Identity-aware attribute recognition via real-time distributed inference in mobile edge clouds

Identity-Aware Attribute Recognition via Real-Time Distributed Inference in Mobile Edge Clouds

ZICHUAN XU, The Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, School of Software, Dalian University of Technology

JIANGKAI WU, School of Software, Dalian University of Technology

QIUFEN XIA*, The Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, International School of Information Science and Engineering, Dalian University of Technology, Dalian, China

PAN ZHOU, The Hubei Engineering Research Center on Big Data Security, School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan, China

JIANKANG REN, School of Computer Science, Dalian University of Technology

HUIZHI LIANG, Department of Computer Science, University of Reading

With the development of deep learning technologies, attribute recognition and person re-identification (re-ID) have attracted extensive attention and achieved continuous improvement via executing computing-intensive deep neural networks in cloud datacenters. However, the datacenter deployment cannot meet the real-time requirement of attribute recognition and person re-ID, due to the prohibitive delay of backhaul networks and large data transmissions from cameras to datacenters. A feasible solution thus is to employ mobile edge clouds (MEC) within the proximity of cameras and enable distributed inference.

In this paper, we design novel models for pedestrian attribute recognition with re-ID in an MEC-enabled camera monitoring system. We also investigate the problem of distributed inference in the MEC-enabled camera network. To this end, we first propose a novel inference framework with a set of distributed modules, by jointly considering the attribute recognition and person re-ID. We then devise a learning-based algorithm for the distributions of the modules of the proposed distributed inference framework, considering the dynamic MEC-enabled camera network with uncertainties. We finally evaluate the performance of the proposed algorithm by both simulations with real datasets and system implementation in a real testbed. Evaluation results show that the performance of the proposed algorithm with distributed inference framework is promising, by reaching the accuracies of attribute recognition and person identification up to 92.9% and 96.6% respectively, and significantly reducing the inference delay by at least 40.6% compared with existing methods.

CCS Concepts: • **Computing methodologies → Distributed computing methodologies**; • **Networks → Data path algorithms**.

Additional Key Words and Phrases: Attribute Recognition, Distributed Inference, Online Learning, Mobile Edge Computing

*Corresponding author. This work is partially supported by the National Natural Science Foundation of China (Grant No. 61802048 and 61802047), the fundamental research funds for the central universities in China (Grant No. DUT19RC(4)035), DUT-RU Co-Research Center of Advanced ICT for Active Life, and the "Xinghai Scholar Program" in Dalian University of Technology, China.

Manuscript submitted to ACM

ACM Reference Format:
Zichuan Xu, Jiangkai Wu, Qiufen Xia, Pan Zhou, Jiankang Ren, and Huizhi Liang. 2020. Identity-Aware Attribute Recognition via Real-Time Distributed Inference in Mobile Edge Clouds. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394171.3414048

1 INTRODUCTION

Person re-identification (re-ID) and attribute recognition both are employed in critical applications of surveillance. Person re-ID is to identify pedestrians according to views from different video surveillance or spans a different time using a single camera [1], the main concern is whether the two pedestrian images taken by different cameras are the same person. While attribute recognition is to distinguish the presence of a set of attributes from an image, such as hairstyle, wearing, carrying, gender, age, etc. Person re-ID and attribute recognition have a common target to learn pedestrian features, and both rely on deep learning techniques. So combining the two techniques together makes sense for establishing an intelligent multi-camera monitoring system. Accuracy and response time are crucial factors in intelligent multi-camera monitoring systems.

Deep learning models are complex and resource-consuming, the resource-limited devices on the side of users are hard to compute their inference results. These deep learning services thus are mostly deployed in cloud datacenters with abundant GPU resources. However, all cameras continuously transmit their surveillance data to datacenters for processing. Such centralized architecture based on remote datacenters cannot satisfy the delay requirements of real-time analytics based on deep learning models, due to the congested links and severe transmission latency in cloud backhaul networks. With the development of 5G technique, mobile edge computing (MEC) consisting of multiple computing nodes emerges as an attractive and promising paradigm to host computation tasks as close as possible to the data sources and end users. In an MEC-enabled camera network, the inference tasks of person re-ID and attribute recognition can thus be offloaded to their nearby computing nodes, e.g., cloudlets or base stations with artificial intelligence accelerators, thereby reducing transmission delay significantly.

Several works have been proposed to study the problem of person re-ID and attribute recognition [10, 11, 13–16, 20, 22, 27–29, 37–40]. However, these existing algorithms cannot be simply combined to build a real-time system and cannot be directly employed in MEC-enabled camera networks. The reasons are as follows. First, conventional re-ID algorithm is used for offline image retrieval, so it needs to be redesigned for real-time identification. Existing attribute recognition algorithms typically ignore the identity-association of geographically distributed cameras. So the re-ID and attribute recognition need to be jointly performed in a distributed manner to realize cross-camera attribute recognition. Second, the distributed inference framework consisting of multiple modules has to be effectively deployed into the MEC-enabled camera network; otherwise, it may incur prohibitive inter-module communication delays or processing delays. Therefore, a finer grained integration of distributed inference framework and task assignment policy in MECs are urgently needed to accelerate the smart surveillance system.

Realizing real-time attribute recognition and person re-ID in an MEC-enabled camera network presents major challenges. First, existing methods did not elaborate cross-camera attribute recognition, so how to identify the new coming pedestrians and fuse the recognized attributes in the level of identity are challenging. Second, conventional methods usually process person re-ID and attribute recognition tasks separately. However, the two tasks have a common target of feature learning and benefit each other. Designing a Convolutional Neural Network (CNN) model that can accomplish the two tasks with high accuracy is challenge. Third, inference tasks for person re-ID and attribute recognition are computationally intensive, incurring large computation workload to implement. It will lead to prohibitive

computation latency if executing the inference only on surveillance cameras, due to the constrained computing capability of the cameras. Also, surveillance videos contain redundant data, which causes bandwidth waste and tremendous transmission latency if sending all videos to edge servers to process. Therefore, how to design an inference framework that can fully make use of the edge servers to minimize processing and transmission latencies is challenging. Fourth, the MEC-enabled camera network has many network uncertainties, such as the processing and transmission latencies of cloudlets and base stations of the network. How to distribute the dynamic inference framework to computing nodes while incorporating network uncertainties is another challenge.

In this paper, we focus on designing the distributed inference framework for joint person re-ID and attribute recognition, and explore a non-trivial interplay between the inference framework and network uncertainties of an MEC-enabled camera network, with an aim to maximize the accuracy of the cross-camera attribute recognition and enable real-time inference.

To the best of our knowledge, we are the first to consider cross-camera attribute recognition and person re-ID in an MEC-enabled camera network. The major contributions of this paper include

- We formulate a problem of joint attribute recognition and person re-ID in an MEC-enabled camera network.
- We design a distributed inference framework with a set of carefully designed modules that enables module distribution in a network.
- We propose a CNN-based model that trains attribute recognition and re-ID simultaneously. The accuracy is greatly improved while the size of the model is reduced by 55.2%.
- We investigate the problem of efficiently and effectively distributing its modules to computing nodes in the MEC-enabled camera network, by considering various uncertainties of the network.
- We evaluate the performance of the proposed framework using real datasets in a real test-bed. Results demonstrate that the accuracies of attribute recognition and re-ID are up to 92.9% and 96.6% respectively, while the latency of distributed inference is reduced by at least 40.6%.

