RCBS-ReID: Rank-Reversal-Aware Compression and Budget-Adaptive Search for Efficient Edge Wooden Pallet Re-Identification

Yaxiong Liu[1], Guanzhi Lyu[2], Yue Yang[3,1], Jingsong Li[1], Ke Chen[4], and Yunling Liu[1(✉)]

[1] College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China
[2] College of Engineering, Nanjing Agricultural University, Nanjing 211800, China
[3] Information Technology Center, Beijing Foreign Studies University, Beijing 100089, China
[4] Chinese Academy of Quality and Inspection & Testing, Beijing 100176, China
liuyunling@cau.edu.cn

Abstract. Wooden pallet re-identification (ReID) is essential for reliable logistics traceability and warehouse management. Although deep ReID models have achieved promising retrieval accuracy, practical deployment remains limited by high-dimensional gallery descriptor storage and full-gallery matching latency. Existing feature compression methods mainly reduce descriptor size, but compression-induced approximation errors may disturb or even reverse the ranking order between positive samples and hard negatives. Meanwhile, many retrieval acceleration methods rely on fixed search and reranking budgets, causing redundant computation for easy queries and missed relevant candidates for hard queries. To address these limitations, we propose RCBS-ReID, a rank-reversal-aware compression and budget-adaptive search framework for efficient wooden pallet ReID. Specifically, Rank-Reversal-Aware Compact Encoding (RACE) estimates rank-reversal risk to guide compact subspace selection, reducing gallery storage while preserving retrieval ranking stability after compression. In addition, Dual Budget Adaptive Search (DBAS) combines coarse candidate recall with fine reranking, and adaptively allocates search and reranking budgets according to query difficulty to reduce redundant retrieval cost. Experiments on an in-house wooden pallet ReID dataset from real logistics scenarios show that RCBS-ReID achieves superior overall performance compared with 15 representative ReID methods. Compared with the baseline, RCBS-ReID reduces gallery storage by 95.0%, achieves 3.23× retrieval acceleration, and improves mAP by 1.11%.

Keywords: Re-identification, Feature compression, Retrieval acceleration, Edge deployment, Wooden pallet

1 Introduction

Modern logistics systems rely on continuous tracking of logistics carriers to ensure that objects can be reliably identified during warehousing, transportation, and management [21].

However, many carriers are passive objects that usually have no sensors, storage units, or communication modules, and therefore cannot actively provide identity information. Wooden pallets are among the most common passive carriers in supply chains. Previous studies have reported that hundreds of millions of wooden pallets are in circulation in Europe alone [2], indicating their practical importance in logistics management.

Reliable identification of individual wooden pallets is essential for logistics traceability, but it is difficult to achieve with passive carriers. Existing solutions mainly rely on external identifiers, such as barcodes, Data Matrix codes, or RFID tags [21], which require additional reading devices and maintenance costs and may fail when the identifiers are occluded or damaged. To reduce this dependence, recent studies [21, 18, 19] have explored image-based visual re-identification (ReID), which matches instances using the intrinsic visual features of wooden pallets. This strategy avoids additional identifiers, making it more suitable for low-cost and automated logistics traceability.

However, existing wooden pallet ReID studies mainly focus on recognition accuracy, with limited attention to deployment efficiency. In warehouse inspection and logistics traceability, queries are often acquired on edge devices, while the number of registered pallets continuously increases during warehouse operations. In this setting, complex feature encoders, high-dimensional gallery descriptors, and full-gallery matching respectively introduce substantial computational, storage, and retrieval costs. Therefore, practical wooden pallet ReID should not only provide reliable recognition, but also support lightweight feature extraction, compact gallery storage, and fast online retrieval.

Gallery feature compression is a direct way to reduce deployment cost. Existing methods usually reduce feature size through low-precision representation, dimensionality reduction, scalar quantization, product quantization, or binary coding [9, 10, 4, 11]. However, such compression usually approximates the original high-dimensional features using low-dimensional projections, discrete quantization, or compact codes, which may introduce approximation errors [10, 4, 3]. In ReID, retrieval depends on the similarity ranking between the query and gallery candidates. Such errors may perturb similarity scores and alter the final retrieval order. When the similarity gap between a positive sample and a hard negative is small, even a slight error may reverse their relative order, causing the positive sample to be ranked behind the hard negative and degrading retrieval accuracy. Therefore, reducing gallery feature storage while preserving retrieval ranking stability after compression becomes a key problem for deployable ReID.

Retrieval speed is another key requirement for lightweight deployment. In practical retrieval, each query needs to search a large gallery and rank gallery samples according to their similarity to the query. Full-gallery matching with continuous descriptors can provide more reliable similarity ranking, but its matching and ranking costs grow rapidly as the gallery size increases, leading to high per-query latency. In contrast, low-cost candidate recall based on indexing or quantization [11, 15, 10] is more suitable for rapidly obtaining potential matches, but its approximate search results may not provide equally fine-grained ranking. On the other hand, existing retrieval acceleration methods usually adopt fixed budgets, such as a fixed search range or a fixed number of candidates. Although fixed budgets are easy to implement, they cannot adapt well to different query difficulties. For easy queries, an excessive budget introduces redundant search and reranking costs. For hard queries, an insufficient budget may lead to incomplete candidate recall, causing correct matches to be missed during coarse recall.

To address these issues, we propose RCBS-ReID, a rank-reversal-aware compression and budget-adaptive search framework for efficient edge-oriented wooden pallet ReID, which aims to maintain low model complexity while reducing gallery storage, accelerating online retrieval, and improving retrieval accuracy. To enable lightweight feature extraction, MobileNetV4 [16] is adopted as the base encoder to reduce the parameter size and computational cost of online feature encoding. To reduce gallery storage cost, we propose Rank-Reversal-Aware Compact Encoding (RACE), which compresses high-dimensional gallery descriptors into compact low-bit descriptors. Since compression may disturb the relative order of similar candidates in ReID retrieval, RACE further introduces Reversal-Risk-Guided Subspace Mining (RRSM), which estimates rank-reversal risk from the ranking boundary between positive samples and hard negatives and uses this risk to guide key subspace selection. In this way, RACE reduces gallery storage cost while improving retrieval performance. To accelerate online retrieval, we propose Dual Budget Adaptive Search (DBAS). DBAS adopts Coarse Recall and Fine Reranking Search, where a binary index is used to rapidly recall potential matches and compact continuous descriptors are used to rerank the recalled candidates, thereby balancing candidate recall efficiency and ranking accuracy. Furthermore, to alleviate fixed-budget mismatch, DBAS introduces Index Search Budget (ISB) and Candidate Reranking Budget (CRB), which adaptively adjust the index search range and the number of reranked candidates according to the candidate distance distribution of each query. In this way, DBAS reduces redundant search and lowers online retrieval latency. We evaluate RCBS-ReID on a wooden pallet ReID dataset collected from real logistics scenarios. Results show that RCBS-ReID achieves superior overall performance over 15 state-of-the-art ReID methods. Compared with the baseline, RCBS-ReID reduces gallery feature storage by 95.0%, achieves 3.23× online retrieval acceleration, and improves mAP by 1.11% without increasing model complexity. The main contributions of this work are summarized as follows.

