2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

UPLIFT: Unsupervised Person Labeling and Identification via Cooperative Learning with Mobile Robots

Yu-Chee Tseng[1,3], Ting-Yuan Ke[1], and Fang-Jing Wu[2]
[1] National Yang Ming Chiao Tung University, Taiwan
[2] TU Dortmund University, Germany
[3] Miin Wu School of Computing, National Cheng Kung University, Taiwan

***Abstract*— As robots are widely used in assisting manual tasks, an interesting challenge is: Can mobile robots help create a *labeled knowledge dataset* that can be used for efficiently creating deep learning models for other sensors? This paper proposes an Unsupervised Person Labeling and Identification (UPLIFT) framework to automatically enlarge the labeled knowledge dataset. Typically, manual data labeling is very costly, especially when the user population is large and dynamic. To reduce the cost, we use a mobile robot to serve as a knowledge seed and to provide the pseudo-ground-truth for the system so that unlabeled images from other fixed surveillance cameras can be paired with the pseudo-ground-truth. Ultimately, the knowledge dataset can be generated via a system-to-system knowledge transfer process from the former to the latter and gradually expanded as the system operates longer. Experimental results in two environments indicate that UPLIFT achieves an accuracy of 94.1% on average to detect pedestrians' IDs every 10 seconds.**

I. Introduction

More and more robots have been used in data-driven artificial intelligence applications, such as intelligent surveillance [1], indoor localization [2], people identification [3], and object picking [4]. While more advanced learning algorithms are being developed for robots, a challenging question is: how can robots help establish and accumulate knowledge and data that can in return improve other data-driven learning models?

Supervised learning approaches follow the principle: *"knowledge dataset" + "supervised learning" = "intelligent model"*. However, collecting a high-quality labeled dataset is costly. Recently, to reduce the burden of labor-intensive data labeling, pseudo-label learning methods are considered in [5][6][7] to iteratively add unlabeled samples into training data by using a weak model learnt from a combination of labeled samples and pseudo-labeled samples. This work tackles the data labeling challenge by automatically creating a knowledge dataset in an unsupervised manner and further growing the dataset to keep improving the accuracy of the model. In contrast to the emerging data labeling techniques, we exploit a mobile robot to achieve this goal.

The work proposes the UPLIFT framework for autonomous data labeling in a large-scale environment. A mobile robot plays the role of *knowledge seed* to cooperatively operate with several fixed surveillance cameras for labeling pedestrians on images. The autonomous mobile robot proposed in [8] is used, which is capable of labeling the ID of a wearable device on a detected image object via pairing wearable sensing data with body skeletons on images. The autonomous labeling process is rule-based according to synchronicity between multi-modal data and thus has no manual effort. The labeled data by the robot is called "pseudo-ground-truth". The pseudo dataset may contain images at dynamic visual distances and angles and can be increased and improved as the robot moves and perceives more and more users over time. Concurrent with the mobile robot, several fixed surveillance cameras continuously capture pedestrians in the targeted regions. Their data are unlabeled and thus have no knowledge value in terms of person identification in the beginning. To create a PID model for these surveillance cameras, we can use the knowledge seed (pseudo dataset) to help label their data. Note that the labeled data from the mobile robot cannot directly be applied to model creation since the visual characteristics and view angles of the robot camera differ significantly from these fixed cameras. This work focuses on (1) how to create an initial knowledge seed that can be transferred to surveillance cameras for the PID purpose in a large-scale environment and (2) how to gradually grow the knowledge dataset over time to expand the knowledge on the surveillance system.

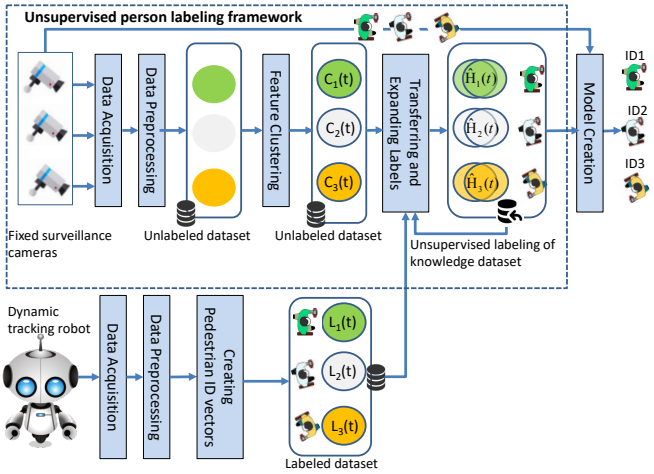


Fig. 1: An overview of the UPLIFT framework.

The demo video of UPLIFT is available here.

The key idea of the UPLIFT framework is (i) to first pair up the wearable devices and body skeletons taken by the robot, and (ii) then to pair up the features extracted in the labeled dataset (acquired by the mobile robot) with the features extracted in the unlabeled dataset (acquired by the surveillance cameras) based on the similarity among them. If proper pairing is done, the unlabeled dataset (acquired by the surveillance cameras) is tagged by the corresponding IDs of the wearable devices. As a result, the initial knowledge dataset can be subsequently expanded to a larger one by clustering new features perceived by the surveillance cameras. The proposed framework interactively establishes a cooperative learning loop between the knowledge seed (i.e., the robot) and the surveillance cameras. A prototype is implemented to verify the idea and evaluate the performance. The experimental results indicate that the proposed framework can achieve an average accuracy of 94.1% for person identification every 10 seconds. The key contributions of the work are as follows. First, compared to the existing efforts on developing advanced learning algorithms, we develop a new data labeling method. Second, to the best of our knowledge, this is the first work using a dynamic robot to transfer and expand knowledge from one system to another via a cooperative learning process.