2 RELATED WORK

Upon the success of deep learning on automatic feature extraction, CNN-based methods are dominating the pedestrian attribute recognition [10, 11, 13–15, 20, 39, 40] and re-ID [16, 22, 27–29, 37, 38]. Person re-ID and attribute recognition tasks can share the same target of feature learning. Several deep learning approaches are thus proposed, which can execute both tasks in a single model [12, 21, 24]. For example, Lin et al. [12] manually annotated attribute labels for two large-scale re-ID datasets: Market-1501 and DukeMTMC-reID, to learn the feature representation and attribute classifiers at the same time. Almost none of the listed references focused on the design of distributed inference in MEC environments.

Some studies focused on proposing distributed CNN models. For example, Kang et al. [8] developed a regression model to predict the processing latency and energy consumption for each layer with the configuration information. Study in [7] proposed a heuristic that divided a DNN into several partitions and distributed them to the edge servers incrementally. Teerapittayanon et al. [25] mapped partitions of a DNN onto distributed end devices, edge servers and the cloud, by jointly training to consider the trade-off of transmission latency and model accuracy. However, the distribution of partitioned parts into different servers and the network uncertainties are not considered [8], or the design and training of DNN model are neglected [7], or only general DNN models are adopted [25]. Edge computing emerges as a new deployment paradigm of low-latency services for IoT or mobile applications. In the bottom layer, many general

algorithms were proposed to optimize the Qos from the perspective of NFV [18, 30, 31, 34, 35] or caching [32, 33, 36]. In the application layer, some researchers focused on video processing on edges [2–4, 6, 26], which are more similar to our scenario. However, none of the listed video processing references consider the design of a novel NN architecture or ignored the latency uncertainties.

3 PRELIMINARY

In this section, we first introduce the system model, notations and notions. We then define the problems precisely.

3.1 An MEC-enabled surveillance system

We consider a surveillance environment with multiple cameras interconnected by an MEC-enabled network, as illustrated in Figure 1. Let $G = (C \cup V; E)$ be the MEC-enabled camera network, with a set C of surveillance cameras, a set V of edge servers that can implement the inference of deep learning model, and a set E of links (wired or wireless) connecting cameras and edge servers. Denote by $c_i \in C$ a camera. Transferring video stream data out of each camera c_i through links in E consumes network bandwidth resources and incurs communication latencies. The latencies of network links depend on many factors, such as the congestion level of the link. Such latencies thus usually are uncertain and cannot be obtained in advance. The computing resource in each edge server $v \in V$ is used to perform real-time person re-ID and attribute recognition, thereby incurring processing delay.

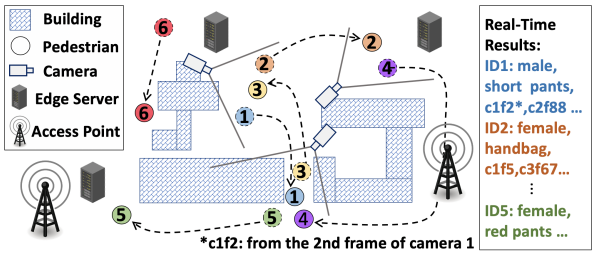

Fig. 1. An example of the MEC-enabled camera network for joint attribute recognition and person re-ID.

3.2 Uncertainties of latencies and inference requests

Processing video streams in an MEC-enabled network incurs latency, which includes the transmission delay by transferring videos or features from cameras to edge servers and the processing delay in edge servers. Such latencies in different edge servers and communication links in the MEC vary significantly from time to time. Notice that we did not consider the transmission delay by transferring analysis results from edge servers to end devices, since the size of analysis results is typical small.

Let r_j be an inference request for processing in the MEC-enabled network. Each r_j carries an amount of video data for processing. The delay experienced by r_j for processing its data stream in an edge server depends on both its data rate and the workload of the server during time slot t. It thus varies in different time slots and is usually not known in advance. Since the inference request is scheduled per time slot, we assume that the delay of experienced by each request does not change during time slot t, and can be obtained at the very beginning of time slot t. Similarly, the latency of transmitting the video stream of inference request r_j in each link $e \in E$ varies time by time and is uncertain when r_j arrives into the system.

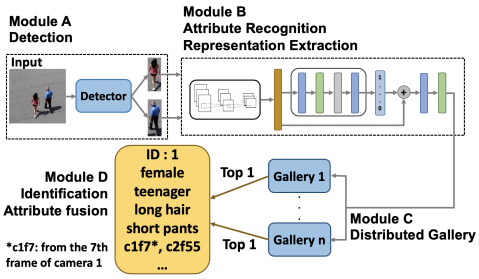

Fig. 2. The distributed inference framework for joint attributes recognition and person re-ID.

3.3 Problem definition

Given an MEC-enabled camera network $G = (C \cup V; E)$, a set of online inference requests $R(t)$ at a beginning of each time slot t, the *distributed inference problem for real-time attribute recognition and person re-ID* is to partition an inference request into different modules and distributing the modules to edge servers, such that the accuracies of person re-ID and attribute recognition are maximized while the latency of inference is minimized, subject to the resource constraints of edge servers.

4 A DISTRIBUTED INFERENCE FRAMEWORK

4.1 The framework design

We observe that there is a lot of redundant information in the original video streams collected by cameras in each request r_j. To avoid the long delay incurred of transmitting such information, we use a pedestrian detector in each camera to extract effective images of pedestrians from the video stream. In addition, to recognize the attributes and id of a pedestrian, conventional methods train two separate models for attribute recognition and re-ID, respectively. This however may not be efficient in an MEC-enabled network, because two CNN models need to be executed and stored in each edge server, thereby introducing additional processing delays and occupying more storage resource. Motivated by this, we design a single CNN model for both person re-ID and attribute recognition.

Our basic idea is to partition the inference framework to different modules, with a minimum amount of inter-module communications. These modules can be distributed to different edge servers, realizing distributed inference. Besides, multiple modules can be run in a server. Since the modules need to be stored in the edge servers, the size of the trained model of the modules is also minimized.

In pedestrian re-ID, calculating feature similarity or distance between the query image and all the pedestrian images in gallery is time consuming, where the gallery is a set of photos held by the system. With the continuous running of the system, the gallery can be very large, such that the edge server with the gallery becomes the bottleneck of the system. To accelerate this process, we consider the distributed gallery for parallel computing of pedestrian re-ID. Specifically, we split a single centralized gallery into multiple distributed galleries. We store labeled features of pedestrian images in the gallery, instead of storing images in the conventional method. After obtaining the attributes of a person, we need to look-up the whole distributed gallery group to identify the person. The final analysis result is shown in Figure 1. The system records an instance for each detected person, which contains the id, the attributes and the frame information. The frame information describes when and where the person was detected.

As shown in Figure 2, the proposed distributed inference framework consists of the following key modules:

Module A: This module is called pedestrian detector that runs in each camera c_i. The pedestrian detector detects pedestrian images from real-time video stream, and transmits the obtained images as the query persons to the successor modules of the proposed inference framework. Note that the frame information is always attached to the inter-module transmission, although we will not explicitly indicate it due to its negligible size.