- We introduce RCBS-ReID, a rank-reversal-aware compression and budget-adaptive search framework for efficient edge-oriented wooden pallet ReID, which maintains low model complexity while reducing gallery storage, accelerating online retrieval, and improving retrieval accuracy.
- We propose Rank-Reversal-Aware Compact Encoding (RACE), which reduces gallery storage by compressing high-dimensional gallery descriptors into compact low-bit descriptors, and uses Reversal-Risk-Guided Subspace Mining (RRSM) to preserve important ranking relations during compression.
- We propose Dual Budget Adaptive Search (DBAS), which performs Coarse Recall and Fine Reranking Search to balance candidate recall efficiency and ranking accuracy. It further introduces Index Search Budget (ISB) and Candidate Reranking Budget (CRB), which adaptively allocate the search and reranking budgets to accelerate online retrieval.
- We construct a wooden pallet ReID dataset from real logistics scenarios and conduct extensive comparisons with 15 representative state-of-the-art ReID methods, verifying the effectiveness and superiority of RCBS-ReID.

2 Related Work

2.1 Wooden Pallet Re-Identification

Wooden pallet ReID has been studied for warehouse logistics traceability. Rutinowski et al. [18] adopted PCB to learn texture features from pallet images and verified the feasibility of pallet ReID based on wood surface patterns. Schwenzfeier et al. [21] compared cross-entropy, contrastive, and triplet losses, showing that training objectives affect pallet block matching. Rutinowski et al. [19] explored the applicability of synthetic data for pallet ReID to reduce the dependence on costly industrial data collection. Overall, existing wooden pallet ReID studies mainly focus on improving recognition accuracy, while efficiency issues for edge deployment, such as model complexity, gallery storage, and retrieval latency, remain insufficiently explored. Therefore, maintaining reliable recognition while reducing model, storage, and retrieval costs remains an important problem for practical wooden pallet ReID.

2.2 Compact Gallery Representation for Re-Identification

Compact gallery representation is essential for efficient ReID systems. PCA reduces descriptor dimensionality through linear projection, while PCA-Whitening (PCAW) further decorrelates and normalizes the projected dimensions [9]. PQ represents high-dimensional descriptors with compact codebook indices [10], and OPQ further optimizes the feature space before product quantization [4]. RaBitQ [3] proposes low-bit quantization for high-dimensional nearest neighbor search to reduce storage and computation costs. Overall, these methods can effectively reduce gallery storage. However, ReID retrieval performance is closely related to the ranking relationship between positive samples and hard negatives. When their similarities are close, approximation errors introduced by compression may reverse this ranking and degrade retrieval performance. To address this issue, we propose RACE, which introduces rank-reversal risk into compact gallery representation to reduce gallery storage while preserving retrieval ranking relations under compression.

2.3 Fast Re-Identification

Fast retrieval is a key component of efficient ReID deployment. Full-gallery continuous descriptor matching can provide reliable similarity ranking, but its distance computation and sorting costs increase rapidly as the gallery size grows. To reduce search cost, existing retrieval acceleration methods often use approximate nearest neighbor search, inverted indices, graph indices, or quantization indices to narrow the candidate search range or approximate distance computation [11, 15, 10]. These methods enable efficient candidate recall, but the returned approximate ranking may be less accurate than full-gallery continuous descriptor matching. In addition, many accelerated retrieval methods rely on fixed search ranges or fixed candidate numbers, which cannot adapt well to different query difficulties. To address these issues, we propose DBAS, which combines low-cost candidate recall with fine reranking to balance candidate recall efficiency and ranking accuracy, and adaptively allocates the index search budget and candidate reranking budget according to query difficulty, thereby reducing online retrieval latency.

3 Method

3.1 Overview

RCBS-ReID consists of a lightweight feature encoder, Rank-Reversal-Aware Compact Encoding (RACE), and Dual Budget Adaptive Search (DBAS). As shown in Fig. 1, RACE is used for offline gallery construction, while DBAS is used for online query retrieval. MobileNetV4 [16] is adopted as the lightweight feature encoder to extract real-valued descriptors and keep feature extraction efficient. Since directly storing high-dimensional gallery descriptors and performing full-gallery matching still introduce considerable storage and retrieval overhead, RACE and DBAS address these two overheads in the offline and online stages, respectively. Specifically, RACE maps high-dimensional descriptors into a compact space, selects key subspaces guided by rank-reversal risk between positive samples and hard negatives, and encodes them with low-bit codes to obtain compact gallery features. DBAS recalls candidates with a binary index, reranks them using compact continuous descriptors, and adaptively adjusts the index search budget and candidate reranking budget according to the candidate distance distribution of each query. Overall, RCBS-ReID maintains low model complexity while reducing gallery storage, accelerating online retrieval, and improving retrieval accuracy. For training, the feature encoder is optimized using label-smoothed identity classification loss and batch-hard triplet loss to learn discriminative ReID descriptors.

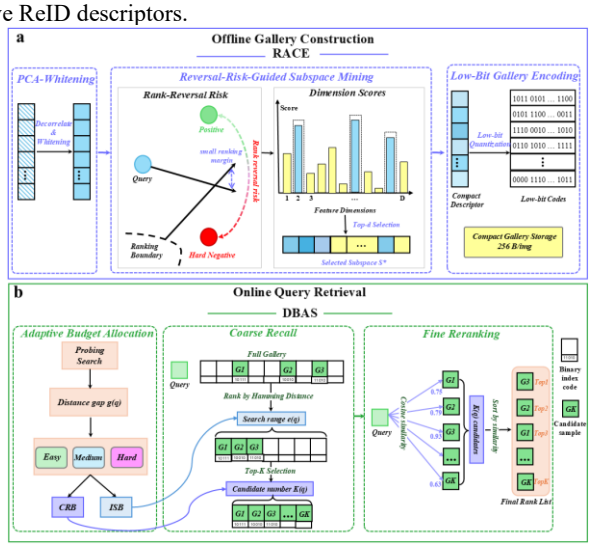

Fig. 1. Illustration of the proposed RACE and DBAS modules. (a) RACE constructs compact gallery representations during offline gallery construction. (b) DBAS performs budget-adaptive online retrieval.

