Multi-camera People Tracking With Mixture of Realistic and Synthetic Knowledge

Quang Qui-Vinh Nguyen¹,², Huy Dinh-Anh Le¹,², Truc Thi-Thanh Chau¹,², Duc Trung Luu¹,², Nhat Minh Chung¹,², Synh Viet-Uyen Ha¹,²,∗

¹School of Computer Science and Engineering, International University, Ho Chi Minh City, Vietnam
²Vietnam National University, Ho Chi Minh City, Vietnam

Abstract

This paper presents a solution for Track 1 of the AI City Challenge 2023, which involves Multi-Camera People Tracking in indoor scenarios. The proposed framework comprises four modules: Vehicle detection, ReID feature extraction, single-camera multi-target tracking (SCMT), single-camera matching, and multi-camera matching. A significant contribution of our approach is the introduction of ID switch detection and ID switch splitting using the Gaussian mixture model, which efficiently addresses the problem of tracklets with ID switches. Furthermore, our system performs well in matching both synthetic and real data. The proposed R-matching algorithm performs exceptionally well in real scenarios despite being trained on synthetic data. Experimental results on the public test set of 2023 AI City Challenge Track 1 demonstrate the efficacy of the proposed approach, achieving an IDF1 of 94.17% and securing 2nd position on the leaderboard. Codes will be available at https://github.com/nguyenvinhquang/Multi-camera-People-Tracking-With-Mixture-of-Realistic-and-Synthetic-Knowledge

1. Introduction

Multi-Target Multi-Camera (MTMC) tracking approaches pose significant challenges as they require solving multiple computer vision problems, including person detection, single-camera multi-target tracking, and person re-identification. These challenges involve addressing variations in camera resolution, distance, view angle, non-overlapping camera views, crowded areas, and changes in illumination. Furthermore, due to the limited availability of labeled datasets, multi people multi-camera tracking remains a challenge.

∗Corresponding author. Email: hvusynh@hcmiu.edu.vn

The 7th AI City Challenge workshop’s AIC23 [34] Track 1 dataset aims to address these challenges by combining real and synthetic data to track people across multiple cameras. However, the dataset’s indoor setting poses additional challenges as people can occur multiple times under each camera, which is not typically the case in urban environments. Additionally, the dataset’s training data is entirely synthetic, making it challenging to train models with discriminative features for real data. A desired solution ought to carefully address not only the domain gap between synthetic test data and training data but also generalize on the real-life test data with suitable constraints.

In this paper, we present a new MCMT tracking system designed to track multiple people across different cameras in an indoor scenario. Existing multi-object multi-camera tracking systems [15], [17], [6], and [43] only track objects until they exit the camera’s field of view, making it difficult to track them if they re-enter later. In contrast, our framework can track people even if they exit and re-enter the camera’s field of view. Additionally, our framework detects when a tracklet has two different user IDs and adjusts the tracklet to improve the results. Most importantly, our proposed solution can be applied on both real and synthetic test settings, despite having been trained on only synthetic data.

In order to accomplish the above, our proposed system presents several technical contributions, most notably including: (a) Our detector and single-camera tracking module can detect and track people effectively in both synthetic and real data. Additionally, we have incorporated a Gaussian Mixture Model (GMM) to alleviate the problems of ID switches. (b) Our single-camera matching and multi-camera matching modules perform well on both synthetic and real data. For real data, we introduced an R-matching algorithm to tackle the domain gap from the synthetic training set, significantly improving the IDF1 scores by 8.46%. (c) Our system’s performance has been evaluated in the Multi-camera people tracking track of the 2023 AI City
2. Related Works

Various designs for an MTMC tracking system have been proposed over the recent years, as summarised by Naphade et.al. [33][32]. Authors have typically followed the aforementioned processes: (1) Object Detection, (2) Multi-target Single-camera Tracking, (3) Appearance Feature Extraction, and (4) Cross-Camera Tracklet Matching. The performance of a particular design apparently correlates with how well authors can develop contrastive models for extracting appearance features and constrain the data domain’s search space [29][55][47].