II. Related Work

Biometrics technology has been exploited for PID through features such as iris patterns [9] and fingerprints [10]. This relies on a large labeled dataset to create a CNN model [11]. However, such technologies usually operate well only at short perception ranges and are sensitive to view angles and lighting intensity. Also, manually labeling data with human intervention is usually required.

Sensor fusion technology uses multi-modal data to identify a person. Data from an RGB-D depth camera and wearable inertial sensors are fused in [12] to recognize people and their activities. The work in [10] identifies a person based on fusing data from an RGB-D depth camera and RFID antennas. Fusing IoT and video data is discussed in [13]. Fusing UAV video data for PID has been addressed in [3], [14]. The device-object pairing problem is addressed in [15].

Data labeling technology is an important key to data-driven applications. A high-quality dataset must be unbiased and contains sufficient features for model creation. The activity learning in [16] iteratively chooses better unlabeled data for being labeled by a human annotator. Unsupervised clustering techniques are considered in [17] to discover new categories in an unlabeled dataset by partitioning data into different groups based on their similarity measures. To reduce labeling efforts, [18] tags a single label to a group of unlabeled data. The work [19] labels the objects with significant regularity in the features at earlier iterations. The work [20] jointly performs clustering and labeling. Data labeling methods for person re-identification (re-ID) are designed in [21][22][23]. When a model is used on a testing dataset which is different from the labeled training dataset, the performance drop due to the cross-dataset person re-ID is addressed in [21]. In [22], similar image samples of people are clustered and assigned the same initialized pseudo label to update the person re-ID model. Clustering methods are designed in [23] to merge image frames of people according to their closeness and spatial structure in their neighborhoods for re-identifying people.

III. Unsupervised Person Labeling Framework

A. Problem and Methodology Overview

We consider an environment with a set of static surveillance cameras and a dynamic mobile robot. These static surveillance cameras are mounted at fixed locations and may cover most of the environment [24], [25]. The views of these static surveillance cameras may or may not overlap. In contrast, the robot is capable of moving around but has a narrower view of the environment as compared to those surveillance cameras. Persons in the environment may carry ID badges, wearable devices, or mobile phones, which can reveal their identities that can be recognized by the robot. Whenever an identity is recognized, the person on the image frames taken by the robot are labeled by his/her ID. However, static surveillance cameras are unable to label persons on their image frames due to unavailability of IDs. On the other hand, the quality of labeled images by the robot depends on the robot's view angles. Such labeled image data may not be sufficient to create a good neural network for the surveillance cameras to perform PID. In particular, a model created from cross-camera data may lead to poor performance. This motivates us to design UPLIFT to transfer knowledge seed from the robot towards the set of static surveillance cameras.

Fig. 1 is an overview of UPLIFT. Given a time interval t, the detected persons on image frames captured by the static surveillance cameras during t are represented as a set of bounding boxes, denoted by $B(t)$. The bounding boxes are unlabeled initially. Let $Z(t)$ denote the set of pedestrians' bounding boxes that can be successfully assigned IDs by the robot during t. The ID-assigning work can be autonomously done by the robot via pairing visual data and ID-embedded devices carried by the persons. $Z(t)$ is further transformed into several vectors, denoted by $L_1(t), L_2(t), \ldots, L_m(t)$, where m is the number of labeled persons and each $L_i(t)$ is a representative feature vector of color characteristics of person i, $i \leq m$. Similarly, the dataset $B(t)$ is preprocessed in the same way to extract the feature vectors of color characteristics. Since $B(t)$ is unlabeled, we must first cluster these vectors and then extract these clusters' representative vectors, denoted by $C_1(t), C_2(t), \ldots, C_n(t)$, where n is the number of clusters. Here, each $C_i(t)$ is considered the representative feature vector of cluster i. The goal is to establish the two abilities.

- *Knowledge transfer*: To pair up the labeled vectors (i.e., $L_1(t)$, $L_2(t)$, ..., $L_m(t)$) and the unlabeled vectors (i.e., $C_1(t)$, $C_2(t)$, ..., $C_n(t)$), meaning transferring knowledge from the robot to those static surveillance cameras.
- *Knowledge expansion*: To grow up the size of the knowledge dataset for the surveillance system over time.

B. Data Acquisition

1) Unlabeled Dataset $B(t)$ Acquisition: This sub-module is designed for the fixed surveillance cameras to extract unlabeled bounding boxes from image frames during time interval t. The real-time object detection framework in [26] is modified to a single-class model for pedestrians detection. Only full-body images of persons are included to ensure visual and dataset quality. During t, this sub-module extracts a set of bounding boxes $B(t) = \{b_1(t), b_2(t), \ldots\}$, where each $b_i(t) = \{x_i(t), y_i(t), w_i(t), h_i(t), s_i(t)\}$ includes the center coordinate, width, height, and confidence score of the box. We denote the maximum number of detected persons during t by $N_{max}(t)$.

