1PQ	FP32 TensorRT Jetson	68	61	12	450 MB
		FP16 TensorRT Jetson	68	58	13	418 MB
		INT8 TensorRT Jetson	63	52	10	300 MB
EfficientNet-B0		PyTorch GPU	72	71	38	558 MB
	-	PyTorch Jetson	71	70	12	590 MB
	Quant.	FP32 TensorRT Jetson	72	70	**22**	540 MB
		FP16 TensorRT Jetson	68	67	24	540 MB
		INT8 TensorRT Jetson	64	58	26	475 MB
	L1PQ	FP32 TensorRT Jetson	67	48	21	401 MB
		FP16 TensorRT Jetson	62	43	23	392 MB
		INT8 TensorRT Jetson	60	40	26	380 MB

FP32. The green-coloured bounding box is the ground truth, and the white-coloured bounding box is the retrieved person based on the given query. Figures 5a and 5b demonstrate the system's capability to function effectively in low-light conditions. Figure 5c illustrates the system's robustness in accurately retrieving a person amidst multiple individuals wearing similar colour clothing. Figure 5c showcases the system's ability to identify the target person from compromised viewpoints correctly. Furthermore, Figure 5e and Figure 5f highlight the proficiency of the PRS in dealing with scenarios involving dense crowds, effectively navigating and extracting information from such complex environments.

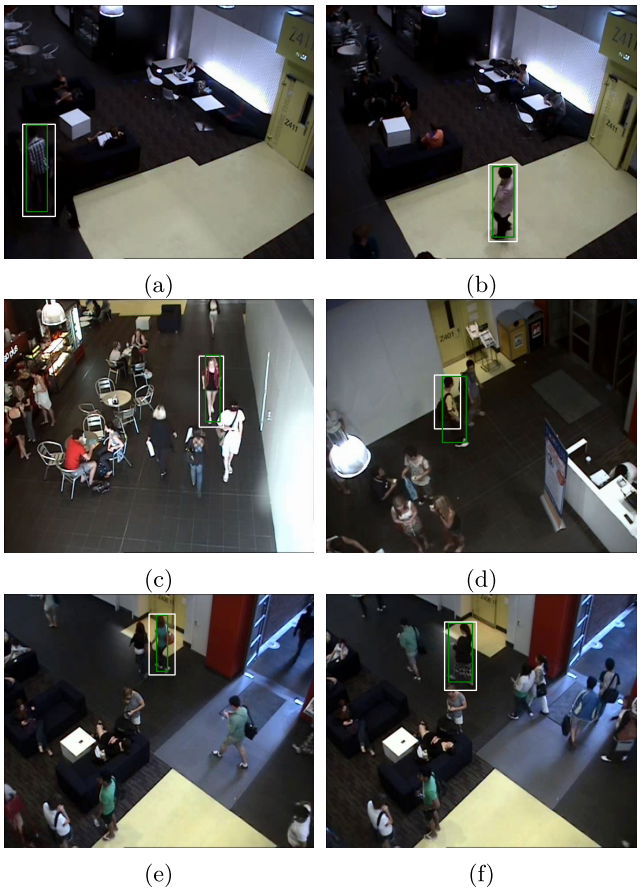


FIGURE 5. Successful retrievals under challenging conditions by CorPAR. Figure 5a and 5b represents poor illumination. Figure 5c person with similar clothing colour. Figure 5d shows a difficult viewpoint. Figure 5e, and 5f have a dense crowd.

IV. CONCLUSION

This paper has thoroughly analysed the performance of the PRS on Nvidia Jetson Orin AGX. We introduced CorPAR, a novel architecture that can be deployed within the Pytorch environment on such devices. CorPAR minimises parameter load and boosts the F1-Score, demonstrating a strategic balance between computational efficiency and accuracy.

Our investigation into various MC techniques highlighted their impact on critical performance metrics like TPR and IOU. Extensive experimentation (Table 6) revealed that while the ConvNeXT-Base backbone generally demonstrated robustness, it showed a marked decrease in TPR under combined L1 regularisation and quantisation. The ResNet-50 backbone exhibited even more significant performance declines under similar conditions, highlighting a vulnerability to extensive pruning and quantisation. CorPAR with EfficientNet-B0 backbone achieved superior outcomes compared to the W-PAR model. Its FP32 implementation for PRS resulted in a TPR of 72% at 22 FPS and GPU memory usage of 540 MB for real-time performance.