Module B: This module receives the pedestrian images from the pedestrian detector of each camera c_i. It then recognizes the attributes of the person by employing CNN model. Eventually, the output of attribute prediction is concatenated with the extracted feature. The subsequent two layers are the full connection layer and the batch normalization layer. Finally, the output feature will be sent to each instance of Module C for re-ID.

Module C: There are multiple instances of Module C in the framework. Each instance is distributed to different edge servers and holds a unique distributed gallery. The $gallery_i$ stores the features of pedestrians from c_i. After receiving the query feature from Module B, all instances of Module C calculates the similarity between the query and all the features stored in parallel. The similarity will be sorted in descending order. Afterwards, the maximum similarity with its corresponding ID label will be sent to Module D.

Module D: This module receives the local maximum labeled similarity from every instance of Module C, and sorts them in descending order. It then obtains the global maximum similarity. We define a threshold ε, which is discussed statistically in Section. 6.3. If the global maximum similarity is greater than ε, it shows that the person has been detected by the system, and the ID of the query person is the one with the global maximum similarity. Then, the new attributes need to be fused with the old attributes in the instance by the method based on exponentially weighted average. Otherwise, the query person has not been detected by any camera. In this case, a new instance for the person needs to created, by assigning the person with a unique ID and storing its attributes. In either case, the frame information will be added and the id result needs to be sent back for gallery update.

4.2 Model training

Existing joint training models of attribute recognition and re-ID usually consider the recognition of each attribute as a single task, by using separate classifier for each attribute [12]. This however contains many redundant information and leads to large training models that cannot be effectively distributed in the MEC-enabled camera network. Instead, we consider all the attributes at the same time and the relationship among attributes are learnt simultaneously. Partially similar to [12], we adopt a multi-task network in the model training, which can learn an identity classifier and an attribute classifier at the same time. Regarded as auxiliary cues, the output of attribute prediction is concatenated with the feature extracted by the CNN. The concatenated feature is the input of the identity classifier, as shown in Figure 3. For re-ID, the loss function $Loss_{reid}$ is a categorial cross entropy over identity label, i.e.,

$$Loss_{reid} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=0}^{K} y_{ik}\log(\hat{p}_{ik}), \tag{1}$$

where $\hat{p}_{ik}$ is the predicted probability that $sample_i$ is the person k. y_{ik} is the ground truth label indicates whether the $sample_i$ is the person k or not. N is the number of training samples. K represents the number of identities in the training set.

To minimize the size and improve the accuracy of the trained model, we reduce the redundant and inefficiency in attribute recognition, by considering all the attributes at the same time and learning the relationship among attributes simultaneously [10]. The loss function $Loss_{attr}$ is a sigmoid cross entropy loss function which learns all the attributes

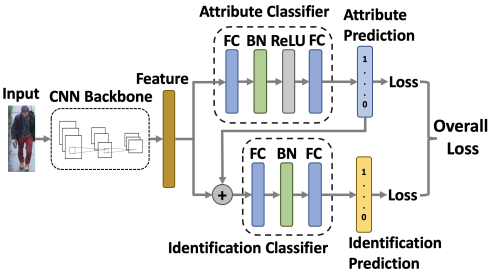

Fig. 3. Training model

jointly:

$$Loss_{attr} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=0}^{M} w_j(y_{ij}\log(\hat{p}_{ij}) + (1-y_{ij})\log(1-\hat{p}_{ij})),$$

with a loss weight for attribute j to deal with the unbalancedness of attribute distributions, i.e.,

$$w_j = exp(-\rho_j/\sigma^2), \tag{2}$$

where $\hat{p}_{ij}$ is the predicted probability for the attribute j of $sample_i$. y_{ij} is the ground truth label indicating whether the $sample_i$ has the attribute j or not. ρ_j is the ratio of attribute j in the training set. σ is a hyper-parameter. M is the number of attributes. We define the overall loss function:

$$Loss = \lambda Loss_{reid} + ((1-\lambda)/M)Loss_{attr}, \tag{3}$$

where λ is a hyper parameter to balance the two sub losses, which is set to 0.5 in our experiments.

5 MODULE DISTRIBUTION BASED ON CONTEXTUAL MULTI-ARMED BANDITS

The basic idea of the proposed module distribution algorithm is to adopt an online-learning framework based on the Contextual Multi-Armed Bandit model, as shown in Figure 4. In the proposed inference framework, the locations of Modules B and C play a vital role in the latency of online inference requests. We observe that the network latencies, size of gallery and their popularity depend on a *context* of the current camera network and the environment it monitors, such as the busyness of a camera, the time (rush hours or holiday hours), and etc. Thus, the proposed online-learning algorithm does not run all the time but re-learns when the context space changes dramatically. Normally, such contexts can be observed by the agents before making decisions. For simplicity, we consider the transmission and processing delays as the contexts of the network.

We consider that there is an agent representing each to-be-distributed modules. For an edge server v and a link e that connects to v, the agent of each module decides whether to assign its module to v and transmit its input data via link e. An edge server v with a link e that connects to v is considered as an arm, also known as an action. Each of such agents needs to choose an *arm* based on the current context of the camera network. For clarity, we define a *policy* as a mapping from contexts to arms. Denote by Π the set of policies from which an agent chooses its policy. We divide the processing and transmission delays into L levels, with each level l ($1 \le l \le L$) denoting a fixed range of delay. Let d_v^{max} and d_v^{min} be the maximum and minimum values of the delay experienced by request r_j in server v in time slot t of a monitoring period T, and denote by d_e^{max} and d_e^{min} the maximum and minimum values for of links. The context thus consists of different levels of processing and transmission delays of the camera network. Contexts arrive as independent elements from some fixed contexts set. The arm of each agent thus can be represented as a set of $\{0,1\}$, where 1 denotes the selection of v and e while 0 denotes they are not selected.

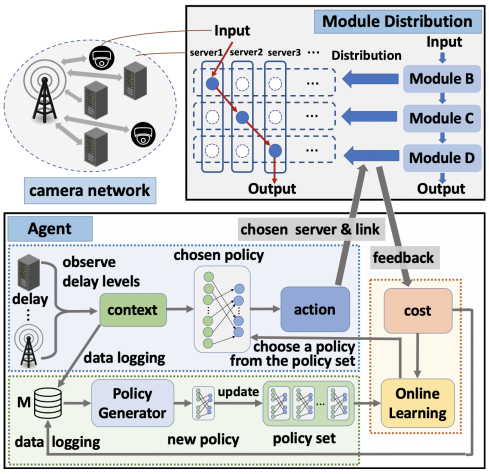

Fig. 4. The online-learning framework.