2.1. Object Detection

An object detection model is essential in determining vehicle positions throughout a camera image. Many state-of-the-art models have been proposed that include single-shot detectors such as YOLOv4 [1], YOLOv5 [10], YOLOv6 [24], YOLOv7 [44] CenterNet [9], and EfficientDet [42] to directly output object positions alongside their classes, and two-shot detectors Mask-RCNN [11] and Cascade-RCNN [3] that rely on the generation of bounding box priors before classifying them. In the MTMC vehicle tracking literature, authors [29][47] have leveraged pre-trained models’ generalization capabilities without training on the test set with good results.

2.2. Multi-Target Single-Camera Tracking

The aim of tracking is to identify the trajectory, i.e. a set of bounding boxes, of each ID using the detection result. Most recent works on MOT are categorized into two methods: online and offline, which differ in the way of observation processing. Online approaches [20][58][59][48][56][39][46] utilize the bounding boxes on the current frame to extend the existing trajectories, which results in a short processing time, but lower accuracy. On the other hand, offline approaches [40][36][52][23][16][41][5][50][2][51] gain higher accuracy due to optimizing the solution for linking all of the bounding boxes in the video with different trajectories. In order to improve the performance on switching ID and missing tracking occluded objects cases, DeepSORT [46], an online tracking framework, uses a CNN model that is trained on a large-scale person ReID dataset to learn a deep association metric which combines motion and appearance information. the Yang et al. [53] modified DeepSORT to deal with occlusion and applied forward and backward tracking in time. MedianFlow [22], an offline tracking framework, also performed forward and backward tracking in time and then compared the two trajectories to detect the tracking failures based on the assumption that the tracklet is independent of the direction of time flow. Li et al. [25] implemented the modified version of MedianFlow to tackle the cases that the object moves too fast or moves with a rapidly direction-changing trajectory. As this version is combined with Efficient convolution operators to evaluate the correlation or similarity of two signals, the ID switches could be reduced.

2.3. Image-based Re-Identification

Vision-based re-identification refers to recognizing the same object across different images or videos captured from different cameras, which take a critical role in MTMC tracking problems. The challenging problem is that depending on each camera and point in time, lighting conditions, occlusions, pose variations, and camera viewpoint could not be invariant, which results in the visual difference of a target vehicle. The state-of-the-art techniques in image-based reID can be categorized into two main types: feature-based and deep learning-based methods. Feature-based methods [35][28][7] extract handcrafted features, i.e. the features containing the information from images such as color, texture, and shape information, and use them for matching. Regarding deep learning-based methods, many approaches [12][19][49] gain great performance by implementing CNN-based models to learn robust feature representations. However, CNN-based methods still have limitations as they focus on a local neighborhood and lose information due to downsampling operations. Thus, recent approaches [13][8] deployed Transformers to improve the result by the ability to capture global relations. For example, TransReID [13] encodes an image as a series of patches and establishes a transformer-based strong baseline with a few essential enhancements. Besides, to tackle the low-diverse and limited dataset, many GAN-based ReID works [4][62][30][61][27] have been proposed.

2.4. Cross-Camera Tracklet Matching

For MTMC tracking, an appropriate method to associate tracklets across cameras is indispensable. Many approaches [57][37][26][21][18][38] principally utilize the embedding feature vectors of IDs to compute the appearance similarity, then evaluate the result to match the tracklets. However, relying solely on appearance features may be prone to ID switches. To enhance the performance, the majority of works [14][6][25][53][54][60] combine different constraints such as camera topology, temporal information, motion rules, etc.

In this work, we improve performance by first performing single-camera matching and then using the results to enhance multi-camera matching rather than matching multiple cameras directly.

3. Methodology

This section introduces our proposed frameworks, which consist of five components, namely, person detection,
single-camera tracking, single-camera matching, and multi-camera matching. An overview of our framework is presented in Figure 1.

3.1. Person Detection-based Continuous Tracking

Our first major component is designed to track people in a single camera robustly. Motivated by the problems of occlusion (e.g., two people overlapping each other) and intermittent tracking (e.g., temporarily missing detections), we aim to robustly tackle ID switches and maintain consistent track trajectories with a detection-based tracking framework. Hence, we propose a 2-stage approach via (1) scene-conditioned person detection and (2) continuous tracking with a mixture model.