3.2 Rank-Reversal-Aware Compact Encoding (RACE)

Directly storing high-dimensional FP32 gallery descriptors preserves rich discriminative information, but introduces a large storage burden. Dimensionality reduction or quantization can

reduce this burden, but may disturb the original retrieval order, causing positive samples to be ranked behind hard negatives. We argue that the key objective of compact gallery representation is not merely to reduce representation distortion, but to avoid rank reversal between positive samples and hard negatives after compression. For a query sample q, the expected ranking relation between a positive sample p and a hard negative sample n in the original feature space is formulated as:

$$s(\boldsymbol{f}_q, \boldsymbol{f}_p) > s(\boldsymbol{f}_q, \boldsymbol{f}_n). \tag{1}$$

In the compressed feature space, the corresponding rank-reversal condition is defined as:

$$s(\boldsymbol{z}_q, \hat{\boldsymbol{z}}_p) < s(\boldsymbol{z}_q, \hat{\boldsymbol{z}}_n). \tag{2}$$

This condition indicates that the positive sample is ranked behind the hard negative, which may degrade retrieval performance. Based on this motivation, we propose Rank-Reversal-Aware Compact Encoding (RACE). RACE incorporates rank-reversal risk into compact gallery encoding to reduce storage cost while preserving retrieval order. RACE consists of three steps. First, PCA-Whitening (PCAW) projects high-dimensional descriptors into a decorrelated and variance-normalized candidate space. Second, Reversal-Risk-Guided Subspace Mining (RRSM) selects ranking-preserving dimensions according to the rank-reversal risk between positive samples and hard negatives. Finally, low-bit scalar quantization encodes the selected compact descriptors into storage-efficient gallery codes.

PCA-Whitened Candidate Space. Let $\boldsymbol{f}_i = \phi(x_i) \in \mathbb{R}^D$ denote the original descriptor extracted by the feature network $\phi(\cdot)$ for the i-th sample, where D is the original feature dimension. To obtain a candidate space for compact encoding, we first center the gallery descriptors by their mean $\boldsymbol{\mu}$ and compute the covariance matrix $\boldsymbol{\Sigma}$. The covariance matrix is decomposed as:

$$\boldsymbol{\Sigma} = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^{\top}, \tag{3}$$

where $\boldsymbol{P}$ denotes the eigenvector matrix and $\boldsymbol{\Lambda}$ denotes the diagonal matrix of eigenvalues. Based on this decomposition, the PCAW projection is formulated as:

$$\boldsymbol{u}_i = \boldsymbol{W}^{\top}(\boldsymbol{f}_i - \boldsymbol{\mu}), \boldsymbol{W} = \boldsymbol{P}(\boldsymbol{\Lambda}+\epsilon\boldsymbol{I})^{-\frac{1}{2}}, \tag{4}$$

where $\boldsymbol{u}_i$ is the whitened descriptor, $\boldsymbol{W}$ is the whitening projection matrix, ϵ is a small constant for numerical stability, and $\boldsymbol{I}$ is the identity matrix. We further apply L_2 normalization:

$$\overline{\boldsymbol{u}}_i = \frac{\boldsymbol{u}_i}{\|\boldsymbol{u}_i\|_2}, \tag{5}$$

where $\overline{\boldsymbol{u}}_i$ denotes the normalized descriptor in the PCAW candidate space. This space provides a decorrelated and variance-normalized basis for compact representation. Directly retaining the first d dimensions provides a simple compact descriptor. However, this truncation mainly follows the variance-based order of PCA components and does not explicitly consider which dimensions are more important for preserving retrieval ranking after compression. Therefore, RRSM is designed to select dimensions from the perspective of ranking stability.

Reversal-Risk-Guided Subspace Mining (RRSM). RRSM uses boundary triplets to model rank-reversal risk and scores each dimension by its risk-weighted contribution to ranking

stability, thereby selecting a compact subspace that better preserves retrieval ranking relations. For a query sample q, let p denote a positive sample with the same identity and n denote a hard negative sample with a different identity. The positive-negative ranking margin in the uncompressed descriptor space is defined as:

$$\Delta_f(q,p,n) = s(\boldsymbol{f}_q, \boldsymbol{f}_p) - s(\boldsymbol{f}_q, \boldsymbol{f}_n), \tag{6}$$

where $s(\cdot,\cdot)$ denotes cosine similarity. Based on this margin, the ranking boundary between the positive sample and the hard negative is defined as:

$$B = \{(q,p,n) \mid \Delta_f(q,p,n) = 0\}. \tag{7}$$

This boundary corresponds to the case where the positive and hard negative samples are equally similar to the query in the original feature space. A smaller $|\Delta_f(q,p,n)|$ therefore indicates that the triplet lies closer to the boundary and is more vulnerable to rank reversal after compression. We define the rank-reversal risk weight as:

$$\alpha(q,p,n) = \frac{1}{|\Delta_f(q,p,n)| + \tau}, \tag{8}$$

where $\alpha(q,p,n)$ is the rank-reversal risk weight and τ is a stability term. When a hard negative has a similarity close to that of the positive sample with respect to the query, $|\Delta_f(q,p,n)|$ becomes small, increasing the risk weight $\alpha(q,p,n)$. Such triplets near the boundary therefore make larger weighted contributions to R_j through $\alpha(q,p,n)$. In the PCAW space, selecting a subset of dimensions can be viewed as projecting the descriptor onto a lower-dimensional coordinate subspace. Let $\mathcal{S}$ denote the selected dimension set. The ranking margin within this subspace can be written as:

$$\Delta_{\mathcal{S}}(q,p,n) = \sum\nolimits_{j\in\mathcal{S}} \tilde{u}_{qj}\left(\tilde{u}_{pj} - \tilde{u}_{nj}\right). \tag{9}$$