2) Labeled Dataset $Z(t)$ Acquisition: This sub-module is designed for the robot to collect bounding boxes of pedestrians with their IDs. We adopt the work in [8] for person identification through fusing inertial data from wearable devices on pedestrians and skeleton data from the RGB-D camera on the robot. (We remark that the approach can be easily extended to wearable ID badges and smart watches/phones.) The robot meets pedestrians occasionally and thus captures them dynamically from different view angles. Each wearable device has a built-in inertial sensor for movement sensing and a pre-registered unique ID. During t, the robot collects the set $Z(t) = \{z_1(t), z_2(t), \ldots, z_m(t)\}$, where $z_i(t)$ is the bounding box of a human object that the robot recognizes and the box should cover the person's whole body for quality ensuring. This can be done by sending a detected bounding box to a pose model, such as OpenPose, for skeleton analysis. According to [8], once a person's body motion data matches with its inertial sensing data, its bounding box is automatically labeled by the ID of the wearable device. The labeling completely requires no human intervention. It is called *pseudo-ground-truth* because the bounding boxes are taken by a robot camera rather than surveillance cameras.

C. Data Preprocessing

This module processes the two sets, $B(t)$ and $Z(t)$, to extract visual features. Fig. 2 shows its workflow.

1) Non-ideal Image Filter: This sub-module filters out non-ideal bounding boxes. This facilitates subsequent extraction of visual features.

- *Remove highly overlapped images*: The bounding box occlusion problem is solved by calculating *Intersection over Union (IoU)*. If the IoU between any two bounding boxes is higher than a predefined threshold of δ_{IoU}, the two bounding boxes are removed from the dataset.
- *Remove misaligned images*: The bounding box misalignment problem appears when a large portion of a person is outside its bounding box due to poor human detection or a person staying too close to the robot. Therefore, a bounding box is removed from the dataset if its width exceeds a threshold δ_w or its height exceeds a threshold δ_h.
- *Resize images with inconsistent resolution*: To ensure consistency in resolution, all the bounding boxes in the dataset after the above filtering are resized to a fixed resolution of $w \times h$. The final datasets are denoted by $B'(t)$ and $Z'(t)$.

2) Appearance-based Feature Extraction: This sub-module represents each bounding box in $B'(t)$ and $Z'(t)$ by its color features. The key idea is to encode an image by its color combination. Specifically, each bounding box is transformed into a vector of an RGB color histogram. First, the model in [27] is applied to remove clutter backgrounds. Based on the Golden Ratio Principle [28], each box is further proportionally partitioned into three portions: head, torso, and leg at 14.58%, 23.61%, and 61.81% of the total height. Since the head portion contains little discriminative information on dress colors, it is excluded from the following feature extraction. Then, the remaining bounding box is represented by two RGB color histograms, one for the torso part and one for the leg part. Specifically, each pixel in the image is mapped to a triplet of $(r_{value}, g_{value}, b_{value})$, where each value is encoded by 8 bits. This quantization maps each RGB pixel to a $8 \times 8 \times 8$ space. The histogram result can be regarded as a 512-dimensional vector representing the color distribution of an image. Combining torso and leg portions leads to a 1024-dimensional vector. Finally, to reduce the computational cost, the principal component analysis (PCA) is applied to reduce the 1024-dimensional vector to a 256-dimensional vector. To summarize, the dataset $B'(t)$ is transformed to $V(t) = \{v_1(t), v_2(t), ...\}$, where $v_i(t)$ is the 256-dimensional feature vector of each bounding box $b'_i(t) \in B'(t)$. Similarly, the dataset $Z'(t)$ is transformed to $U(t) = \{u_1(t), u_2(t), \ldots, \}$, where $u_i(t)$ is the 256-dimensional feature vector of each bounding box $z'_i(t) \in Z'(t)$.

D. Creating Pedestrian ID Vectors

Since the vectors in $U(t)$ are associated with pedestrians' IDs labeled by the robot, the main task of this sub-module is to create a representative vector for each pedestrian. The vectors in $U(t)$ are partitioned into m classes according to their IDs given by the robot. Let $\Phi_i(t) \subseteq U(t)$ denote the subset of vectors with ID = i. The representative vector for pedestrian i is $L_i(t) = \frac{1}{|\Phi_i(t)|} \sum_{\forall u_j(t) \in \Phi_i(t)} u_j(t)$. Here, $L_i(t)$ is the centroid of all the vectors in $\Phi_i(t)$.

E. Feature Clustering

Since vectors in $V(t)$ are not labeled, this module shall partition them into p clusters based on inter-vector distances. The ideal value of p should be the actual number of pedestrians in the environment. However, its actual value is unknown. (For example, if $p = 3$, during t only two bounding boxes may be detected per frame due to visual blocking, while during $t+1$ four bounding boxes may be detected per frame.) The key challenge for this module is to approximate the actual number of clusters to p, when clustering vectors in $V(t)$. Initially, $N_{max}(t)$, which is the maximum number of detected bounding boxes over all image frames during t, can be an estimation of p. Let $\Omega_1(t), \Omega_2(t), \ldots, \Omega_n(t)$ denote the final clustering outcome, where n is the number of clusters.