We use the normalized end-to-end delay as the feedback, denoted by *cost*. The delay of a certain task is approximately fixed when the computation and transmission conditions are the same. So we assume that the *cost* with certain arm and context is chosen by a deterministic oblivious adversary before each round. The *cost* of round t is denoted by $c_t(a)$, where the observed context of round t is x_t and a is the chosen arm. The cost of policy π after T rounds is:

$$cost(\pi) = \sum_{t=1}^{T} c_t(\pi(x_t)). \tag{4}$$

In [23], an existing benchmark of contextual bandits is the best-response policy:

$$\pi^*(x_t) = \min_{a \in \mathcal{A}} c_t(a), \tag{5}$$

where $\mathcal{A}$ is the set of arms. The corresponding regret is:

$$R^*(T) = cost^{OL} - cost^*, \tag{6}$$

where the $cost^{OL}$ is the total regret accumulated in the T rounds and $cost^* = cost(\pi^*)$. The regret is based on the assumption: $\pi^* \in \Pi$ for each round t. However, the Π in **Algorithm** 2 `Online Learning` is updated every I rounds. The best-response policy π^* may not initially be in Π, but be added through an update. So we now define a more general regret:

$$R(T) = cost^{OL} - cost^{\sharp}, \tag{7}$$

where $cost^{\sharp} = \sum_{t=1}^{T} \min_{\pi \in \Pi} c_t(\pi(x_t))$. Note that $R^*(T)$ is a special case of $R(T)$ when π^* is initially in Π.

The detailed steps of the proposed `ModDistMAB` algorithm are illustrated in **Algorithm** 1. It contains two sub algorithms `Online Learning` and `Policy Generator`, which are referred to as **Algorithm** 2 and **Algorithm** 3.

5.1 Regret analysis

We now analyze the bounds of the regret of algorithms `ModDistMAB` and `Online Learning` in the following theorems.

Theorem 5.1. *The regret of algorithm* `Online Learning` *with parameter* $\gamma \in [0, \frac{1}{\sqrt{T}})$ *is bounded by* $O(\sqrt{T|\mathcal{A}|\ln|\Pi|})$.

Proof. We first introduce three facts:

Algorithm 1 ModDistMAB

Input: L: The number of delay levels; $\mathcal{A}$: The set of all arms; M: The memory that holds the statistical information of the delays of the camera network; N: Data collection duration; P: The number of policies;

Initialize a memory M that holds the statistical information of the delays of the camera network. Initialize an empty policy set Π;
for each time slot $t \leftarrow 1, \cdots, N$ **do**
  Observe the processing delay $d_t(v)$ for each edge server v and the transmission delay $d_t(e)$ for each link e;
  Pick an arm a_t from $\mathcal{A}$ uniformly;
  Observe the cost $c_t(a_t)$ of the chosen arm;
  Add the data point $(d_t(\cdot), a_t, c_t(a_t))$ to the memory M;
end for
Get the maximum and minimum delays d_v^{max}, d_v^{min}, d_e^{max} and d_e^{min} from the memory M;
Generate the context set $\mathcal{X}$ of the camera network with the delay ranges being divided into L levels;
Convert all of the data points $(d_t(\cdot), a_t, c_t(a_t))$ to $(x_t, a_t, c_t(a_t))$, $x_t \in \mathcal{X}$;
Call **Algorithm** 3 Policy Generator with different σ and S to get P new policies;
Add the P new policies to the policy set Π;
Online Learning: call **Algorithm** 2.

Algorithm 2 Online Learning

Input: $\mathcal{X}$: The set of all contexts; Π: The set of all policies; $\mathcal{A}$: The set of all arms; M: The memory that holds the statistical information of the delays of the camera network; I: The update cycle of policy; ϵ: The parameter for weight update. $\epsilon \in (0, \frac{1}{2})$; γ: The probability of picking an arm uniformly at random;

Initialize the weight as $w_1(a) = 1$ for each arm a;
for each time slot $t \leftarrow 1, \cdots, T$ **do**
  Observe the context of the camera network x_t;
  Randomly generate a float number n;
  if $n < \gamma$ **then**
    Pick an arm a_t uniformly at random;
    Observe the cost $c_t(a_t)$ of the chosen arm;
    Add data point $(x_t, a_t, c_t(a_t))$ to the memory M;
    For each policy π, $w_{t+1}(\pi) = w_t$;
  else
    For each policy π, let $p_t(\pi) = w_t(\pi)/(\sum_{\pi' \in \Pi} w_t(\pi'))$;
    For each arm a, let $q_t(a) = \sum_{\pi \in \Pi: \pi(x_t)=a} p_t(\pi)$
    Sample a policy π_t from distribution $p_t(\cdot)$;
    Use the policy π_t to pick an arm $a_t = \pi_t(x_t)$;
    Observe the cost $c_t(a_t)$ of the chosen arm;
    Add data point $(x_t, a_t, c_t(a_t))$ to the memory M;
    Get the maximum cost c^{max} and the minimum cost c^{min} from M;
    Normalize the current cost: $c_t(a_t) = (c_t(a_t) - c^{min})/(c^{max} - c^{min})$
    For each arm a, define fake costs: $\hat{c}_t(a) = \begin{cases} \frac{c_t(a_t)}{q_t(a_t)} & a = a_t \\ 0 & otherwise \end{cases}$;
    For each policy π, define fake costs: $\hat{c}_t(\pi) = \hat{c}_t(\pi(x_t))$;
    For each policy π, update its weight $w_{t+1}(\pi) = w_t(\pi) \cdot (1-\epsilon)^{\hat{c}_t(\pi)}$;
  end if
  if $t \equiv 0 \pmod{I}$ **then**
    Call **Algorithm** 3 with different σ and S to get $(|\Pi| - 1)$ new policies;
    Keep the policy $\pi \in \Pi$ with the largest $w_{t+1}(\pi)$. Replace the other $(|\Pi| - 1)$ old policies with the new generated policies, inheriting the w value.
  end if
end for

For all $x > 0$, there exist $\alpha, \beta \geq 0$, possibly dependent on ϵ:

$$fact : (1-\epsilon)^x < 1 - \alpha x + \beta x^2. \tag{8}$$

For $\epsilon \in (0, 1/2)$:

$$fact : \epsilon < \ln(1/(1-\epsilon)) < 3\epsilon. \tag{9}$$

For any $x \in (0, 1)$:

$$fact : \ln(1-x) < -x. \tag{10}$$

Algorithm 3 `Policy Generator`

Input: $\mathcal{A}$: The set of all arms; M: The memory that holds the statistical information of the delays of the camera network; σ: The parameter in arm weight; S: The dictionary of training strategy, such as learning rate, epoch; O: The interface of machine learning models for classification task. We used NN in our experiments;

Output: A new policy π: the trained classification model;

1: Get all data points $(x_t, a_t, c_t(a_t))$ from the memory M;
2: Let X' be the set of x_t in data points;
3: For each context $x \in X'$, find the arm a with the smallest cost $c_t(a_t)$ in data points, named as a^x
4: For each arm $a \in \mathcal{A}$:
let $\rho_a = \sum_{x \in X': a^x = a} \frac{1}{|X'|}$, $w_a = exp(\frac{-\rho_a}{\sigma^2})$
5: We model the policy as a multi-class classification problem in machine learning. The contexts are the samples and the arms are the predicted classes. The set (x, a^x), $x \in X'$ is the training data. Each context x is the input feature and a^x is the label. w weights the loss function to balance the distribution of arms;
6: Train the model O with the strategy S.