Accordingly, the contribution of the j-th dimension to this ranking boundary is defined as:

$$r_j(q,p,n) = \tilde{u}_{qj}\left(\tilde{u}_{pj} - \tilde{u}_{nj}\right). \tag{10}$$

Positive values of $r_j(q,p,n)$ enlarge the positive-negative margin, whereas negative values reduce it. Therefore, only the margin-enhancing positive contribution is accumulated:

$$r_j^+(q,p,n) = \max\left(r_j(q,p,n), 0\right). \tag{11}$$

The score of the j-th dimension is then computed by aggregating its risk-weighted positive contributions:

$$R_j = \sum\nolimits_{(q,p,n)\in\mathcal{T}} \alpha(q,p,n)\, r_j^+(q,p,n), \tag{12}$$

where $\mathcal{T}$ denotes the set of boundary triplets composed of query samples, positive samples, and hard negatives. In our implementation, $\mathcal{T}$ is constructed only from the training set, and no test identities, labels, or triplets are used for computing R_j or selecting $\mathcal{S}^*$. A larger R_j indicates that the dimension is more useful for maintaining the order of difficult positive and negative pairs.

The selected compact subspace is obtained by choosing the top d dimensions with the highest scores:

$$\mathcal{S}^* = \text{TopIdx}_d\left(\{R_j\}_{j=1}^{D}\right). \tag{13}$$

The compact descriptor of each sample is then given by:

$$\boldsymbol{z}_i = \overline{\boldsymbol{u}}_{i,\mathcal{S}^*}, \boldsymbol{z}_i \in \mathbb{R}^d. \tag{14}$$

In this way, RRSM does not simply follow the statistical order of PCA dimensions. Instead, it preserves dimensions that are more important for difficult positive-negative ranking relations, thereby reducing the risk of rank reversal after compression.

Low-Bit Gallery Encoding. After obtaining the compact descriptor $\boldsymbol{z}_i$, we apply low-bit scalar quantization to store gallery descriptors. For the j-th dimension, let $[z_j^{\min}, z_j^{\max}]$ denote the quantization range. The b-bit integer code is computed as:

$$c_{ij} = \text{clip}\left(\left\lfloor \frac{z_{ij}-z_j^{\min}}{z_j^{\max}-z_j^{\min}}(2^b-1)+\frac{1}{2}\right\rfloor, 0, 2^b-1\right), \tag{15}$$

where c_{ij} is the quantized integer code, and clip $(\cdot)$ restricts the code to $[0, 2^b-1]$. During retrieval, the stored code is dequantized as:

$$\hat{z}_{ij} = z_j^{\min} + \frac{c_{ij}}{2^b-1}\left(z_j^{\max} - z_j^{\min}\right). \tag{16}$$

The storage cost of each gallery descriptor is d × b / 8 Bytes. With d = 512 and b = 4, each gallery image requires 256 Bytes, reducing storage by 95% compared with the original 1280-dimensional FP32 descriptor. During retrieval, the query descriptor undergoes the same PCAW and RRSM projection. We then rank gallery descriptors by cosine similarity:

$$s(q,i) = \frac{\boldsymbol{z}_q^{\mathrm{T}}\hat{\boldsymbol{z}}_i}{\|\boldsymbol{z}_q\|_2\|\hat{\boldsymbol{z}}_i\|_2}. \tag{17}$$

3.3 Dual Budget Adaptive Search (DBAS)

Continuous-descriptor matching provides reliable ranking, but full-gallery matching is costly. Binary-index recall can rapidly narrow the search range, but it may not provide sufficiently accurate ranking. Therefore, DBAS first adopts Coarse Recall and Fine Reranking Search, where a binary index recalls candidates and continuous descriptors rerank them to balance retrieval efficiency and ranking accuracy. Furthermore, to alleviate fixed-budget mismatch, DBAS introduces Index Search Budget (ISB) and Candidate Reranking Budget (CRB), which adaptively adjust the binary-index search range and the number of reranked candidates according to query difficulty, thereby reducing redundant retrieval cost.

Coarse Recall and Fine Reranking Search. It first recalls candidates from the gallery using binary index codes, and then reranks them with continuous descriptors to obtain the final retrieval order. In this process, binary codes support efficient candidate recall, while continuous descriptors provide more accurate reranking. Given a query sample q and a gallery $\mathcal{G}$, let $\boldsymbol{b}_q$ denote the binary index code of the query and $\boldsymbol{b}_i$ denote the binary index code of the i-th gallery sample. Binary codes serve as coarse representations because they enable fast bit-level distance

computation. Specifically, the Hamming distance between q and the i-th gallery sample is defined as:

$$d_H(q,i) = d_H(\boldsymbol{b}_q, \boldsymbol{b}_i), \tag{18}$$

where $d_H(\cdot,\cdot)$ denotes the Hamming distance between two binary codes. Let e denote the index search budget, which controls the search range of the binary index. For query q, the gallery samples explored under budget e are denoted as $\mathcal{N}_e(q)$. The recalled candidate set is obtained by selecting the K samples with the smallest Hamming distances:

$$\mathcal{C}_K^e(q) = \mathrm{TopK}_{\,i\in\mathcal{N}_e(q)}(-d_H(q,i)), \tag{19}$$

where K denotes the candidate reranking budget, and $\mathcal{C}_K^e(q)$ denotes the candidate set recalled under the index search budget e and the reranking budget K. The binary index stage therefore reduces the full gallery to a compact candidate set. After candidate recall, compact continuous descriptors are used for fine reranking. Let $\boldsymbol{z}_q$ and $\hat{\boldsymbol{z}}_i$ denote the compact query descriptor and the dequantized compact descriptor of the i-th gallery sample, respectively. For each candidate $i \in \mathcal{C}_K^e(q)$, the reranking similarity is computed by cosine similarity:

$$s(q,i) = \frac{\boldsymbol{z}_q^\top \hat{\boldsymbol{z}}_i}{\|\boldsymbol{z}_q\|_2 \|\hat{\boldsymbol{z}}_i\|_2}, i \in \mathcal{C}_K^e(q). \tag{20}$$

The recalled candidates are then sorted in descending order of $s(q,i)$ to produce the final retrieval order. Thus, this process reduces the search range with binary codes and refines the candidate order with continuous descriptors.