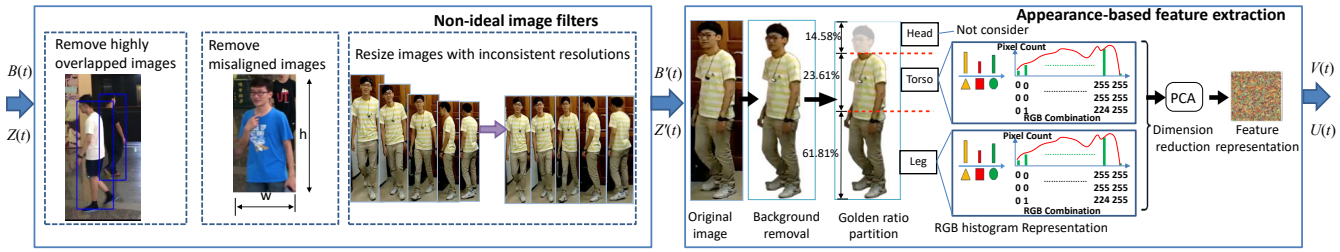


Fig. 2: The workflow of data preprocessing.

Two feature clustering algorithms are proposed. The first algorithm groups vectors by approximating the number of clusters based on intra-cluster variance and inter-cluster variance. The second algorithm groups vectors by approximating the radius of clusters based on density distribution in the 256-dimensional vector space.

1) Variance-based Feature Clustering: This algorithm considers the *Calinski-Harabasz (CH) index* [29] to approximate the number of clusters. Given k clusters, denoted by $\hat{\Omega}_1(t)$, $\hat{\Omega}_2(t)$, ..., $\hat{\Omega}_k(t)$, the CH index is calculated by $CH = \frac{SS_e}{SS_a} \times \frac{(|V(t)|-k)}{(k-1)}$, where $|V(t)|$ the total number of vectors (i.e., data points) in $V(t)$, SS_e is the overall inter-cluster variance, and SS_a is the overall intra-cluster variance. Notations $\hat{\Omega}_1(t), \hat{\Omega}_2(t), \ldots, \hat{\Omega}_k(t)$ are to differentiate from the final clustering outcome $\Omega_1(t)$, $\Omega_2(t)$, ..., $\Omega_n(t)$. SS_e measures the variance of all cluster centroids from the $V(t)$'s grand centroid, i.e., $SS_e = \sum_{i=1}^{k} |\hat{\Omega}_i(t)| \times ||\hat{C}_i(t) - \overline{\Theta}||^2$, where $|\hat{\Omega}_i(t)|$ denotes the number of vectors (i.e., data points) in the cluster $\hat{\Omega}_i(t)$, $\hat{C}_i(t)$ is the centroid of cluster $\hat{\Omega}_i(t)$, and $\overline{\Theta}$ is the grand centroid of $V(t)$. SS_a measures the variance of all the vectors from their own centroid $\hat{C}_i(t)$, i.e., $SS_a = \sum_{i=1}^{k} \sum_{v_j(t) \in \hat{\Omega}_i(t)} ||v_j(t) - \hat{C}_i(t)||^2$. A higher CH implies a better clustering result as we prefer a larger inter-cluster variance and a smaller intra-cluster variance. Therefore, this algorithm examines potential values of k between $[N_{max}(t), N_{max}(t) + \alpha]$ and chooses the k leading to the largest CH. Here, α defines a range to search for the largest CH. CH_{max} is to iteratively keep the maximum value of CH. To approximate the actual p, the algorithm computes the values of CH index for all k in $[N_{max}(t), N_{max}(t) + \alpha]$. For each k, the k-means clustering algorithm is applied. If a better value is found, CH_{max}, n, and the clustering outcome are updated. Finally, for the n clusters, we compute each of their representative vectors $C_1(t), C_2(t), \ldots, C_n(t)$, which are their centroids, respectively.

2) Density-based Feature Clustering: The key idea of this algorithm is to approximate the average radius of clusters that further limits the minimum of data points in a cluster for forming clusters based on the density. Given k clusters, $\hat{\Omega}_1(t), \hat{\Omega}_2(t), \ldots, \hat{\Omega}_k(t)$, the average radius of these k clusters is computed by $\varepsilon_k = \frac{1}{k} \sum_{i=1}^{k} (\frac{1}{|\hat{\Omega}_i(t)|} \sum_{v_j(t) \in \hat{\Omega}_i(t)} ||v_j(t) - \hat{C}_i(t)||^2)$. Given a value of ε_k, the average number of vectors (i.e., data points) in the ε_k-neighborhood of a cluster is defined by $\mu_k = \frac{1}{k} \sum_{i=1}^{k} R(\varepsilon_k, \hat{C}_i(t))$, where $R(\varepsilon_k, \hat{C}_i(t))$ is the number of vectors (i.e. data points) in the ε_k-neighborhood of the cluster $\hat{\Omega}_i(t)$. If $d(v_j(t), \hat{C}_i(t)) \leq \varepsilon_k$, $v_j(t)$ is in the ε_k-neighborhood of $\hat{\Omega}_i(t)$, where $d(v_j(t), \hat{C}_i(t))$ is the distance between $v_j(t)$ and $\hat{C}_i(t)$.

The algorithm is designed to approximate the number of clusters. First, we set $k = N_{max}(t)$ and apply the k-means clustering algorithm on $V(t)$ to get k clusters. Based on these k clusters, the values of ε_k and μ_k can be computed. Then, the two parameters are considered as the input of the DBSCAN algorithm [30] to re-cluster vectors in $V(t)$. If the number of clusters after the DBSCAN is more than $N_{max}(t)$, we sort these clusters according to the numbers of data points in them in descending order and only keep the first $N_{max}(t)$ clusters as the clustering outcome. We denote $\Omega_1(t), \Omega_2(t), \ldots, \Omega_n(t)$ the final clustering outcome of the algorithm, where $n \leq N_{max}(t)$. Finally, from these n clusters, their representative vectors, $C_1(t), C_2(t), \ldots, C_n(t)$, are computed by their centroids.