Interestingly, the disparities in performance between the NVIDIA Quadro P5000 and Jetson Orin AGX platforms are minimal, suggesting relative consistency across hardware in handling compressed models. However, we observed that pruning beyond 20% led to uneven model behaviour and a consequent drop in frame rate performance on both platforms. Our recommendations based on the performance investigation of the PRS are as follows:

1) **Balancing TPR and Frame Rate:** It is critical to achieve an optimal balance between the model's accuracy and the frame rate to meet the demands of real-time edge deployment. Our observations indicate that quantisation to INT8 precision increases false positives. From our findings in Table 6, the EfficientNet-B0 system quantised to FP32 precision demonstrated a delicate balance between TPR and FPS, leading to real-time performance. However, combining pruning with L1 regularisation and quantisation to INT8 precision while improving FPS resulted in poor TPR and IOU metrics. Thus, adjustments in this balance should be tailored to the specific criticality of the application, ensuring that the deployment is efficient and effective.
2) **Model Compression and Memory Usage:** It is advantageous to employ strategic compression to effectively decrease the model's parameter count, reducing memory demands and computational load. Table 3 shows that ConvNeXT-Base has the biggest while EfficientNet-B0 has the smallest MWS. Also, ConvNeXT-Base has more weight sharing than ResNet-50 and EfficientNet-B0. Considering FP32 weight quantisation and implementation on Jetson Orin AGX, the CorPAR with ConvNeXT-Base achieves a compression ratio (CR) of 1.27, ResNet has a CR of 1.58 CR, and EfficientNet-B0 has the smallest CR of 1.09. Therefore, it is crucial to maintain a balance between model compression and memory usage to prevent degradation in performance metrics.
3) **Model Acceleration:** In our case, TensorRT by NVIDIA significantly accelerated the model performance by 10x-20x. Such acceleration frameworks are designed to efficiently use the hardware, optimising the deployment and operation of models on edge devices. As inferred from Table 6, for a system with Resnet-50, The Pytorch Jetson model has an FPS of 10, and for the FP32 TensorRT on the Jetson model, it went to 14 frames. Similarly, for EfficientNet-B0, when accelerating the model using TensorRT, the performance shot up by ten frames. Therefore, it is advantageous to use frameworks like TensorRT.
4) **Multi-Model Synchronisation and Data Pipeline Bottlenecks:** Managing multiple models for person retrieval requires effective synchronisation, which we achieve through threading in this study. It is crucial to optimise the batch size for each model to prevent memory overflow, which can disrupt the data flow pipeline. In this paper, we have used a batch size of 32 for PAR, ensuring that it helps maintain a smooth and efficient system operation, avoiding bottlenecks in the data pipeline.

The future work of this study focuses on transitioning the person retrieval system from offline processing to an online, real-time video streaming framework, which presents several specific technical challenges. Key objectives include optimising edge devices to handle high computational demands, such as processing high-resolution video streams with minimal preprocessing latency. Addressing network latency and jitter will ensure smooth, consistent data transmission during real-time operations. Another challenge for real-time video streams is processor utilisation, which exceeds acceptable thresholds and creates a continuous overhead on edge devices. Implementing robust data compression and transmission protocols will be vital for managing large-scale video streams effectively. Additionally, integrating this system with existing surveillance or monitoring infrastructures shall be seamless while ensuring compatibility and real-time responsiveness. Successfully overcoming these challenges will enable a scalable and efficient real-time person retrieval system for unstructured environments.