Index Search Budget (ISB). ISB adaptively controls the binary-index search range by estimating query difficulty from the Hamming-distance distribution of initial candidates. It assigns smaller search budgets to easy queries and larger budgets to hard queries. Specifically, for each query sample q, an initial search is first performed in the binary index with budget (e_0, K_0). This produces the top K_0 initial candidates and their Hamming distances sorted in ascending order:

$$d_1(q) \le d_2(q) \le \cdots \le d_{K_0}(q), \tag{21}$$

where $d_1(q)$ denotes the Hamming distance of the nearest candidate, and $d_{K_0}(q)$ denotes the Hamming distance of the K_0-th boundary candidate. The candidate distance gap is then defined as:

$$g(q) = d_{K_0}(q) - d_1(q). \tag{22}$$

A larger $g(q)$ indicates clearer separation among the initial candidates and therefore an easier query, whereas a smaller $g(q)$ indicates that multiple candidates have similar Hamming distances and the query is harder. Given the easy-query ratio ρ_e and the hard-query ratio ρ_h, two thresholds are computed from the gap distribution over all queries:

$$\tau_e = Q_{1-\rho_e}(g), \tau_h = Q_{\rho_h}(g), \tag{23}$$

where $Q_\rho(\cdot)$ denotes the ρ-quantile. In DBAS, the query difficulty thresholds are determined from the distribution of $g(q)$ over the query set. Specifically, ρ_e and ρ_h control the relative

proportions of easy and hard queries, while the actual thresholds τ_e and τ_h are computed as $Q_{1-\rho_e}(g)$ and $Q_{\rho_h}(g)$, respectively. If $g(q) \geq \tau_e$, the query is assigned to the easy group. If $g(q) \leq \tau_h$, it is assigned to the hard group. The remaining queries are assigned to the medium group. Based on this partition, DBAS assigns different index search budgets and candidate reranking budgets to the three difficulty groups to balance retrieval efficiency and candidate recall. The index search budget for query q is defined as:

$$e(q) = \begin{cases} e_e, & g(q) \geq \tau_e, \\ e_h, & g(q) \leq \tau_h, \\ e_m, & \text{otherwise}, \end{cases} \tag{24}$$

where $e_e < e_m < e_h$. Easy queries use the smaller budget e_e to reduce redundant index search, hard queries use the larger budget e_h to expand the search range, and medium queries use the intermediate budget e_m.

Candidate Reranking Budget (CRB). CRB controls the number of candidates forwarded for reranking with continuous descriptors. Unlike ISB, which adjusts the binary-index search range, CRB determines the scale of continuous-descriptor similarity computation. Using the same query difficulty signal $g(q)$ as ISB, CRB assigns fewer candidates to easy queries and more candidates to hard queries. The candidate reranking budget for query q is defined as:

$$K(q) = \begin{cases} K_e, & g(q) \geq \tau_e, \\ K_h, & g(q) \leq \tau_h, \\ K_m, & \text{otherwise}, \end{cases} \tag{25}$$

where $K_e < K_m < K_h$.

Combining ISB and CRB, each query sample q obtains the budget pair $(e(q), K(q))$, where $e(q)$ controls the binary-index search range and $K(q)$ controls the number of reranked candidates. The final candidate set is then formulated as:

$$\mathcal{C}(q) = \mathcal{C}_{K(q)}^{e(q)}(q). \tag{26}$$

By replacing the fixed budget pair (e, K) with the query-adaptive pair $(e(q), K(q))$, DBAS reduces redundant search and reranking computation, thereby lowering online retrieval latency.

4 Experiments

4.1 Datasets and Evaluation Metrics

Dataset collection. We construct an in-house wooden pallet ReID dataset for real-world warehousing and logistics scenarios. Images were captured at close range using an iPhone 11 in Xiamen, Beijing, and Yantai, China. The Xiamen subset was collected from two wooden-packaging factories to cover viewpoint variations. It contains 440 pallet identities and 11,323 images, including near-frontal views, small-angle views, and large-angle side views. The Beijing subset was collected at China Agricultural University to cover natural illumination variations. It contains 58 pallet identities and 4,772 images captured under well-lit indoor, low-light indoor, well-lit outdoor, and low-light outdoor conditions, with multi-view acquisition under each

illumination condition. The Yantai subset was collected from two wooden-packaging factories to cover controlled light-source variations. It contains 277 pallet identities and 5,209 images captured under point light, area light, and ambient light, together with multi-view acquisition. In total, the dataset contains 775 pallet identities and 21,304 images. Following existing wooden pallet ReID studies [21], all identities from the three subsets are merged and then split by identity into training and testing sets using a 20%/80% ratio, with no identity overlap. Specifically, the training set contains 155 identities and 4,813 images, while the testing set contains 620 identities, 8,432 query images, and 8,059 gallery images. Overall, the dataset covers variations in acquisition sites, illumination conditions, and camera viewpoints, reflecting scene and imaging variations in practical data collection. In future work, the temporal span of data collection can be further extended to evaluate model robustness to continuous appearance degradation of the same pallets in long-term deployment scenarios.

Evaluation metrics. We evaluate both retrieval accuracy and system efficiency. For model complexity, we report the number of parameters (Params, M) and floating-point operations (FLOPs, G). For inference efficiency, we report CPU latency (ms) for feature extraction. For retrieval efficiency, we report gallery feature storage (Storage, B/img) and per-query Retrieval Time (ms/query). Retrieval accuracy is evaluated using mean average precision (mAP) and Rank-1 accuracy.

4.2 Implementation Details

All experiments are implemented with PyTorch. The model is trained on a single NVIDIA GeForce RTX 3090 Ti GPU (24 GB) under CUDA 11.3. The input images are resized to 256×128. During training, random horizontal flipping and random padding and cropping are used for data augmentation. The model is trained for 160 epochs with a batch size of 64, where each mini-batch contains 4 images per identity. AdamW is used as the optimizer with an initial learning rate of 3×10^{-4} and a weight decay of 5×10^{-4}.

4.3 Comparison with State-of-the-Art Methods

Table 1 reports the comparison between our method and representative existing methods on the Pallet ReID dataset. All methods are evaluated using the same data split and evaluation protocol for fair comparison. In terms of retrieval accuracy, our method achieves the best mAP and Rank-1, reaching 93.17% and 99.09%, respectively, with improvements of 1.11% and 0.20% over MobileNetV4. In terms of model complexity, MobileNetV2 has the fewest parameters and GhostNet has the lowest FLOPs. Although our method is not the smallest in these two metrics, it remains lightweight with 2.694M parameters and 0.121G FLOPs, achieves the lowest CPU inference time of 5.439 ms, and provides substantially higher retrieval accuracy than MobileNetV2 and GhostNet. In terms of feature storage, our method requires only 256 B/img, which is 20 times smaller than MobileNetV4 and GhostNet, 6 times smaller than PersonViT, and about 1.33 times smaller than FasterReID. In terms of Retrieval Time, PersonViT achieves the lowest value, but its mAP is 4.67% lower than ours. FasterReID has a Retrieval Time about 34.89 times that of our method. Therefore, our method achieves higher retrieval accuracy while maintaining low model complexity and reducing gallery storage and Retrieval Time, making it more suitable for practical edge wooden pallet ReID deployment.