F. Transferring and Expanding Labels

This module generates the knowledge dataset for the surveillance cameras by matching the unlabeled representative vectors $C_1(t), C_2(t), \ldots, C_n(t)$ with the labeled vectors $L_1(t)$, $L_2(t)$, ..., $L_m(t)$. Since it is possible that $n \neq m$, the pairing results may assign labels to only some of the clusters $\Omega_1(t)$, $\Omega_2(t)$, ..., $\Omega_m(t)$. (Note that it is not necessary to assign each cluster a label.) We shall even gradually expand our existing knowledge dataset by merging the newly labeled datasets with the existing datasets. Two sub-modules are designed to achieve this.

1) Knowledge Transfer via Pairing: Given unlabeled $C_1(t)$, $C_2(t)$, ..., $C_n(t)$ and labeled $L_1(t), L_2(t), \ldots, L_m(t)$, this module assigns the pedestrian ID associated with $L_j(t)$ to $C_i(t)$ if we find high *similarity* between $C_i(t)$ and $L_j(t)$. Since the $C_1(t), C_2(t), \ldots, C_n(t)$ and $L_1(t)$, $L_2(t)$, ..., $L_m(t)$ are created using images from different cameras, their visual characteristics may be slightly different. We use color distribution as the main feature and apply cosine similarity to measure a similarity score: $Sim(C_i(t), L_j(t)) = \frac{C_i(t) \cdot L_j(t)}{\|C_i(t)\| \|L_j(t)\|}$. Here, $C_i(t)$ and $L_j(t)$ are vectors in a 256-dimensional space, and the score should fall in $[-1, 1]$. The following pairing process is designed.

1) Compute all-pair similarity scores between $C_i(t)$ and $L_j(t)$, $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$.
2) Sort all pairs based on their computed similarity scores in descending order.

3) Choose the pair from with the maximal similarity score, denoted by $(C_i(t), L_j(t))$. Assign the ID associated with $L_j(t)$ to $C_i(t)$ and all vectors in the corresponding cluster $\Omega_i(t)$.
4) Remove those pairs containing $C_i(t)$ from the sorted list.
5) Repeat Step (3) until each $L_j(t)$ has been paired.

Ideally, after the pairing process, if $m = n$, all $\Omega_i(t)$ can be labeled. However, since the robot has only partial views of the environment, it sometimes cannot capture all the pedestrians in the environment during time interval t. In contrast to the robot, surveillance cameras are more likely to capture all the pedestrians in the environment. So, it is more likely that $n \geq m$. In this case, some $C_i(t)$ may not be paired and labeled.

2) Knowledge Expansion via Merging and Sampling: To address the above unpaired issue, a backtracking process is designed to integrate historical vectors during time interval $(t-1)$ and the current vectors during interval t. The results will be iteratively forwarded to interval $(t+1)$. Such a labeling loop also opens an opportunity for expanding the knowledge dataset. To achieve this, we need to keep a set of historical vectors $H(t-1) = \{h_1(t-1), h_2(t-1), \ldots, h_{p'}(t-1)\}$ and a superset $\{\hat{H}_1(t-1), \hat{H}_2(t-1), \ldots, \hat{H}_{p'}(t-1)\}$ at the end of the time interval $(t-1)$, where p' is the number of pedestrians who are already labeled during $(t-1)$, $h_i(t-1)$ is the weighted feature vector for the i-th pedestrian by taking all historical feature vectors up to time interval $(t-1)$ into account, and $\hat{H}_i(t-1)$ is the expanded set of images of the pedestrian corresponding to $h_i(t-1)$. For each paired tuple found in the pairing process during t, we denote as $Pair(C_i(t), L_j(t)) = 1$, where $Pair(\cdot) = 1$ indicates a successful pairing. Otherwise, $Pair(\cdot) = 0$. Let $\Psi(t-1) = \{ID(h_1(t-1)), ID(h_2(t-1)), \ldots, ID(h_{p'}(t-1))\}$ be the set of identities of the p' pedestrians which already have labeled image datasets during $(t-1)$, where $ID(\cdot)$ returns the identity of the pedestrian for a given feature vector.

Fig. 3 shows all cases of the iterative pairing process. For each paired $Pair(C_i(t), L_j(t))$=1, there are five cases to update the historical vectors and the superset at $(t-1)$.

• *Case 1*: $ID(L_j(t)) = ID(h_k(t-1))$ for some k: This indicates that the pedestrian was already labeled during $(t-1)$ by $ID(h_k(t-1))$. Thus, we set

$$h_k(t) = \beta \cdot h_k(t-1) + \gamma \cdot L_j(t) + (1-\beta-\gamma) C_i(t), \quad (1)$$

where β and γ are two weighting factors in $[0, 1]$. Intuitively, $h_k(t)$ is a weighted vector of $h_k(t-1)$, $L_j(t)$, and $C_i(t)$. The expanded set of images of the pedestrian corresponding $h_k(t)$ is set to

$$\hat{H}_k(t) = sub(\hat{H}_k(t-1)) \bigcup sub(\Phi_j(t)) \bigcup \Omega_i(t), \quad (2)$$

where $sub(Z)$ means a subsample of the set Z by randomly keeping a certain ratio of elements of Z in $sub(Z)$ (i.e., $sub(Z) \subseteq Z$). That is, we only include a portion of historical images (from historical labeled images) of $\hat{H}_k(t-1)$ in $\hat{H}_k(t)$. Similarly, we also include a portion of images (from the robot's labeled images) of $\Phi_j(t)$ into $\hat{H}_k(t)$. So, this process includes new labeled images and gradually removes aged labeled images of persons who have not been captured for a long time.