3.1.1 Scene-conditioned Person Detection

Person detection plays a crucial role in initiating Multi-camera people tracking. Our approach utilizes YOLOv5, a state-of-the-art network, specifically the pre-trained YOLOv5x6 model on the COCO dataset for object detection. Our approach seeks to address a major challenge: the domain gap between the synthetic dataset and the pre-trained COCO dataset used to train the model. To address this, we propose to develop meta-data for denoting scenes of real and synthetic scenes and apply an appropriate detection model:

**On real-life scenes:** We rely solely on the pre-trained model to achieve optimal performance. This is because the pre-trained model has already been trained on the COCO dataset to capture the features of a real person and the limited domain gap between real-life scenes in certain settings.

**On synthetic scenes:** We fine-tuned our model using the synthetic animated people dataset using the NVIDIA Omniverse Platform, particularly adapting Yolov5x6 to a similar test domain. By leveraging pre-trained patterns in the model, the model is tuned for person detection on the synthetic domain’s properties.

Regardless of the scene, if an input frame \( I_t \) captured at a specific time step \( t \), we can extract a set of \( m \) detections denoted as \( D_t := \{d_1, d_2, d_3, ..., d_m\} \). Each detection \( d_i \) can be represented by a tuple \( d_i := (x_i, y_i, w_i, h_i, c_i, t) \), which represents the \( i \)-th bounding box coordinates with the centre at \( x \) and \( y \), and size of width \( w \), height \( h \); finally \( c_i \) represents the confidence score of the bounding box, which we used to ensure that \( c_i > c_t \). In our case, \( c_t = 0.5 \).

3.1.2 Continuous Person Tracking on Mixture Model

Our Single-Camera Multi-Target Tracking system utilizes mostly reliable detection results, and we use features extracted by models in subsection 3.2 to associate targets across video frames using the tracking-by-detection paradigm. For our case, we adapted the DeepSORT tracker as our baseline method and implemented advanced techniques as mentioned in [53], such as occlusion handling, interpolating missing detections with Kalman-Filter-predicted boxes, and forward-backwards tracking. However, since it is common for individuals to walk in unpredictable patterns such as straight lines, stopping, turning around, and continuing on, or for one person to stop and another to pass closely by. We propose a Gaussian Mixture Model-based approach for splitting ID-switching tracklets. In particular, suppose a tracklet \( t \) is denoted by \( t := \{(d_1, f_1), (d_2, f_2), ..., (d_k, f_k)\} \), where there are \( k \) detections for this tracklet and \( f_i = f(d_i) \) denotes the extracted feature vector of the \( i \)-th detection.

Since each tracklet should belong to only one identity, the distribution of the features should follow a Gaussian distribution, with a single mean \( \mu \) and variance \( \Sigma \) that captures the overall characteristics of the tracklet. Therefore, we propose an ID switch detection approach to identify the tracklet where ID switching occurs. Then, we utilize ID Switch splitting to split this tracklet into two separate tracklets by modelling each tracklet’s feature set in a Gaussian Mixture Model. In particular, given a feature set, we model it as a Gaussian Mixture Model through Expectation Maximization, i.e. minimizing the likelihood:

\[
L(\theta) = \prod_{i=1}^{n} \sum_{j=1}^{k} \pi_j \mathcal{N}(f_i | \mu_j, \Sigma_j)
\]

where \( \theta = (\pi, \mu, \Sigma) \) are the mixture model parameters, with \( \mathcal{N}(f_i | \mu_j, \Sigma_j) \) as the Gaussian probability density function with mean \( \mu_j \) and covariance matrix \( \Sigma_j \). \( \pi_j \) is the mixture coefficient for the \( j \)-th Gaussian component.

**ID switch detection:** With \( n = 2 \), we constructed two Gaussian distributions on \( t \). Next, we compute the cosine distance between the mean points of these two distributions. If the distance is found to be less than a pre-defined threshold (which we set to 0.4), we can infer that an identity switch has occurred.