We now show the upper bound of $\hat{c}_t(\pi)$ and $\hat{c}_t(a)$. Let $a = \pi(x_t)$ be the arm chosen by policy π. After the normalization, $c_t(a) \in [0, 1]$ for each arm a. We obtain:

$$\hat{c}_t(\pi) = \hat{c}_t(a) \leq c_t(a)/q_t(a) \leq 1/q_t(a). \tag{11}$$

In order to demonstrate that the fake cost $\hat{c}_t(a)$ is the unbiased estimate of the true cost $c_t(a)$. For each arm a, we have:

$$\mathbb{E}[\hat{c}_t(a)] = Pr[a = a_t] \cdot c_t(a)/q_t(a) + Pr[a \neq a_t] \cdot 0 = c_t(a). \tag{12}$$

We start with the case of $\gamma = 0$, which means that the algorithm always picks an arm by policy. Let $W_t = \sum_{\pi \in \Pi} w_t(\pi)$, we have:

$$W_{T+1} > w_{T+1}(\pi^*) = (1-\epsilon)^{cost^*} \geq (1-\epsilon)^{cost^\sharp}. \tag{13}$$

Let $\alpha = \ln(\frac{1}{1-\epsilon}), \beta = \alpha^2$ and use the fact in Eq. (8), we can get

$$\begin{aligned}
&W_{t+1}/W_t = \sum\nolimits_{\pi \in \Pi} (1-\epsilon)^{\hat{c}_t(\pi)} \cdot (w_t(\pi))/W_t \\
&\leq \sum\nolimits_{\pi \in \Pi} (1 - \alpha \hat{c}_t(\pi) + \beta \hat{c}_t(\pi)^2) \cdot p_t(\pi) \\
&= \sum\nolimits_{\pi \in \Pi} p_t(\pi) - \alpha \sum\nolimits_{\pi \in \Pi} p_t(\pi) \hat{c}_t(\pi) + \beta \sum\nolimits_{\pi \in \Pi} p_t(\pi) \hat{c}_t(\pi)^2) \\
&= 1 - \alpha \mathbb{E}[\hat{c}_t(\pi)] + \beta \mathbb{E}[\hat{c}_t(\pi)^2].
\end{aligned}$$

To get the upper bound of $\mathbb{E}[\hat{c}_t(\pi)^2]$, we have:

$$\begin{aligned}
&\mathbb{E}[\hat{c}_t(\pi)^2] = \sum_{\pi \in \Pi} p_t(\pi) \cdot \hat{c}_t(\pi)^2 = \sum_{a \in \mathcal{A}} \sum_{\pi \in \Pi: \pi(x_t) = a} p_t(\pi) \hat{c}_t(\pi) \hat{c}_t(\pi) \\
&\leq \sum_{a \in \mathcal{A}} \sum_{\pi \in \Pi: \pi(x_t) = a} \frac{p_t(\pi)}{q_t(a)} \hat{c}_t(a) = \sum_{a \in \mathcal{A}} \frac{\hat{c}_t(a)}{q_t(a)} \sum_{\pi \in \Pi: \pi(x_t) = a} p_t(\pi) \\
&= \sum\nolimits_{a \in \mathcal{A}} \hat{c}_t(a).
\end{aligned} \tag{14}$$

The mathematical expectation of $\mathbb{E}[\hat{c}_t(\pi)^2]$ thus is:

$$\begin{aligned}
&\mathbb{E}(\sum\nolimits_{\pi \in \Pi} p_t(\pi) \cdot \hat{c}_t(\pi)^2) \leq \mathbb{E}(\sum\nolimits_{a \in \mathcal{A}} \hat{c}_t(a)) \\
&= \sum\nolimits_{a \in \mathcal{A}} \mathbb{E}(\hat{c}_t(a)) = \sum\nolimits_{a \in \mathcal{A}} \mathbb{E}(c_t(a)) \leq |\mathcal{A}|.
\end{aligned} \tag{15}$$

According to the fact in Eq. (10) and $\alpha\mathbb{E}[\hat{c}_t(\pi)] - \beta\mathbb{E}[\hat{c}_t(\pi)^2] \in (0,1)$, we can derive

$$\begin{aligned} \ln(W_{t+1}/W_t) &< \ln(1 - \alpha\mathbb{E}[\hat{c}_t(\pi)] + \beta\mathbb{E}[\hat{c}_t(\pi)^2]) \\ &< -\alpha\mathbb{E}[\hat{c}_t(\pi)] + \beta\mathbb{E}[\hat{c}_t(\pi)^2]. \end{aligned} \tag{16}$$

To calculate the expected regret, we need to connect $cost^{OL}$ and $cost^{\sharp}$. By transposing and summing over t on both sides, we have:

$$\begin{aligned} &\sum_{t\in[T]} (\alpha\mathbb{E}[\hat{c}_t(\pi)] - \beta\mathbb{E}[\hat{c}_t(\pi)^2]) = \alpha\mathbb{E}[cost^{OL}] - \beta \sum_{t\in[T]} \mathbb{E}[\hat{c}_t(\pi)^2] \\ &< -\sum_{t\in[T]} \ln(W_{t+1}/W_t) = -\ln \prod_{t\in[T]} W_{t+1}/W_t = -\ln(W_{T+1}/W_1) \\ &= \ln W_1 - \ln W_{T+1} < \ln|\Pi| - cost^{\sharp} \ln(1-\epsilon) \\ &= \ln|\Pi| + \alpha\mathbb{E}[cost^{\sharp}]. \end{aligned} \tag{17}$$

We now calculate the expected regret:

$$\begin{aligned} &\mathbb{E}[cost^{OL} - cost^{\sharp}] < (\ln|\Pi|)/\alpha + \alpha \sum\nolimits_{t\in[T]} \mathbb{E}[\hat{c}_t(\pi)^2] \\ &\leq (\ln|\Pi|)/\alpha + \alpha T|\mathcal{A}| \leq (\ln|\Pi|)/\epsilon + 3\epsilon T|\mathcal{A}|, \end{aligned} \tag{18}$$

where the upper bound of $\mathbb{E}[\hat{c}_t(\pi)^2]$ is obtained in Eq. (15) and the last step is based on the fact in in Eq. (9). Let $\epsilon = \sqrt{(\ln|\Pi|)/(3T|\mathcal{A}|)}$, and we have

$$\mathbb{E}[R(T)] = \mathbb{E}[cost^{OL} - cost^{\sharp}] < 2\sqrt{3}\sqrt{T|\mathcal{A}|\ln|\Pi|}, \tag{19}$$

due to Eq. (7).

For the extended case of $\gamma \in [0, \frac{1}{\sqrt{T}})$, each round has the probability γ to pick an arm randomly, contributing at most 1 to the expected regret. So we have:

$$\begin{aligned} \mathbb{E}[R(T)] &< 2\sqrt{3}\sqrt{T(1-\gamma)|\mathcal{A}|\ln|\Pi|} + \gamma T \\ &\leq 2\sqrt{3}\sqrt{T|\mathcal{A}|\ln|\Pi|} + \sqrt{T} \leq O(\sqrt{T|\mathcal{A}|\ln|\Pi|}). \end{aligned}$$

□

Theorem 5.2. *The regret of algorithm* `ModDistMAB` *is bounded by* $O(\sqrt{T|E||V|})$.