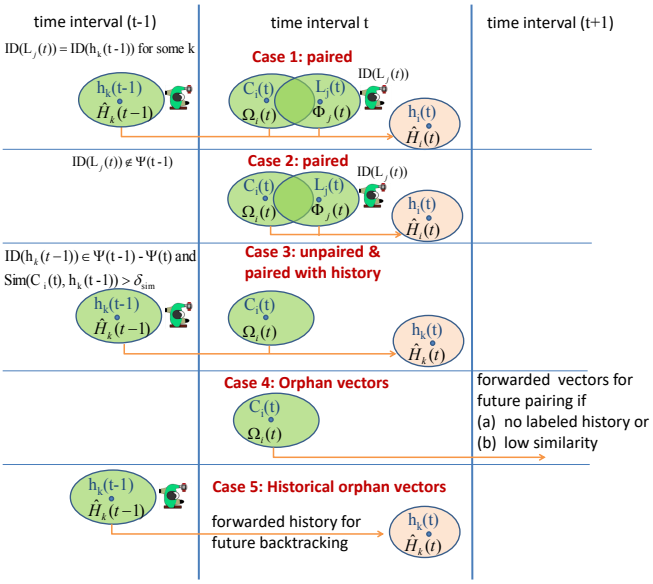


Fig. 3: Feature vector backtracking and forwarding.

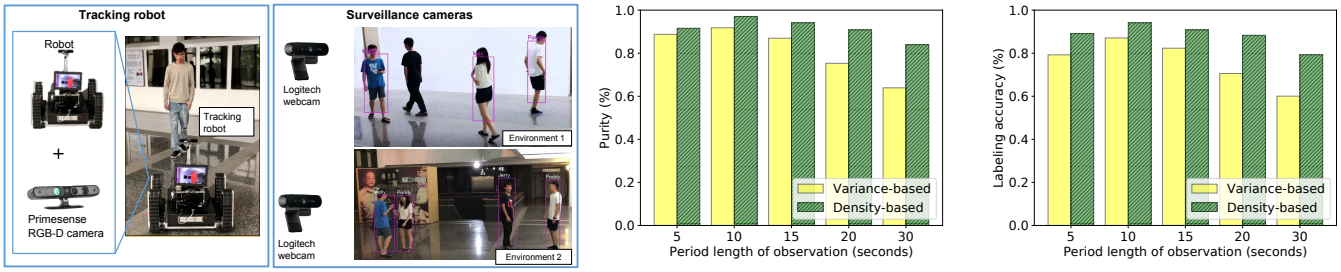
• *Case 2*: $ID(L_j) \notin \Psi(t-1)$: This indicates that the pedestrian was not captured by the robot before $(t-1)$ but is captured and paired during t. This is a new pedestrian coming to the system. For a new pedestrian, we generate a new historical vector $h_k(t)$ for the new pedestrian by Eq. (1), where the $h_k(t-1)$ is degenerated in the equation. Meanwhile, the knowledge dataset $\hat{H}_k(t)$ is updated by Eq. (2), where the $sub(\hat{H}_k(t-1))$ is degenerated.

For each $C_i(t)$ such that $Pair(C_i(t), L_j(t))$=0 for all j, the unpaired cluster $C_i(t)$ is processed by the cases below.

• *Case 3*: There exists an $ID(h_k(t-1)) \in \Psi(t-1) - \Psi(t)$ and $Sim(C_i(t), h_k(t-1)) > \delta_{sim}$ for some k: This indicates that the pedestrian with identity $ID(h_k(t-1))$ was captured by the robot during $(t-1)$ but is not captured by the robot during t due to a partial view. In this case, if $Sim(C_i(t), h_k(t-1))$ is greater than a threshold δ_{sim}, we merge the historical $(h_k(t-1), \hat{H}_k(t-1))$ and the current $(C_i(t), \Omega_i(t))$ together and assign $ID(h_k(t-1))$ to the vectors in $\Omega_i(t)$. Thus, we set the historical vector for the pedestrian by Eq. (1), where the $L_j(t)$ is degenerated in the equation. Meanwhile, the expanded knowledge dataset is updated by Eq. (2), where the $sub(\Phi_j(t))$ is degenerated. If the similarity is not high enough, the vectors in $\Omega_i(t)$ are called *orphan vectors*, which will be processed in Case 4.

After processing the above Cases 1-3, there may be some orphan $(C_i(t), \Omega_i(t))$ which is still not labeled, and some orphan $(h_k(t-1), \hat{H}_k(t-1))$ such that $ID(h_k(t-1))$ is not included in $\Psi(t)$ yet. The cases below address them.