Table 1. Comparison with representative state-of-the-art methods. Red bold and Blue bold values indicate the best and second-best results in each column, respectively.

Method	Ref.	Params (M) ↓	FLOPs (G) ↓	CPU (ms) ↓	Storage (B/img) ↓	Retrieval Time (ms/query) ↓	mAP ↑	Rank1 ↑
ResNet50 [7]	CVPR'16	23.508	2.669	58.088	8192	0.3095	86.00	97.80
MobileNetV2 [20]	CVPR'18	**2.225**	0.204	10.139	5120	0.2191	80.80	94.40
MLFN [1]	CVPR'18	32.473	2.771	67.462	4096	0.1911	85.00	96.50
GhostNet [6]	CVPR'20	4.102	**0.101**	10.783	5120	0.4363	90.96	98.80
OSNet [25]	TPAMI'22	**2.249**	0.979	22.305	2048	0.4317	87.50	97.90
PASS [26]	ECCV'22	86.764	11.621	205.970	6144	0.3258	89.30	98.30
CLIP-ReID [13]	AAAI'23	124.790	11.402	67.856	5120	**0.0327**	82.20	96.70
DCFormer [14]	AAAI'23	93.870	28.410	455.508	3072	0.1928	89.10	98.30
MSINet [5]	CVPR'23	2.328	1.093	34.160	3072	0.4962	87.80	98.10
FasterReID [23]	TPAMI'24	15.664	1.994	38.517	**340**	4.5146	81.86	96.74
RepViT [22]	CVPR'24	6.473	0.746	19.286	1792	0.4156	82.30	96.41
MobileNetV4 [16]	ECCV'24	2.694	**0.121**	**5.439**	5120	0.4363	92.06	**98.89**
FusionReID [24]	TITS'25	169.891	29.116	514.812	23552	1.0837	**92.30**	97.40
PersonViT [8]	MVA'25	22.085	2.936	18.805	1536	**0.0068**	88.50	98.00
OSKT [12]	ICCV'25	2.502	0.601	**8.938**	2048	0.0785	81.20	95.58
Ours	–	2.694	**0.121**	**5.439**	**256**	0.1294	**93.17**	**99.09**

4.4 Ablation Study on Key Components

Table 2 analyzes the contribution of each key component to the overall system performance. We use MobileNetV4 as the baseline. After introducing RACE, the feature storage is reduced from 5120 B/img to 256 B/img, yielding a 95% reduction, while mAP and Rank-1 are improved by 1.05% and 0.20%, respectively. This indicates that RACE significantly reduces gallery feature storage through compact encoding while improving retrieval accuracy. When RRSM is removed from RACE, mAP decreases from 93.11% to 92.58% and Rank-1 decreases from 99.09% to 98.97% under the same 256 B/img storage budget. This performance drop indicates that compact subspace selection guided by rank-reversal risk helps preserve effective retrieval ranking relations after compression. Since the selected subspace is determined only from training triplets and fixed for evaluation on disjoint test identities, the improvement brought by RRSM further indicates that this subspace can effectively generalize to unseen pallet identities. After introducing DBAS, Mean TopK is reduced from 8059 to 215, and Retrieval Time is reduced from 0.4178 ms to 0.0895 ms. This indicates that DBAS adaptively allocates search and reranking budgets in the coarse recall and fine reranking process, thereby accelerating online retrieval. Removing ISB increases Retrieval Time from 0.0895 ms to 0.0964 ms, indicating that ISB reduces unnecessary index search. Removing CRB increases Mean TopK from 215 to 500 and raises Retrieval Time from 0.0895 ms to 0.1037 ms, indicating that CRB reduces unnecessary candidate reranking. When RACE and DBAS are used together, the final method reduces feature storage by 95%, Mean TopK by about 37.66 times, and Retrieval Time by about 3.23 times, while improving mAP and Rank-1 by 1.11% and 0.20% over the baseline, respectively. These results show that

RACE and DBAS jointly improve retrieval accuracy while reducing gallery storage and Retrieval Time.

Table 2. Ablation study of RACE and DBAS.

Method	Storage (B/img) ↓	Mean TopK ↓	Retrieval Time (ms/query) ↓	mAP ↑	Rank1 ↑
Baseline	5120	8059	0.4178	92.06	98.89
Baseline + RACE	256	8059	0.4137	93.11	99.09
Baseline + RACE w/o RRSM	256	8059	0.4616	92.58	98.97
Baseline + DBAS	5120	215	0.0895	92.09	98.89
Baseline + DBAS w/o ISB	5120	215	0.0964	92.03	98.89
Baseline + DBAS w/o CRB	5120	500	0.1037	92.11	98.89
Baseline + RACE + DBAS	256	214	0.1294	93.17	99.09

4.5 Analysis of Different Feature Compression Methods

Table 3 compares different gallery feature compression methods using the same baseline descriptors. Compared with existing compression methods [9, 10, 4, 11], RACE achieves the best results under the 256B, 512B, and 1280B budgets, showing that compact subspace selection guided by rank-reversal risk can better preserve retrieval ranking relations and improve retrieval performance. In particular, under the 256B budget, RACE reduces gallery feature storage by 95% and improves mAP from 92.06% to 93.11%. This indicates that RACE does not simply trade accuracy for storage compression, but improves retrieval ranking quality under low storage conditions. As the storage budget increases from 128B to 1280B, the mAP of RACE improves from 89.03% to 93.87%. This suggests that larger storage budgets generally preserve more discriminative information. In contrast, under the extremely low 128B budget, all methods perform below the uncompressed baseline, indicating that excessive compression weakens feature representation. Therefore, 256B is adopted as the main setting in this work, as it substantially reduces gallery storage while maintaining and improving retrieval accuracy.

Table 3. Comparison of feature compression methods under different storage budgets. Red bold and Blue bold values indicate the best and second-best results in each column, respectively.