Proof. To obtain the bound of expected regret, we show the size of arm space and policy space. First, we know that there are $|V|$ servers in the MEC-enabled camera network and $|E|$ links. Denote by K the number of arms in the arm space. Clearly, we have $K = O(|V||E|)$. Second, for the policy space, according to algorithm `ModDistMAB`, the processing and transmission delays are divided into L levels. There are at most $O(L^{|E||V|})$ contexts for all edge servers and links. By enumeration, there are $O(K^{L^{|E||V|}})$ possible policies in Π. As shown in Theorem 5.1, we know that the regret bound of **Algorithm** 2 is $O(\sqrt{T|\mathcal{A}|\ln|\Pi|})$, where $|\mathcal{A}|$ is the size of arm set. Plugging in this Π, we obtain the regret is $O(\sqrt{T|E||V|L^{|E||V|}\ln(|E||V|)})$. Due to the exponential dependence on $|E||V|$, the regret increases dramatically with the scaling up of the system. In addition, the running time per round of **Algorithm** 2 is $O(K^{L^{|E||V|}})$, which reflects the algorithm is extremely slow in practice. To solve this problem, the key is to reduce the size of policy set Π. Instead of getting all possible policies by enumeration, we use **Algorithm** 3 to generate several suboptimal policies. Obtaining the suboptimal policy set, **Algorithm** 2 finds the optimal one by online learning. Denote by P the number of generated

policies. We obtain the regret bound of **Algorithm** 2 is $O(\sqrt{T|E||V|\ln P})$. Both regret bound and running time are significantly reduced.

In `ModDistMAB`, the algorithm picks the arm uniformly to collect data for the first N rounds. Each round in this duration contributes at most 1 to regret. The **Algorithm** 2 runs in the subsequent $(T - N)$ rounds. So we can get the regret bound

$$O(\sqrt{(T-N)|E||V|\log P + N}) = O(\sqrt{T|E||V|}),$$

assuming that N and P are the given constants. □

6 EXPERIMENTS

6.1 Training and test-bed settings

For the training process, we adopt ResNet-50 [5] as the backbone, which is pre-trained on ImageNet [19]. For pedestrian re-ID, we append a 512-dim fully connected layer, a batch normalization layer and a dropout layer after the pool5 layer [12], without applying the ReLU activation function. For the attribute recognition, we use a similar structure as re-id part. The difference is that we append a ReLU layer after batch normalization layer. For the training strategy, the number of epochs is 80. The batch size is 32. The learning rate is 0.02 and warm up in 10 epochs. The input image will be resized to 384×128 and random erasing is applied.

In our test-bed, we use two Raspberry Pi 4B boards (4GB RAM) with cameras, as shown in Figure 5 (a). We adopt the OpenVINO toolkit and use a pre-trained pedestrian detector in it. Then we use Intel Neural Compute Stick 2 plugged in USB ports to accelerate the inference of CNN. Raspberry Pi will send the detected pedestrian images to the edge nodes for further inference. The test-bed also consists of four edge servers with GPU (2080 Ti) that are interconnected by five hardware switches, as shown in Figure 5(b).

(a) (b)

Fig. 5. A test-bed for the MEC-enabled camera network.

6.2 Datasets and baseline algorithm

The accuracy is evaluated on two pedestrian re-ID datasets with attribute annotation: Market-1501 [41] and DukeMTMC-reID [43]. We use video sequences from MOTChallenge [17] [9] and evaluate our distributed algorithm for acceleration on them, which are filmed with static and elevated cameras in unconstrained environments. For the sake of fairness, we should compare our method with the similar one that considering the joint training of attribute recognition and person re-ID. So we choose an existing study by [12], which is referred to as Baseline.

Methods	Rank-1	Rank-5	Rank-10	mAP
Baseline	87.04	95.10	96.42	66.89
Ours	93.05	97.57	98.34	81.60
$Ours_{framejunk}$	**96.44**	**98.72**	**99.23**	**84.30**

Table 1. CMC and mAP on Market-1501

Methods	Rank-1	Rank-5	Rank-10	mAP
Baseline	73.92	-	-	55.56
Ours	85.37	92.55	94.84	70.75
$Ours_{framejunk}$	**96.59**	**98.79**	**99.10**	**79.83**

Table 2. CMC and mAP on DukeMTMC-reID

6.3 Model accuracy results

The Cumulative Matching Characteristic (CMC) curve and the mean average precision (mAP) are adopted for pedestrian re-ID evaluation. In the previous CMC calculation, if the gallery image is under the same camera as the query image, it is considered as junk image. However, in our application, images from the same camera are used as effective information to identify the current query image. So we only consider the gallery images which are from the same frame as the query image as junk images. Our proposed CNN model based on this new evaluation metric is referred to as $Ours_{framejunk}$, while the proposed model based on conventional CMC is denoted by *Ours*. The results on Market-1501 and DukeMTMC-reID are shown in Tables 1 and 2, respectively. It can be seen from the tables that *Ours* has the improvement of 6.0% in Rank-1 and 14.7% in mAP compared with the Baseline approach. The reason is that higher attribute accuracy promotes the accuracy of re-ID. Also, we adopt a novel training strategy that is suitable for person re-ID. $Ours_{framejunk}$ has an accuracy of up to 96.44% for Rank-1, indicating the identification accuracy of our system is up to 96.44%. The accuracy improvement of $Ours_{framejunk}$ compared with *Ours* confirms that the images from the same camera are the conducive information for identification.

For the attribute recognition, we test the classification accuracy for each attribute. For geographically distributed cameras fusion, we fuse the prediction output of the same person in different images, which is referred to as $Ours_{fused}$. The results on Market-1501 are shown in the table 3. *Ours* is much better than Baseline. The reason is that we adopt a single classifier for all the attributes, which is better than a classifier for each attribute. This allows the relationship among attributes to be learnt simultaneously. In addition, the accuracy improvement of $Ours_{fused}$ compared with *Ours* means that cross-camera can improve the performance of attribute recognition.

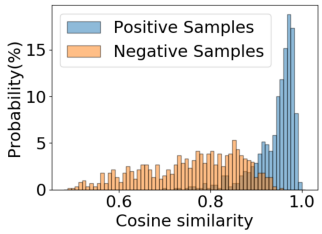

Fig. 6. The maximum similarity between query images and the gallery

To distinguish whether the current query person exists in the gallery, we make a statistic on the dataset, inspired by [42]. For each identity in the gallery, we select one image as a positive query sample. For each identity not in the gallery, we select one image as a negative query sample. The similarity results are shown in Figure 6. From the results, we can see that when the threshold as 0.9 the negative and positive samples can be separated sufficiently. We thus set the threshold ε to 0.9 in Module D.