• *Case 4*: Orphan $(C_i(t), \Omega_i(t))$: The orphan vectors in $\Omega_i(t)$ are forwarded for future pairing in the next time interval $(t+1)$. We set $V(t+1) = V(t+1) \bigcup \Omega_i(t)$.



(a) (b) (c)

Fig. 4: Experimental results: (a) the prototype, (b) performance of feature clustering, and (c) performance of knowledge transfer via pairing.

• *Case 5*: Orphan $(h_k(t-1), \hat{H}_k(t-1))$: If there exists a historical vector $h_k(t-1)$ which cannot be processed by Cases 1-3, we will forward it to interval t by setting $h_k(t) = h_k(t-1)$ and $\hat{H}_k(t) = \hat{H}_k(t-1)$.

IV. Performance Evaluation

A prototype is implemented with the hardware components shown in Fig. 4a. Two Logitech web cameras, which have a higher resolution, are mounted at fixed positions. Each surveillance camera initially collects image frames for 60 seconds at a sampling rate of 30 FPS. The robot is equipped with a PrimeSense RGB-D camera, which has a lower resolution and collects image frames at the same sampling rate. Experiments are conducted in two environments, one with a better light condition and one with a poorer one. Eight persons are considered. The number of extracted bounding boxes is 4988, and only 4294 of which are selected for feature clustering. The robot in [8] is implemented to collect the pseudo-ground-truth with an accuracy of 88%. The number of detected bounding boxes by the robot is 214, which is significantly fewer than that by the surveillance cameras. The other settings are $\delta_{IoU} = 0.15$, $\delta_w = 480$, $\delta_h = 1200$, $w = 200$, and $h = 500$ are for the data preprocessing; $\alpha = 2$ is used for feature clustering; $\beta = 0.45$ and $\gamma = 0$ are for transferring and expanding labels.

A. Performance of Feature Clustering

Since the proposed framework clusters vectors obtained from surveillance cameras before assigning labels to them, we first study if these vectors can be correctly grouped to the right classes. To evaluate the performance of clustering, we define *purity* to be the percentage of vectors that belong to the same cluster. Let $\omega_j^i(t)$ denote the set of the j-th pedestrian's vectors clustered into $\Omega_i(t)$ during time interval t. We define $Purity = \frac{1}{\sum_{\forall t}\sum_{i=1}^{n}|\Omega_i(t)|}\sum_{\forall t}\sum_{i=1}^{n}\max\{|\omega_j^i(t)|, 1 \leq j \leq p\}$, where n is the number of clusters, and p is the actual number of pedestrians. Since a cluster may contain multiple pedestrians' vectors, the max function is to pick the number of vectors in the majority for each cluster. A higher value of purity indicates a better performance of feature clustering. The observation period is varied from 5, 10, 15, 20, to 30 seconds. When a longer period is considered, more image frames can be captured, but it also incurs much more people's movements that increasingly influence the performance of feature clustering. Fig. 4b shows the experimental results, where the best choice of period length is 10 seconds. With a period length of 10 seconds, the density-based feature clustering algorithm achieves a purity level of 97.1%. The level of purity resulting from the density-based feature clustering slightly decreases as the period length increases. This is because a longer period does not affect the data distribution in the dataset but may affect the number of clusters due to noises arising from changes in light characteristics caused by human movements.

B. Performance of Knowledge Transfer

Next, the accuracy of pairing for knowledge transfer presented in Section III-F is evaluated. Each given label to a cluster after pairing is manually examined with the actual label to a pedestrian. The accuracy is the percentage of labels that are correctly assigned by our algorithm. Similarly, the observation period is changed from 5, 10, 15, 20, to 30 seconds. As shown in Fig. 4c, the accuracy can achieve 94.1% when the density-based feature clustering is applied with an observation period of 10 seconds. When the variance-based feature clustering is applied, the accuracy is 87.1%.

V. Conclusions

The work proposes the UPLIFT framework to transfer labeling ability from one camera system to another camera system. However, how often the robot meets pedestrians affects the efficiency of data labeling. Future work may consider a multi-robot collaborative framework for improving data labeling, especially for an extremely large environment. Furthermore, cross-system data matching could be considered in the future by incorporating not only the color profiles of objects but also their trajectories on frames.

VI. Acknowledgment

Y.-C. Tseng's research is co-sponsored by ITRI and NSTC, Taiwan. This work is also financially supported by "Center for Open Intelligent Connectivity" of "Higher Education Sprout Project" of NYCU and MOE, Taiwan.

REFERENCES

[1] X. Wang, "Intelligent multi-camera video surveillance: A review," in *Pattern Recognition Letters*, vol. 34, no. 1, 2013, pp. 3–19.

[2] T.-H. Chiang, Z.-H. Sun, H.-R. Shiu, K. C.-J. Lin, and Y.-C. Tseng, "Magnetic field-based localization in factories using neural network with robotic sampling," *IEEE Sensors Journal*, vol. 20, no. 21, pp. 13 110–13 118, 2020.

[3] A. Mukashev, L.-D. Van, S. Sharma, M. F. Tandia, and Y.-C. Tseng, "Person tracking by fusing posture data from uav video and wearable sensors," *IEEE Sensors Journal*, vol. 22, no. 24, pp. 24 150–24 160, 2022.

[4] C. Eppner, S. Höfer, R. Jonschkowski, R. Martín-Martín, A. Sieverling, V. Wall, and O. Brock, "Lessons from the amazon picking challenge: Four aspects of building robotic systems." in *Robotics: science and systems*, 2016.