**ID switch splitting:** Let \( \theta_1 = (\pi_1, \mu_1, \Sigma_1) \) and \( \theta_2 = (\pi_2, \mu_2, \Sigma_2) \) be the two Gaussian distributions obtained from GMM, to split the tracklet, we use the longest subsequence of consecutive bounding boxes in the higher-weighted Gaussian distribution, starting from the first box. We assign these boxes to the first person and the remaining to the other. Specifically, if \( \pi_1 > \pi_2 \), let \( s \) and \( e \) be the start and end indices of the longest subsequence of consecutive bounding boxes that are clustered to \( \theta_1 \), then we assign the bounding boxes in \( \{(d_s, f_s), (d_{s+1}, f_{s+1}), ..., (d_e, f_e)\} \) to the first person and the remaining boxes to the other person, and the same otherwise when \( \pi_1 \leq \pi_2 \). This splitting strategy alleviates ID switches and improves tracking performance.
3.2. Person Re-identification

3.2.1 Feature Extractor

Re-identification is a critical component of Multi-Camera Multi-Target (MCMT) tracking, as it relies on obtaining each individual’s reliable and discriminative appearance features. To achieve this, we explore two types of deep feature extraction methods: (1) transformer-based and (2) CNN-based for our feature extraction model. For transformer-based methods, we use TransReID, while for CNN-based methods, we employ a range of models such as ResNet [12], ResNeXt [49], and HRNet [45]. After that, we apply bags of tricks from [31], which achieves state-of-the-art results in the re-identification field, to train our ReID model. The input image was resized to 256 × 128 in the training and feature extraction stages. We apply some data augmentation for the preprocessed input data, such as random horizontal flip, random erasing and random padding.

For the feature extraction stage, we generate a global feature with the dim of 2048 before batch normalization neck as the final output of the input image. We simply concatenate the feature extract from each model above for the ensemble ReID feature. As mentioned previously, we refer to our feature extractor as \( f(\cdot) \), which represents either our use of TransReID, ResNet50, ResNeXt101, HRNet or an ensemble of them.

3.2.2 Objective Losses

We jointly used (1) an ID loss function using cross-entropy loss with label smoothing, and (2) a contrastive loss function using triplet loss for the optimization.

**Regarding ID loss**, the probability that person image \( x \) corresponds to person \( i \) is denoted as \( p(i|x) \). Let the true person ID being represented by \( y \), the cross-entropy loss with label smoothing is defined as:

\[
L_{ID} = \mathbb{E}_{x,y} \left[ \sum_{i=1}^{N} -q(i|y) \log [p(i|x)] \right]
\]

such that \( q(i|y) \) is the smoothed label distribution:

\[
q(i|y) = \begin{cases} 
1 - \frac{N-1}{N} \varepsilon & \text{if } i = y \\
\frac{\varepsilon}{N} & \text{otherwise}
\end{cases}
\]

where \( N \) denotes the number of persons in the training set. At the same time, \( \varepsilon \) is a small positive constant that regulates the smoothing level applied to the label distribution. This smoothing technique prevents the model from overfitting to person IDs of the training set.

**Regarding Triplet loss**, the goal is to minimize the distance between an anchor sample \( x^a \) and a positive sample \( x^p \) while maximizing the distance between the anchor and a
negative sample $x^n$. This is achieved by setting a margin $m$ and optimizing the following loss function for $N$ samples:

$$L_{tri} = \sum_{i=1}^{N} \max (m + d(f^n_i, f^n_j) - d(f^n_i, f^n_p), 0)$$ (4)

Here, $f^n_a, f^n_p, f^n_n$ denote the feature embeddings of the anchor, positive, and negative samples, respectively, and $d(\cdot)$ calculates the distance between two feature embeddings. Hence, the combined loss function that we used is:

$$L_{reid} := L_{ID} + L_{tri}$$ (5)

### 3.3. Single-camera Tracklet Matching

In dealing with inconsistent tracking trajectories within a camera, e.g., a person moving out of the view and back again, we propose a single-camera matching approach to cluster-generated tracklets in terms of appearance feature. The purpose of single-camera tracklet matching is to generate groups of tracklets originating from the same person id, within the same camera.