6.4 Testbed experiments

To evaluate the effectiveness, we compare the proposed `ModDistMAB` algorithm with the other four, as shown in the Figure 7. In order for the algorithms to converge, we ensure that no other tasks except for our modules are being processed in each edge server and no other data is being transferred in each link. It means that the network environment

	gender	age	hair	L.slv	L.low	S.clth	
Baseline	88.9	88.6	84.4	93.6	93.7	92.8	
Ours	93.1	93.8	89.3	93.5	94.1	92.6	
$Ours_{fused}$	**94.5**	**95.2**	**90.3**	**94.0**	**95.5**	**93.5**	
	B.pack	H.bag	bag	hat	C.up	C.low	Avg
Baseline	84.9	90.4	76.4	97.1	74.0	73.8	86.6
Ours	87.7	89.8	78.5	**97.7**	95.4	94.5	91.7
$Ours_{fused}$	**91.3**	**89.9**	**81.7**	97.3	**96.0**	**95.0**	**92.9**

Table 3. Attribute recognition accuracy on Market1501-attribute

is static and our modules are the only variables. The `Fixed` algorithm permanently assigns all the modules to the same edge server with the strongest computing power. We can see that this method performs worst. The reason is that the chosen edge server is overloaded, causing prohibitive huge processing overhead. The `Greedy` algorithm always selects the edge server that can achieve the minimum processing latency, based on the observations of the current network context and the historical information. However, the `Greedy` algorithm do not consider the transmission delay, so it converges to a bad delay. `ModDistMAB without policy updating` always uses the initially generated policy set Π. It can be seen from the Figure 7, its delay is hardly reduced and converges to a large value. We found that the context space is small because of the static network. During the data collection phase, every possible context is recorded. It means that the policy training set completely covers the feature space, so the classification in **Algorithm** 3 does not predict, but records the optimal action before. Although different training strategies are adopted, the initial P policies still have the same mapping relationship. Therefore, the online learning algorithm does not work, and the delay does not decrease with running. Besides, due to insufficient data collection, there are many wrong labels in **Algorithm** 3. The delay is large because the policies make many mistakes. `ModDistMAB without online learning` only trains one policy based on the constantly updated data and uses it. Because online learning does not work in the static network, its performance is similar to `ModDistMAB`, converging to a small delay.

However, in practice, the network is dynamic and must have other tasks. The context space is large in the dynamic network. So the policies generated by machine learning will vary greatly, which means the online learning algorithm works. We randomly add other computation and transmission tasks to the Testbed to simulate the dynamic network. As shown in Figure 8, we can see that `ModDistMAB` is always better than `ModDistMAB without online learning` after going through 1900 rounds. The reason is that `ModDistMAB without online learning` only uses a suboptimal policy generated by **Algorithm** 3, while `ModDistMAB` algorithm will further find out the optimal policy from the set of suboptimal policies through online learning. It shows that the `ModDistMAB` algorithm is the most efficient one in practice.

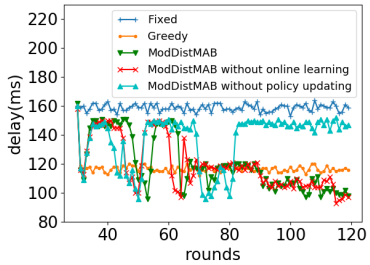

Fig. 7. The performance of algorithms in the static network with a data collection duration of 30.

7 CONCLUSIONS

In this paper, we studied the problem of real-time pedestrian attribute recognition and re-ID in an MEC-enabled camera network. We first proposed a novel distributed inference framework with a set of distributed modules jointly considering

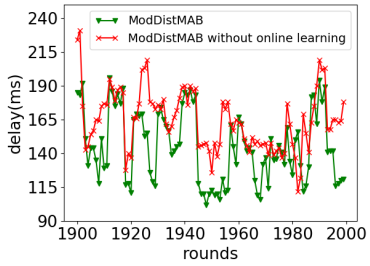

Fig. 8. The performance of algorithms ModDistMAB and ModDistMAB without online learning in the dynamic network. In practice, 1900 rounds may take only 6 minutes and the policy may be trained in earlier rounds.

the attribute recognition and person re-ID. We also devised an algorithm for the module distributions by proposing a novel Contextual Multi-Armed Bandit model, to address the network uncertainties of an MEC-enabled camera network. We then evaluated the performance of the proposed distributed inference framework and algorithm by both simulations with real datasets and system implementation. Results show that the accuracy of the distributed inference framework is up to 96.6% for identification and 92.9% for attribute recognition, and the inference delay is at least 40.6% lower than algorithm Fixed.

REFERENCES

[1] E. Ahmed, M. Jones, and T. K. Marks. 2015. An improved deep learning architecture for person re-identification. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 3908–3916.

[2] G. Ananthanarayanan, P. Bahl, P. BodÃŋk, K. Chintalapudi, M. Philipose, L. Ravindranath, and S. Sinha. 2017. Real-Time Video Analytics: The Killer App for Edge Computing. Computer 50, 10 (2017), 58–67.

[3] Tarek Elgamal and Klara Nahrstedt. 2020. Serdab: An IoT Framework for Partitioning Neural Networks Computation across Multiple Enclaves. (2020).

[4] Tarek Elgamal, Atul Sandur, Phuong Nguyen, Klara Nahrstedt, and Gul Agha. 2018. DROPLET: Distributed Operator Placement for IoT Applications Spanning Edge and Cloud Resources. (2018), 1–8.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. (2016), 770–778.

[6] Chienchun Hung, Ganesh Ananthanarayanan, Peter Bodik, Leana Golubchik, Minlan Yu, Paramvir Bahl, and Matthai Philipose. 2018. VideoEdge: Processing Camera Streams using Hierarchical Clusters. (2018), 115–131.

[7] Hyuk-Jin Jeong, Hyeon-Jae Lee, Chang Hyun Shin, and Soo-Mook Moon. 2018. IONN: Incremental offloading of neural network computations from mobile devices to edge servers. In Proceedings of the ACM Symposium on Cloud Computing. 401–411.

[8] Yiping Kang, Johann Hauswald, Cao Gao, Austin Rovinski, Trevor Mudge, Jason Mars, and Lingjia Tang. 2017. Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. In ACM SIGARCH Computer Architecture News, Vol. 45. ACM, 615–629.

[9] Laura Leal-Taixé, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler. 2015. Motchallenge 2015: Towards a benchmark for multi-target tracking. arXiv preprint arXiv:1504.01942 (2015).

[10] Dangwei Li, Xiaotang Chen, and Kaiqi Huang. 2015. Multi-attribute learning for pedestrian attribute recognition in surveillance scenarios. In 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR). IEEE, 111–115.

[11] Dangwei Li, Xiaotang Chen, Zhang Zhang, and Kaiqi Huang. 2018. Pose guided deep model for pedestrian attribute recognition in surveillance scenarios. In 2018 IEEE International Conference on Multimedia and Expo (ICME). IEEE, 1–6.

[12] Yutian Lin, Liang Zheng, Zhedong Zheng, Yu Wu, Zhilan Hu, Chenggang Yan, and Yi Yang. 2019. Improving Person Re-identification by Attribute and Identity Learning. Pattern Recognition (2019). DOI:http://dx.doi.org/https://doi.org/10.1016/j.patcog.2019.06.006

[13] Hao Liu, Jingjing Wu, Jianguo Jiang, Meibin Qi, and Bo Ren. 2018b. Sequence-based person attribute recognition with joint CTC-attention model. arXiv preprint arXiv:1811.08115 (2018).

[14] Pengze Liu, Xihui Liu, Junjie Yan, and Jing Shao. 2018a. Localization guided learning for pedestrian attribute recognition. arXiv preprint arXiv:1808.09102 (2018).

[15] Xihui Liu, Haiyu Zhao, Maoqing Tian, Lu Sheng, Jing Shao, Shuai Yi, Junjie Yan, and Xiaogang Wang. 2017. Hydraplus-net: Attentive deep features for pedestrian analysis. In Proceedings of the IEEE international conference on computer vision. 350–359.