REFERENCES

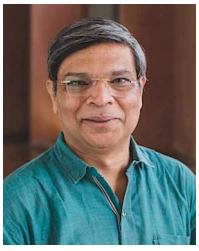
[1] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," 2017, *arXiv:1710.09282*.
[2] C. Bucilua, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2006, pp. 535–541.
[3] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4820–4828.
[4] H. Galiyawala, K. Shah, V. Gajjar, and M. S. Raval, "Person retrieval in surveillance video using height, color and gender," in *Proc. 15th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Nov. 2018, pp. 1–6.
[5] H. Galiyawala, M. S. Raval, and S. Dave, "Visual appearance based person retrieval in unconstrained environment videos," *Image Vis. Comput.*, vol. 92, Dec. 2019, Art. no. 103816.
[6] E. Bekele and W. Lawson, "The deeper, the better: Analysis of person attributes recognition," in *Proc. 14th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May 2019, pp. 1–8.
[7] H. Galiyawala, M. S. Raval, and M. Patel, "Person retrieval in surveillance videos using attribute recognition," *J. Ambient Intell. Humanized Comput.*, vol. 15, no. 1, pp. 291–303, Jan. 2024.
[8] A. Specker, M. Cormier, and J. Beyerer, "UPAR: Unified pedestrian attribute recognition and person retrieval," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 981–990.
[9] A. Specker and J. Beyerer, "Balanced pedestrian attribute recognition for improved attribute-based person retrieval," in *Proc. IEEE 13th Int. Conf. Pattern Recognit. Syst. (ICPRS)*, Jul. 2023, pp. 1–7.
[10] X. Lin, P. Ren, Y. Xiao, X. Chang, and A. Hauptmann, "Person search challenges and solutions: A survey," 2021, *arXiv:2105.01605*.
[11] M. R. Parate, K. M. Bhurchandi, and A. G. Kothari, "Anomaly detection in residential video surveillance on edge devices in IoT framework," 2021, *arXiv:2107.04767*.
[12] A. A. Suzen, B. Duman, and B. Sen, "Benchmark analysis of Jetson TX2, Jetson nano and raspberry PI using deep-CNN," in *Proc. Int. Congr. Human-Computer Interact., Optim. Robotic Appl. (HORA)*, Jun. 2020, pp. 1–5.
[13] S. Ullah and D.-H. Kim, "Benchmarking Jetson platform for 3D point-cloud and hyper-spectral image classification," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Feb. 2020, pp. 477–482.
[14] T. L. Dang, T. H. Pham, D. L. Le, X. T. Tran, H. N. Le, K. H. Nguyen, and T. T. N. Trinh, "Person re-identification on lightweight devices: End-to-end approach," *Multimedia Tools Appl.*, vol. 83, no. 29, pp. 73569–73582, Apr. 2024.
[15] J. N. Chaudhari, H. Galiyawala, M. Kuribayashi, P. Sharma, and M. S. Raval, "Designing practical end-to-end system for soft biometric-based person retrieval from surveillance videos," *IEEE Access*, vol. 11, pp. 133640–133657, 2023.
[16] G. Jocher, A. Chaurasia, and J. Qiu. (2023). *Ultralytics Yolov8*. [Online]. Available: https://github.com/ultralytics/ultralytics
[17] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun, "CrowdHuman: A benchmark for detecting human in a crowd," 2018, *arXiv:1805.00123*.
[18] H. Galiyawala and M. S. Raval, "Person retrieval in surveillance using textual query: A review," *Multimedia Tools Appl.*, vol. 80, no. 18, pp. 27343–27383, Jul. 2021.
[19] D. Li, X. Chen, and K. Huang, "Multi-attribute learning for pedestrian attribute recognition in surveillance scenarios," in *Proc. 3rd IAPR Asian Conf. Pattern Recognit. (ACPR)*, Nov. 2015, pp. 111–115.
[20] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.
[21] M. Halstead, S. Denman, C. Fookes, Y. Tian, and M. S. Nixon, "Semantic person retrieval in surveillance using soft biometrics: AVSS 2018 challenge II," in *Proc. 15th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Nov. 2018, pp. 1–6.
[22] D. Li, Z. Zhang, X. Chen, and K. Huang, "A richly annotated pedestrian dataset for person retrieval in real surveillance scenarios," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1575–1590, Apr. 2019.
[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[24] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11966–11976.
[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[26] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2019, pp. 6105–6114.
[27] *Datasheet: Quadro P5000*. Accessed: Mar. 20, 2024. [Online]. Available: https://images.nvidia.com/content/pdf/quadro/data-sheets/192195-DS-NV-Quadro-P5000-U.S.-12Sept-NV-FNL-WEB.pdf
[28] *NVIDIA Jetson AGX Orin Series*. Accessed: Jun. 19, 2024. [Online]. Available: https://www.nvidia.com/content/dam/en-zz/Solutions/gtcf21/jetson-orin/nvidia-jetson-agx-orin-technical-brief.pdf
[29] *NVIDIA TensorRT—Developer*. Accessed: Jun. 20, 2024. [Online]. Available: https://developer.nvidia.com/tensorrt/
[30] (2024). *Jetpack*. Accessed: Jun. 20, 2024. [Online]. Available: https://developer.nvidia.com/embedded/jetpack

JAY N. CHAUDHARI received the Bachelor of Technology degree in electrical engineering from the Institute of Infrastructure, Technology, Research and Management, Gujarat, India, in 2019, and the Master of Engineering degree in automatic control and robotics from The Maharaja Sayajirao University of Baroda, Gujarat, in 2022. He is currently a Senior Research Fellow on the GUJCOST Project with Ahmedabad University. His research interests include computer vision and deploying deep learning models on edge devices. He holds one Indian patent (published) and has one publication in time series forecasting.