To match the people under the camera, we find the group of tracklet belonging to one person by clustering the tracklets together based on their similarities. Suppose the function $f(\cdot)$ when applied on a tracklet will extract the average feature vector across all features, then the similarity of two tracklets $t_i$ and tracklet $t_j$ is determined by the cosine distance formula:

$$\text{dist}(t_i, t_j) = 1 - \frac{f(t_i) \cdot f(t_j)}{\|f(t_i)\| \|f(t_j)\|}$$ (6)

For all pairs of tracklets, we can then generate a single-camera distance matrix of all tracklets,

$$D := \begin{bmatrix}
\text{dist}(t_1, t_1) & \cdots & \text{dist}(t_1, t_N) \\
\vdots & \ddots & \vdots \\
\text{dist}(t_N, t_1) & \cdots & \text{dist}(t_N, t_N)
\end{bmatrix}$$ (7)

Furthermore, if the sequence in terms of time steps between two tracklets $t_i, t_j$ intersects, then in $D_{i,j} \leftarrow \infty$, tracklets will then be clustered together based on the distance matrix $D$.

**In synthetic scenes**, our approach has a high discriminative ability to distinguish between the features of different people, the accuracy of clustering using traditional algorithms on this dataset is significantly high. Hence, the clustering approach can perform robustly. For synthetic scenes, the set $R$ of short tracklets are simply discarded.

**In real-life scenes**, however, the feature model’s ability to discriminate between individuals with different features for inter-class persons is unclear, as it is trained on the synthetic person dataset. Figure 3b depicts the similarity score between two individuals wearing white shirts is 0.85. Hence, during clustering, inter-class persons may be incorrectly grouped together. To address this issue, we propose an algorithm that can overcome this issue. First, we select the frames with the highest number of people. Then,
we construct a graph for the individuals in each frame, with vertices representing people and edges determined by their cosine similarity. Then, we select the frame with the smallest sum of edges as the initial cluster. Finally, we employ the algorithm depicted in algorithm 1 to merge tracklets that fit into the same cluster.

After determining the group of clusters, denoted by $C_v$ as the $v$-th set of different tracklets, the feature of each cluster will calculate based on the following:

$$f(C_v) = \frac{1}{|C_v|} \sum_{t_i \in C_v} f(t_i)$$  \hspace{1cm} (8)

We treat each cluster $C_v$ as a gallery list and all the temporarily removed tracklets (mentioned above) as a query list $R$. Then we simply solve a ReID problem with the method outlined in [31] to find the best matches for each sample in $R$ and each cluster $C_v$. This enables us to group both long and short tracklets in one cluster from the same person in a single camera. We refer to this extension for real-life scenes as R-matching.

![Figure 3](image-url)  
Figure 3. High inter-class similarity. The red line indicates that two individuals have different IDs in both figures, while the green line represents a match. In Figure (a), there are different individuals who happen to be wearing similar clothing. If similarity scores were used for matching, it would result in an incorrect match. The same issue occurs in Figure (b).

### 3.4. Multi-camera Tracklet Matching

The idea of multi-camera tracklet matching is to match the person’s ID across multiple cameras. Here, the person id in each camera was represented by the cluster in that camera. We propose using the aforementioned cluster feature calculations to perform Agglomerative Clustering. Therefore, the distance between two clusters was defined by:

$$\text{dist}(C_v, C_u) = 1 - \frac{f(C_v) \cdot f(C_u)}{\|f(C_v)\| \|f(C_u)\|}$$  \hspace{1cm} (9)

It follows that the distance matrix of $N \times N$ pairwise cluster appearance distances is:

$$S := \begin{bmatrix} \text{dist}(C_1, C_1) & \cdots & \text{dist}(C_1, C_N) \\ \vdots & \ddots & \vdots \\ \text{dist}(C_N, C_1) & \cdots & \text{dist}(C_N, C_N) \end{bmatrix}$$  \hspace{1cm} (10)

Persons in the same camera cannot be in the same group after clustering. Therefore, we redefine their distance for pairs of people with the same camera. For each cluster $C_i$ and $C_j$ that come from the same camera, we redefine its distance on matrix $S_{i,j} = \infty$. This ensures that clusters from
the same camera are not grouped together during clustering. Finally, we use agglomerative clustering on the updated distance matrix $S$ to group these clusters.

4. Experiments

4.1. Dataset

The dataset used in this challenge comprises both real and synthetic data, totaling 2,607,781 frames across 22 different scenarios. Specifically, ten scenes are dedicated to training, with 4,375,736 bounding boxes and 71 unique person IDs. Similarly, ten scenes are reserved for validation, consisting of 1,950,917 bounding boxes and 35 unique person IDs. In terms of testing, there are two types of data to consider. The first one is Scene 001, which includes 388,671 frames of real data. The second type is synthetic data, which accounts for 648,360 frames.