[16] Yiheng Liu, Wengang Zhou, Jianzhuang Liu, Guojun Qi, Qi Tian, and Houqiang Li. 2019. An End-to-End Foreground-Aware Network for Person Re-Identification. arXiv: Computer Vision and Pattern Recognition (2019).

[17] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. 2016. MOT16: A benchmark for multi-object tracking. arXiv preprint

arXiv:1603.00831 (2016).

[18] Haozhe Ren, Zichuan Xu, Weifa Liang, Qiufen Xia, Pan Zhou, Omer F Rana, Alex Galis, and Guowei Wu. 2020. Efficient algorithms for delay-aware NFV-enabled multicasting in mobile edge clouds with resource sharing. IEEE Transactions on Parallel and Distributed Systems 31, 9 (2020), 2050–2066.

[19] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S Bernstein, and others. 2015. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision 115, 3 (2015), 211–252.

[20] Nikolaos Sarafianos, Xiang Xu, and Ioannis A Kakadiaris. 2018. Deep imbalanced attribute classification using visual attention aggregation. In Proceedings of the European Conference on Computer Vision (ECCV). 680–697.

[21] Arne Schumann and Rainer Stiefelhagen. 2017. Person re-identification by deep learning attribute-complementary information. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 20–28.

[22] Chen Shen, Guo-Jun Qi, Rongxin Jiang, Zhongming Jin, Hongwei Yong, Yaowu Chen, and Xian-Sheng Hua. 2018. Sharp Attention Network via Adaptive Sampling for Person Re-identification. CoRR abs/1805.02336 (2018). http://arxiv.org/abs/1805.02336

[23] Aleksandrs Slivkins. 2019. Introduction to multi-armed bandits. arXiv preprint arXiv:1904.07272 (2019).

[24] Chi Su, Shiliang Zhang, Junliang Xing, Wen Gao, and Qi Tian. 2018. Multi-type attributes driven multi-camera person re-identification. Pattern Recognition 75 (2018), 77–89.

[25] Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. 2017. Distributed deep neural networks over the cloud, the edge and end devices. In 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). IEEE, 328–339.

[26] Junjue Wang, Ziqiang Feng, Zhuo Chen, Shilpa George, Mihir Bala, Padmanabhan Pillai, Shaowen Yang, and M Satyanarayanan. 2018a. Bandwidth-Efficient Live Video Analytics for Drones Via Edge Computing. (2018), 159–173.

[27] Menglin Wang, Baisheng Lai, Zhongming Jin, Xiaojin Gong, Jianqiang Huang, and Xiansheng Hua. 2018b. Deep Active Learning for Video-based Person Re-identification. arXiv: Computer Vision and Pattern Recognition (2018).

[28] Wangmeng Xiang, Jianqiang Huang, Xianbiao Qi, Xiansheng Hua, and Lei Zhang. 2018. Homocentric Hypersphere Feature Embedding for Person Re-identification. arXiv: Computer Vision and Pattern Recognition (2018).

[29] Qiaokang Xie, Wengang Zhou, Guojun Qi, Qi Tian, and Houqiang Li. 2019. Progressive Unsupervised Person Re-identification by Tracklet Association with Spatio-Temporal Regularization. arXiv: Computer Vision and Pattern Recognition (2019).

[30] Zichuan Xu, Wanli Gong, Qiufen Xia, Weifa Liang, Omer F Rana, and Guowei Wu. 2020. NFV-enabled IoT service provisioning in mobile edge clouds. IEEE Transactions on Mobile Computing (2020), 1–1.

[31] Zichuan Xu, Weifa Liang, Mike Jia, Meitian Huang, and Guoqiang Mao. 2019. Task Offloading with Network Function Requirements in a Mobile Edge-Cloud Network. IEEE Transactions on Mobile Computing 18, 11 (2019), 2672–2685.

[32] Zichuan Xu, Yugen Qin, Pan Zhou, John C. S. Lui, Weifa Liang, Qiufen Xia, Wenzheng Xu, and Guowei Wu. 2020a. To cache or not to cache: Stable service caching in mobile edge-clouds of a service market. Proc of The 40th IEEE International Conference on Distributed Computing Systems (ICDCS) (2020).

[33] Zichuan Xu, Shengnan Wang, Shipei Liu, Haipeng Dai, Qiufen Xia, Weifa Liang, and Guowei Wu. 2020b. Learning for exception: Dynamic service caching in 5G-enabled MECs with bursty user demands. Proc of The 40th IEEE International Conference on Distributed Computing Systems (ICDCS) (2020).

[34] Zichuan Xu, Yutong Zhang, Weifa Liang, Qiufen Xia, Omer F Rana, Alex Galis, Guowei Wu, and Pan Zhou. 2019. NFV-Enabled Multicasting in Mobile Edge Clouds with Resource Sharing. (2019).

[35] Zichuan Xu, Zhiheng Zhang, Weifa Liang, Qiufen Xia, Omer Rana, and Guowei Wu. 2020c. QoS-Aware VNF Placement and Service Chaining for IoT Applications in Multi-Tier Mobile Edge Networks. ACM Trans. Sen. Netw. 16, 3, Article 23 (June 2020), 27 pages. DOI: http://dx.doi.org/10.1145/3387705

[36] Zichuan Xu, Lizhen Zhou, Sid Chau, Weifa Liang, Qiufen Xia, and Pan Zhou. 2020. Collaborate or separate? Distributed service caching in mobile edge clouds. Proc of The 39th IEEE International Conference on Computer Communications (INFOCOM) (2020).

[37] Jiwei Yang, Xu Shen, Xinmei Tian, Houqiang Li, Jianqiang Huang, and Xiansheng Hua. 2018. Local Convolutional Neural Networks for Person Re-Identification. (2018), 1074–1082.

[38] Hong-Xing Yu, Wei-Shi Zheng, Ancong Wu, Xiaowei Guo, Shaogang Gong, and Jian-Huang Lai. 2019. Unsupervised Person Re-identification by Soft Multilabel Learning. CoRR abs/1903.06325 (2019). http://arxiv.org/abs/1903.06325

[39] Xin Zhao, Liufang Sang, Guiguang Ding, Yuchen Guo, and Xiaoming Jin. 2018. Grouping Attribute Recognition for Pedestrian with Joint Recurrent Learning.. In IJCAI. 3177–3183.

[40] Xin Zhao, Liufang Sang, Guiguang Ding, Jungong Han, Na Di, and Chenggang Yan. 2019. Recurrent attention model for pedestrian attribute recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33. 9275–9282.

[41] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. 2015. Scalable person re-identification: A benchmark. In Proceedings of the IEEE international conference on computer vision. 1116–1124.

[42] Zhedong Zheng, Liang Zheng, Michael Garrett, Yi Yang, and Yi-Dong Shen. 2017b. Dual-Path Convolutional Image-Text Embedding. CoRR abs/1711.05535 (2017). http://arxiv.org/abs/1711.05535

[43] Zhedong Zheng, Liang Zheng, and Yi Yang. 2017a. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In Proceedings of the IEEE International Conference on Computer Vision. 3754–3762.