Budget	Method	mAP ↑	Rank1 ↑
5120B (1×)	Baseline	92.06	98.89
128B (40×)	PCAW+SQ [9]	88.76	**98.36**
	PQ [10]	**89.09**	97.64
	OPQ [4]	86.21	95.71
	RACE	**89.03**	**98.51**
256B (20×)	PCA+SQ [9]	91.27	98.70
	PCAW+SQ [9]	**92.49**	**98.91**
	PQ [10]	91.03	98.53
	OPQ [4]	90.78	98.53

	RACE	**93.11**	**99.09**
512B (10×)	PCA+SQ [9]	91.79	98.75
	PCAW+SQ [9]	**93.64**	**99.19**
	RACE	**93.85**	**99.23**
1280B (4×)	SQ8 [11]	**92.07**	**98.89**
	PCA+SQ [9]	91.98	98.83
	RACE	**93.87**	**99.21**

4.6 Analysis of Different Retrieval Acceleration Methods

Table 4 compares different retrieval acceleration methods using the same Baseline+RACE compressed descriptors. Exact Search achieves 93.11% mAP and 99.09% Rank-1, but has a Retrieval Time of 0.4137 ms/query. FAISS Flat does not bring acceleration in this setting, with a Retrieval Time of 0.8024 ms/query. In contrast, index-based retrieval methods can effectively reduce Retrieval Time. HNSW-Flat reduces Retrieval Time to 0.2224 ms/query through a graph index, IVF-Flat reduces it to 0.2454 ms/query through an inverted index, and IVF-PQ combines an inverted index with product quantization and reduces it to 0.2170 ms/query. However, the mAP values of these methods are all lower than that of Exact Search, indicating that these acceleration strategies introduce slight retrieval accuracy degradation in this setting. Compared with Exact Search, Ours reduces Retrieval Time from 0.4137 ms/query to 0.1294 ms/query, achieving about 3.20× acceleration. Meanwhile, mAP increases from 93.11% to 93.17%. This shows that DBAS further reduces Retrieval Time while maintaining retrieval accuracy.

Table 4. Comparison of retrieval acceleration methods. Red bold and Blue bold values indicate the best and second-best results in each column, respectively.

Method	mAP ↑	Rank-1 ↑	Retrieval Time (ms/query) ↓
Exact Search	**93.11**	**99.09**	0.4137
FAISS Flat [11]	**93.11**	**99.09**	0.8024
HNSW-Flat [15]	93.09	**99.09**	0.2224
IVF-Flat [11]	93.03	**99.08**	0.2454
IVF-PQ [10]	93.05	**99.08**	**0.2170**
DBAS	**93.17**	**99.09**	**0.1294**

4.7 Analysis of Gallery Size Effects

To evaluate DBAS in large-gallery retrieval, we add 32,307 images from a public pallet dataset [17] as distractors and expand the gallery to 12k, 16k, 24k, 32k, and 40k images. As shown in Fig. 2, increasing the gallery size generally decreases Rank-1 and mAP while increasing Retrieval Time. The 1280-d Non-Hashing ReID maintains the highest retrieval accuracy, but its Retrieval Time increases from 0.5378 ms/query to 3.4257 ms/query, indicating that high-dimensional real-valued features introduce high computational cost in large-gallery retrieval. The 32-bit Hashing ReID has shorter Retrieval Time on small galleries, but its mAP remains much lower,

suggesting that overly compact binary codes lose discriminative information. The 1280-bit Hashing ReID improves accuracy over 32-bit hashing, but its Retrieval Time still increases from 0.1083 ms/query to 0.5430 ms/query as the gallery grows. In contrast, our method keeps Retrieval Time nearly stable, increasing only from 0.1580 ms/query to 0.1791 ms/query, while maintaining 90.68% mAP and 97.98% Rank-1 at the 40k gallery size. Compared with 1280-d Non-Hashing ReID and 1280-bit Hashing ReID, our method achieves about 19.13× and 3.03× acceleration at 40k, respectively. These results show that DBAS adaptively adjusts search and reranking budgets to keep Retrieval Time stable in large-scale galleries.

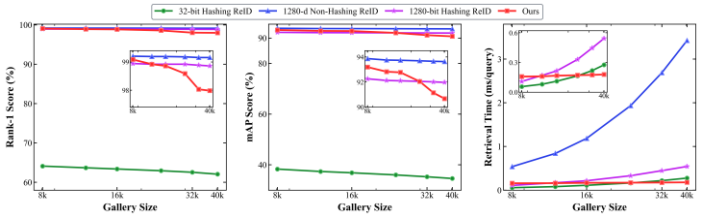

Fig. 2. Comparison of retrieval accuracy and Retrieval Time under different gallery sizes.

4.8 Parameter Sensitivity Analysis of DBAS

We conduct a parameter sensitivity analysis of DBAS by varying the initial probing budget (e_0, K_0) and the query difficulty partition ratios (ρ_e, ρ_h) while keeping the other experimental settings unchanged. As shown in Fig. 3, mAP and Retrieval Time exhibit only limited fluctuations across different parameter settings. These results suggest that DBAS is not highly sensitive to these hyperparameters and that the difficulty thresholds determined by (ρ_e, ρ_h) do not lead to noticeable retrieval accuracy degradation or excessive retrieval cost within the tested range.

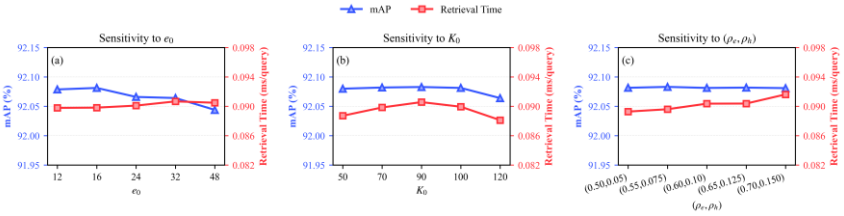

Fig. 3. Parameter sensitivity analysis of DBAS under different initial probing budgets and query difficulty partition ratios.

5 Visualization

As shown in Fig. 4, we present the Top-5 retrieval results of different methods on two query examples. Some existing methods tend to misidentify different pallets with similar wood textures and stamp marks as the same instance, leading to more incorrect matches in the Top-5 results. In addition, variations in camera viewpoint and illumination further increase the difficulty of retrieval. In contrast, Ours obtains consistently correct Top-5 retrieval results in both query examples, showing stronger discriminative ability for visually similar pallet instances.