HIREN GALIYAWALA received the bachelor's degree in electronics and communication engineering from the Sarvajanik College of Engineering, Surat, in 2007, the Master of Technology degree from the College of Engineering, Pune, in 2010, and the Ph.D. degree in engineering from Ahmedabad University, in 2022. He is currently a Senior Data Scientist with Rydot Infotech Pvt. Ltd., Ahmedabad. With over 13 years of experience, he has a strong background in computer vision, application development, testing, and engineering education. He has a diverse professional history, having served as a System Engineer with IBM, an Assistant Professor with UTU, and a Senior Research Fellow on the BRNS Project with Ahmedabad University. His research, published in IEEE, Springer, and Elsevier, is widely cited. He has mentored engineers and students in AI solutions, led teams in developing solutions using deep learning, machine learning, computer vision, and natural language processing. He holds certifications, including ISTQB and IBM Certified Solution Designer.

PAAWAN SHARMA (Senior Member, IEEE) was born in the Thar City of Bikaner, Rajasthan, in January, 1983. He received the B.E. degree in ECE from the University of Rajasthan, Jaipur, in 2005, the M.Tech. degree in communication systems from SVNIT, Surat, in 2008, and the Ph.D. (Engineering) degree from Homi Bhabha National Institute, Mumbai, in 2014. He completed his formal education in Ajmer, Rajasthan. His research interests include multidisciplinary, spanning applications/solution development in signal processing, embedded systems, pattern recognition, machine vision, and artificial intelligence. He has four Indian patents (published) and over 40 publications. He has guided/co-guided four Ph.D. scholars who have focused on computer vision, disaster management technology, and smart grid analysis. He has also worked with Wipro Technologies in the VLSI domain. He is also an Associate Professor with the Department of ICT, PDEU, and also the Dean of IT Infrastructure with PDEU, Gandhinagar. He is a Senior Member of ACM, ISSIA, and IAPR.

PANCHAM SHUKLA received the Diploma degree in electrical and electronic engineering from the Imperial College London and the Ph.D. degree from the University of London. He is currently a Faculty Member of the Department of Computing, Imperial College London. He holds more than 28 years of teaching, research, and mentoring experience at various universities in the U.K. and India. In his wide-ranging academic career, he held various administrative and leadership positions, such as the Director of Postgraduate Studies, the Director of Undergraduate Studies, and the Deputy Director of a research centre. He also received the Vice Chancellor's Award for Outstanding Contribution to Learning and Teaching and the Student Union's Award as an Outstanding Academic Staff Member. His current research interests include signal/image processing and machine learning. He enjoys reading and writing poems (in Gujarati) in his free time and is a General Secretary of Gujarati Literary Academy of the U.K.



MEHUL S. RAVAL (Senior Member, IEEE) received the bachelor's degree in electronics and telecommunication engineering (ECE) from the renowned College of Engineering, Pune, India, in 1996, the master's degree in electronics—digital systems, in 2002, and the Ph.D. degree in electronics and telecommunication engineering (ECE), in 2008. He is currently a Distinguished Faculty Member with Ahmedabad University and a Professor, with a career spanning over 27 years, marked by expertise in computer vision. His commitment to academic excellence transcends borders, with notable research stints with Okayama University, Japan, and as an Argosy Visiting Associate Professor with the Olin College of Engineering, USA, in 2016. He further enriched his global academic footprint with a visiting professorship with Sacred Heart University, Connecticut, in 2019. His scholarly contributions extend to esteemed publications and respected reviewers for publishers, such as IEEE, ACM, Springer, Elsevier, IET, and SPIE. His research has garnered support from institutions, such as the Board of Research in Nuclear Science (BRNS) and the Department of Science and Technology of the Government of India. He actively mentors students and contributes to curriculum development in leading Indian universities. He holds senior IEEE memberships and prestigious titles as a fellow of the Institution of Electronics and Telecommunication Engineers (IETE) and the Institution of Engineers (India). His substantial contributions extend to leadership roles within the IEEE Gujarat section, including the IEEE Signal Processing Society (SPS), IEEE Computational Intelligence Society, and IEEE Intelligent Transportation Systems Society's Gujarat Chapters.