[5] P. Cascante-Bonilla, F. Tan, Y. Qi, and V. Ordonez, "Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning," in *AAAI Conference on Artificial Intelligence*, 2021, pp. 6912–6920.

[6] P. Yan, Z. Wu, M. Liu, K. Zeng, L. Lin, and G. Li, "Unsupervised domain adaptive salient object detection through uncertainty-aware pseudo-label learning," in *AAAI Conference on Artificial Intelligence*, 2022, pp. 3000–3008.

[7] S. Gao, W. Zhang, Y. Wang, Q. Guo, C. Zhang, Y. He, and W. Zhang, "Weakly-supervised salient object detection using point supervison," in *AAAI Conference on Artificial Intelligence*, 2022.

[8] R. Tsai, H. Ke, K. Lin, and Y.-C. Tseng, "Enabling identity-aware tracking via fusion of visual and inertial features," in *International Conference on Robotics and Automation*, 2019, pp. 2260–2266.

[9] S. Minaee, A. Abdolrashidiy, and Y. Wang, "An experimental study of deep convolutional features for iris recognition," in *IEEE Signal Processing in Medicine and Biology Symposium*, 2016, pp. 1–6.

[10] F. Cafaro, A. Panella, L. Lyons, J. Roberts, and J. Radinsky, "I see you there!: Developing identity-preserving embodied interaction for museum exhibits," in *The SIGCHI Conference on Human Factors in Computing Systems*, 2013, pp. 1911–1920.

[11] O. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proceedings of the British Machine Vision Conference*, 2015, pp. 41.1–41.12.

[12] W.-C. Chang, C.-W. Wu, R. Tsai, K. Lin, and Y.-C. Tseng, "Eye on you: Fusing gesture data from depth camera and inertial sensors for person identification," in *IEEE International Conference on Robotics and Automation*, 2018, pp. 2021–2026.

[13] L.-Y. Zhang, H.-C. Lin, K.-R. Wu, Y.-B. Lin, and Y.-C. Tseng, "Fusiontalk: An iot-based reconfigurable object identification system," *IEEE Internet of Things Journal*, vol. 8, no. 9, pp. 7333–7345, 2021.

[14] L. Van, L. Zhang, C. Chang, K. Tong, K. Wu, and Y. Tseng, "Things in the air: tagging wearable iot information on drone videos," *Discov. Internet Things*, vol. 1, no. 1, 2021. [Online]. Available: https://doi.org/10.1007/s43926-021-00005-8

[15] K. Tong, K. Wu, and Y. Tseng, "The device-object pairing problem: Matching iot devices with video objects in a multi-camera environment," *Sensors*, vol. 21, no. 16, p. 5518, 2021. [Online]. Available: https://doi.org/10.3390/s21165518

[16] B. Settles, "Active learning: Synthesis lectures on artificial intelligence and machine learning," in *Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publisher*, 2012.

[17] Y. Lee and K. Grauman, "Object-graphs for context-aware category discovery," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1–8.

[18] D. Liu and T. Chen, "Unsupervised image categorization and object localization using topic models and correspondences between images," in *IEEE International Conference on Computer Vision*, 2007, pp. 1–7.

[19] Y. Lee and K. Grauman, "Learning the easy things first: Self-paced visual category discovery," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1721–1728.

[20] S. Hong, J. Choi, J. Feyereisl, B. Han, and L. Davis, "Joint image clustering and labeling by matrix factorization," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 7, 2016, pp. 1411–1424.

[21] F. Dubourvieux, R. Audigier, A. Loesch, S. Ainouz, and S. Canu, "Unsupervised domain adaptation for person re-identification through source-guided pseudo-labeling," in *International Conference on Pattern Recognition (ICPR)*, 2021, pp. 4957–4964.

[22] K. Zeng, M. Ning, Y. Wang, and Y. Guo, "Hierarchical clustering with hard-batch triplet loss for person re-identification," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 13 654–13 662.

[23] G. Wu, X. Zhu, and S. Gong, "Tracklet self-supervised learning for unsupervised person re-identification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 12 362–12 369.

[24] C.-F. Huang, Y.-C. Tseng, and L.-C. Lo, "The coverage problem in three-dimensional wireless sensor networks," *Journal of Interconnection Networks*, vol. 08, no. 03, pp. 209–227, 2007.

[25] Y. Wang, Y. Chen, and Y. Tseng, "Using rotatable and directional (r&d) sensors to achieve temporal coverage of objects and its surveillance application," *IEEE Trans. Mob. Comput.*, vol. 11, no. 8, pp. 1358–1371, 2012. [Online]. Available: https://doi.org/10.1109/TMC.2011.161

[26] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," in *arXiv:1804.02767*, 2018, https://arxiv.org/abs/1804.02767.

[27] N. Jojic, A. Perina, M. Cristani, V. Murino, and B. Frey, "Stel ccomponent analysis: Modeling spatial correlations in image class structure," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2044–2051.

[28] A. Nanda, P. Sa, S. Choudhury, S. Bakshi, and B. Majhi, "A neuromorphic person re-identification framework for video surveillance," in *IEEE Access*, vol. 5, 2017, pp. 6471–6482.

[29] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," in *Communications in Statistics*, vol. 3, no. 1, 1974, pp. 1–27.

[30] P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, *Introduction to Data Mining*. Pearson, 2019.