4.2. Evaluation Metrics

The F1 score of people identity (IDF1) is used to assess how well multi-camera people tracking performs. IDF1 calculates the proportion of correctly identified detection in relation to the average number of ground-truth and computed detection. The AI City Challenge evaluation system will present IDF1, IDP, IDR, Precision (detection), and Recall (detection).

4.3. Implementation Details

We use Pytorch for our main framework. The experiments are performed on one Quadro RTX 6000 with 24GB.

**Re-Identification** In our ReID experiments, we tested both Transformer-based and CNN-based models. Specifically, we employed the TransReID base and TransReID with Jigsaw Patch Module (JPM) and Side Information Embeddings (SIE) from [13]. For our CNN-based models, we used ResNet50, HRNetW48, ResNet101, and ResNeXt101 ibn_a. Our optimization strategy used Stochastic Gradient Descent (SGD) with a base learning rate of 1e-4 and the Cosine Annealing scheduler. During training, we use a batch size of 96 and 4 IDs per batch.

**Detection** We fine-tune our model on the training data for one epoch. In the inference stage, we use the fine-tuned model for synthetic data. For the S001, which is real data, we only use the pre-trained model on the COCO dataset.

4.4. Parameter choosing

Our proposed system considered and optimized several factors to achieve high performance in single-camera and multi-camera matching. These included the selection of an appropriate number of clusters and the identification of a reliable zone. Additionally, we carefully chose the appropriate ReID model to extract features for accurate matching.

4.4.1 Reliable region selection

We pre-defined the regions that offer comprehensive information about the person’s appearance. These regions should capture the complete body parts without any occlusions from static objects, thereby preventing situations where only a single body part is visible. Subsequently, we consider a tracklet reliable if it occurs more than $\theta$ times within these predefined regions. During the feature calculation step of the reliable tracklet, bounding boxes outside these regions are temporarily removed.

4.4.2 Number of Clusters Choosing

For choosing the number of clusters in single-camera matching and multi-camera matching, we use the Silhouette Analysis to determine it. Silhouette analysis involves computing the silhouette coefficient for each data point in each cluster, which measures how similar that point is to other points in its cluster compared to points in neighboring clusters. The brute force approach is used to cluster the data into a range of cluster numbers to determine the optimal number of clusters. Then, the silhouette coefficient is computed for each clustering, and the number of clusters with the highest average silhouette coefficient is chosen as the optimal number of clusters. The figures 5a, and 5b show the average silhouette coefficient score in each camera, and the figures 5c and 5d show silhouette coefficient score in multi-camera in one scene.

**Choosing ReID model strategy**

In order to choose the most effective ReID model strategy, we face a challenge in determining whether a model that
performs well on synthetic validation data will generalize well to real data (Scene 001) in the test set. To address this, we propose a heuristic rule for selecting the best model based on performance on S001. Specifically, we use each ReID model to extract feature vectors and perform single-camera matching to create clusters, which are then used for multi-camera matching to obtain the final results. It is important to note that clusters from the same camera cannot be in the same group in the final results. Violation of this indicates that the model did not provide a sufficiently discriminative feature for single-camera matching, leading to the merging of different people’s tracklets into one group, and consequently negatively impacting the feature in the multi-camera matching step. Any ReID model that fails to satisfy this rule will not be used during inference.

4.5. Experiments Results

4.5.1 Re-identification

To evaluate how well ReID features perform, we only use the person id from Scene 017 and split it into a query set and a gallery set. We then determined the mean Average Precision (mAP) as the evaluation metric. The results in the table 1 demonstrate that TransReID models have taken the top two ranks for extracting features from the synthetic dataset. HRNetW48 has demonstrated the lowest mAP among all tested models. However, after applying the validation method outlined in section 4.4.2, we discovered that only TransReID models and HRNetW48 performed well on real data (S001). Consequently, we only utilized TransReID models and HRNetW48 for feature extraction and ensembling in S001. We determined that the TransReID base produced satisfactory accuracy for synthetic data and was adequate to perform optimally on the data.