Fig. 4. Qualitative retrieval examples of different methods. Green boxes denote correct retrieval results, and red boxes denote incorrect retrieval results.

6 Conclusion

In this paper, we propose RCBS-ReID for efficient wooden pallet ReID in practical logistics traceability. To reduce gallery storage, RACE maps high-dimensional descriptors into a compact space, selects key subspaces guided by rank-reversal risk between positive samples and hard negatives, and encodes gallery descriptors with low-bit codes. This enables compact gallery representation while preserving important retrieval ranking relations. To accelerate online retrieval, DBAS combines coarse candidate recall with fine reranking, and adaptively assigns the index search budget and candidate reranking budget according to query difficulty, reducing redundant computation caused by fixed-budget retrieval. Extensive experiments on an in-house wooden pallet ReID dataset demonstrate the effectiveness of RCBS-ReID and its superior overall performance compared with 15 representative state-of-the-art ReID methods. Compared with the baseline, RCBS-ReID reduces gallery storage by 95.0%, achieves 3.23× retrieval acceleration, and improves mAP by 1.11%. These results indicate that RCBS-ReID improves retrieval accuracy, reduces gallery storage, and accelerates online retrieval while maintaining a lightweight model, providing a practical solution for edge-oriented logistics traceability and warehouse management.

Acknowledgments. This work was supported by the Key Technologies Research and Development Program of China (Grant No. 2022YFD2001405), the Agriculture Research System of China (Grant No. CARS-28), and the National Key Research and Development Program of China (Grant No. 2023YFD2001200).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

7 References

1. Chang, X., Hospedales, T.M., Xiang, T.: Multi-level factorisation net for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2109–2118 (2018).
2. European Pallet Association: EPAL Euro Pallet (EPAL 1). https://www.epal-pallets.org/eu-en/load-carriers/epal-euro-pallet (2022), accessed: 2026-04-29.
3. Gao, J., Long, C.: Rabitq: Quantizing high-dimensional vectors with a theoretical error bound for approximate nearest neighbor search. Proceedings of the ACM on Management of Data 2(3), 1–27 (2024).
4. Ge, T., He, K., Ke, Q., Sun, J.: Optimized product quantization for approximate nearest neighbor search. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2946–2953 (2013).
5. Gu, J., Wang, K., Luo, H., Chen, C., Jiang, W., Fang, Y., Zhang, S., You, Y., Zhao, J.: Msinet: Twins contrastive search of multi-scale interaction for object reid. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 19243–19253 (2023).
6. Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., Xu, C.: Ghostnet: More features from cheap operations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1580–1589 (2020).
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016).
8. Hu, B., Wang, X., Liu, W.: Personvit: large-scale self-supervised vision transformer for person re-identification. Machine Vision and Applications 36(2), 32 (2025).
9. Jégou, H., Chum, O.: Negative evidences and co-occurences in image retrieval: The benefit of pca and whitening. In: European conference on computer vision. pp. 774–787. Springer (2012).
10. Jegou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. IEEE transactions on pattern analysis and machine intelligence 33(1), 117–128 (2010).
11. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with gpus. IEEE transactions on big data 7(3), 535–547 (2019).
12. Li, L., Qi, L., Geng, X.: One-shot knowledge transfer for scalable person re-identification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 668–677 (2025).
13. Li, S., Sun, L., Li, Q.: Clip-reid: exploiting vision-language model for image re-identification without concrete text labels. In: Proceedings of the AAAI conference on artificial intelligence. vol. 37, pp. 1405–1413 (2023).
14. Li, W., Zou, C., Wang, M., Xu, F., Zhao, J., Zheng, R., Cheng, Y., Chu, W.: Dc-former: Diverse and compact transformer for person re-identification. Proceedings of the AAAI Conference on Artificial Intelligence 37(2), 1415–1423 (2023).
15. Malkov, Y.A., Yashunin, D.A.: Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. IEEE transactions on pattern analysis and machine intelligence 42(4), 824–836 (2018).
16. Qin, D., Leichner, C., Delakis, M., Fornoni, M., Luo, S., Yang, F., Wang, W., Banbury, C., Ye, C., Akin, B., et al.: Mobilenetv4: Universal models for the mobile ecosystem. In: European conference on computer vision. pp. 78–96. Springer (2024).
17. Rutinowski, J., Chilla, T., Pionzweski, C.: palletblock-32965–a chipwood re-identification dataset. Zenodo DOI 10 (2022).

18. Rutinowski, J., Pionzewski, C., Chilla, T., Reining, C., Ten Hompel, M.: Towards re-identification for warehousing entities-a work-in-progress study. In: 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). pp. 1–4. IEEE (2021).
19. Rutinowski, J., Vankayalapati, B., Schwenzfeier, N., Acosta, M., Reining, C.: On the applicability of synthetic data for re-identification. arXiv preprint arXiv:2212.10105 (2022).
20. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4510–4520 (2018).
21. Schwenzfeier, N., Rutinowski, J., Hesenius, M., Reining, C., Acosta, M.: Generating embedding spaces for re-identifying pallets from their chipwood patterns. Engineering Applications of Artificial Intelligence 126, 106905 (2023).
22. Wang, A., Chen, H., Lin, Z., Han, J., Ding, G.: Repvit: Revisiting mobile cnn from vit perspective. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15909–15920 (2024).
23. Wang, G., Huang, X., Gong, S., Zhang, J., Gao, W.: Faster person re-identification: One-shot-filter and coarse-to-fine search. IEEE Transactions on Pattern Analysis and Machine Intelligence 46(5), 3013–3030 (2023).
24. Wang, Y., Zhang, P., Liu, X., Tu, Z., Lu, H.: Unity is strength: Unifying convolutional and transformeral features for better person re-identification. IEEE Transactions on Intelligent Transportation Systems 26(3), 3713–3723 (2025). https://doi.org/10.1109/TITS.2024.3521974.
25. Zhou, K., Yang, Y., Cavallaro, A., Xiang, T.: Learning generalisable omni-scale representations for person re-identification. IEEE transactions on pattern analysis and machine intelligence 44(9), 5056–5069 (2021).
26. Zhu, K., Guo, H., Yan, T., Zhu, Y., Wang, J., Tang, M.: Pass: Part-aware self-supervised pre-training for person re-identification. In: European Conference on Computer Vision. pp. 198–214. Springer (2022).