<table>
<thead>
<tr>
<th>Model</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransReID + JPM + SIE</td>
<td>95.92</td>
</tr>
<tr>
<td>TransReID</td>
<td>95.76</td>
</tr>
<tr>
<td>ResNet101</td>
<td>94.43</td>
</tr>
<tr>
<td>ResNext101_ibn_a</td>
<td>94.34</td>
</tr>
<tr>
<td>ResNet50</td>
<td>94.00</td>
</tr>
<tr>
<td>HRNetW48</td>
<td>92.08</td>
</tr>
</tbody>
</table>

Table 1. The ablation study for ReID feature extraction

4.5.2 Ablation Study

The performance of individual components in our proposed system was evaluated through an ablation study. Table 2 shows the results. Compared with the baseline, the ID Switch helps us increase 0.5%. The proposed R-matching method improved the IDF1 score by 8.46%. Finally, the ensemble method, which combines features from TransReID, HRNetw48, and TransReID with JPM and SIE, resulted in the highest IDF1 score of 0.9417.

<table>
<thead>
<tr>
<th>Method</th>
<th>IDF1</th>
<th>IDP</th>
<th>IDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.8253</td>
<td>0.8568</td>
<td>0.7961</td>
</tr>
<tr>
<td>+ ID Switch</td>
<td>0.8322</td>
<td>0.8633</td>
<td>0.8032</td>
</tr>
<tr>
<td>+ R-matching</td>
<td>0.9168</td>
<td>0.9192</td>
<td>0.9145</td>
</tr>
<tr>
<td>+ Ensemble</td>
<td>0.9417</td>
<td>0.9393</td>
<td>0.9441</td>
</tr>
</tbody>
</table>

Table 2. The ablation study of combination of components

4.5.3 Comparison with other teams

Table 3 the evaluation of our proposed system in Track 1 of AI City Challenge 2023. Our system obtained an IDF1 score of 94.17%, which secured the second position among more than 25 teams worldwide.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Team ID</th>
<th>Team Name</th>
<th>IDF1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>UWPL_ETRI</td>
<td>0.9536</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>HCMIU-CVIP (ours)</td>
<td>0.9417</td>
</tr>
<tr>
<td>3</td>
<td>41</td>
<td>AILab</td>
<td>0.9331</td>
</tr>
<tr>
<td>4</td>
<td>51</td>
<td>hust432</td>
<td>0.9207</td>
</tr>
<tr>
<td>5</td>
<td>113</td>
<td>FraunhoferIOSB</td>
<td>0.9284</td>
</tr>
</tbody>
</table>

Table 3. Final results on Track 1 test set.

5. Conclusions

In this paper, a solution for Multi-Camera People Tracking in indoor scenarios is proposed for Track 1 of the AI City Challenge 2023. The proposed framework has four modules, and the introduction of ID switch detection and id switch splitting efficiently addresses the problem of tracklets with ID switches. The system performs well in matching both synthetic and real data, with the r-matching algorithm performing exceptionally well in real scenarios despite being trained on synthetic data. Experimental results on the public test set of 2023 AI City Challenge Track 1 demonstrate the efficacy of the proposed approach, achieving an IDF1 of 94.17% and securing 2nd position on the leaderboard.

6. Acknowledgement

This research is funded by Vietnam National University HoChiMinh City (VNU-HCM) under grant number DS2022-28-04. We would like to express our heartfelt appreciation to Ho Chi Minh City International University—Vietnam National University (HCMIU-VNU) for facilitating our efforts. Additionally, we would like to express our heartfelt appreciation to all of our colleagues for their contributions, which considerably aided in the revision of the manuscript.
References


[24] Chuyuan Li, Lu Li, Hongliang Jiang, Kaicheng Weng, Yiwei Geng, L. Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, Yiduo Li, Bo Zhang, Yufei Liang, Linyuan Zhou, Xiaoming Xu, Xiangxiang Chu, Xiaoming Wei, and Xiaolin Wei. Yolov6: A single-stage object detection framework for industrial applications. ArXiv, abs/2209.02976, 2022.


[51] Bo Yang and Ramakant Nevatia. Multi-target tracking by online learning of non-linear motion patterns and robust ap-


