Multi-shot Pedestrian Re-identification via Sequential Decision Making

Jianfu Zhang[1], Naiyan Wang[2] and Liqing Zhang[1]

[1]Shanghai Jiao Tong University*, [2]TuSimple
c.sis@sjtu.edu.cn,winsty@gmail.com,zhang-lq@cs.sjtu.edu.cn

Abstract

Multi-shot pedestrian re-identification problem is at the core of surveillance video analysis. It matches two tracks of pedestrians from different cameras. In contrary to existing works that aggregate single frames features by time series model such as recurrent neural network, in this paper, we propose an interpretable reinforcement learning based approach to this problem. Particularly, we train an agent to verify a pair of images at each time. The agent could choose to output the result (same or different) or request another pair of images to verify (unsure). By this way, our model implicitly learns the difficulty of image pairs, and postpone the decision when the model does not accumulate enough evidence. Moreover, by adjusting the reward for unsure action, we can easily trade off between speed and accuracy. In three open benchmarks, our method are competitive with the state-of-the-art methods while only using 3% to 6% images. These promising results demonstrate that our method is favorable in both efficiency and performance.

1. Introduction

Pedestrian Re-identification (re-id) aims at matching pedestrians in different tracks from multiple cameras. It helps to recover the trajectory of a certain person in a broad area across different non-overlapping cameras. Thus, it is a fundamental task in a wide range of applications such as video surveillance for security and sports video analysis. The most popular setting for this task is single shot re-id, which judges whether two persons at different video frames are the same one. This setting has been extensively studied in recent years[7, 1, 16, 28, 17]. On the other hand, multi-shot re-id (or a more strict setting, video based re-id) is a more realistic setting in practice, however it is still at its early age compared with single shot re-id task.

Figure 1: Examples to demonstrate the motivation of our work. For most tracks, several even only one pair of images are enough to make confident prediction. However, in other hard cases, it is necessary to use more pairs to alleviate the influence of these samples of bad quality.

Currently, the main stream of solving multi-shot re-id task is first to extract features from single frames, and then aggregate these image level features. Consequently, the key lies in how to leverage the rich yet possibly redundant and noisy information resided in multiple frames to build track level features from image level features. A common choice is pooling[37] or bag of words[38]. Furthermore, if the input tracks are videos (namely, the temporal order of frames is preserved), optical flow[5] or recurrent neural network (RNN)[24, 39] are commonly adopted to utilize the motion cues. However, most of these methods have two main problems: the first one is that it is computationally inefficient to use all the frames in each track due to the redundancy. The second one is there could be noisy frames caused by occlusion, blur or incorrect detections. These noisy frames may significantly deteriorate the performance.

To solve the aforementioned problems, we formulate multi-shot re-id problem as a sequential decision making task. Intuitively, if the agent is confident enough about existing evidences, it could output the result immediately. Otherwise, it needs to ask for another pair to verify. To model such human like decision process, we feed a pair of ima-

*Representing Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Department of Computer Science and Engineering, Shanghai Jiao Tong University.

ges from the two tracks to a verification agent at each time step. Then, the agent could output one of three actions: *same*, *different* or *unsure*. By adjusting the rewards of these three actions, we could trade off between the number of images used and final accuracy. We depict several examples in Fig. 1. In case of easy examples, the agent could decide using only one pair of images, while when the cases are hard, the agent chooses to see more pairs to accumulate evidences. In contrast to previous works that explicitly deduplicate redundant frames[6] or distinguish high quality from low quality frames[21], our method could implicitly consider these factors in a data driven end-to-end manner. Moreover, our method is general enough to accommodate all single shot re-id methods as image level feature extractor even those non-deep learning based methods.

The main contributions of our work are listed as following:

- We are the first to introduce reinforcement learning into multi-shot re-id problem. We train an agent to either output results or request to see more samples. Thus, the agent could early stop or postpone the decision as needed. Thanks to this behavior, we could balance speed and accuracy by only adjusting the rewards.

- We verify the effectiveness and efficiency on three popular multi-shot re-id datasets. Along with the deliberately designed image feature extractor, our method could outperform the state-of-the-art methods while only using 3% to 6% images without resorting to other post-processing or additional metric learning methods.

- We empirically demonstrate that the Q function could implicitly indicate the difficulties of samples. This desirable property makes the results of our method more interpretable.

2. Related Work

Pedestrian re-identification for single still images has been explored extensively in these years. These researches mainly focused on two aspects: the first one is to extract features that are both invariant and discriminative from different viewpoints to overcome difficulties such as illumination changes, occlusions, blurs, etc. Representative works before deep learning age include [30, 14, 36]. However, these hand-crafted features are subverted by the rapidly developed Convolutional Neural Networks (CNN) in recent years. CNN has become *de facto* standard for feature extraction. The second aspect is metric learning. Metric learning embeds each sample into a latent space that preserves certain relationships of samples. Popular methods including Mahalanobis distance metric (RCA)[2], Locally Adaptive Decision Function (LADF)[18] and Large Margin Nearest Neighbor (LMNN)[31].

These two streams have met in the deep learning age: Numerous work focus on learning discriminative features by the guide of metric learning based loss funcions. The earliest work was proposed by Chopra *et al.* in [4]. They presented a Siamese architecture to learn similarity for face verification task with CNN. Schroff *et al.* proposed triplet loss in FaceNet [26] to learn discriminative embeddings by maximizing the relative distance between matched pairs and mismatched pairs. Inspired by these methods for face verification, deep learning methods for image based re-identification have also shown great progress in recent years[7, 16, 1]. Recently, [34, 35] utilized domain knowledge to improve performance: They incorporated pedestrian landmarks to handle body part misalignment problem. Concurrently, many deep learning based multi-task methods are proposed and reported promising performance. Wang *et al.* [28] proposed a joint learning framework by combining patch matching and metric learning. Li *et al.* [17] proposed a multi-loss model combining metric learning and global classification to discover both local and global features.

Compared with image based re-id task, multi-shot re-id problem is a more realistic setting, since the most popular application of re-id problem is surveillance video. It at least provides several representative frames after condensation, or even the entire videos are stored. Consequently, how to utilize such multi-frame information is at the core of multi-shot re-id task. Flow Energy Profile[19] is proposed to detect walking cycles with flow energy profile to extract spatial and temporal invariant features. In [38], Bag-of-words are adopted with learned frame-wised features to generate a global feature. Not surprisingly, deep learning also expressed its power in multi-shot re-id problem. A natural choice for temporal model in deep learning is Recurrent Neural Network (RNN). In the pioneering work [24], McLaughlin *et al.* first extracted features with CNN from images and then use RNN and temporal pooling to aggregate those features. Similarly, Chung *et al.* [5] presented a two stream Siamese network with RNN and temporal pooling for each stream. Recently, this idea was extended with spatial and temporal attention in [39, 33] to automatically pick out discriminative frames and integrate context information. Another interesting work is [21]. In [21], a CNN model learns the quality for each image, and then the video is aggregated with the image features weighted by the quality.

The goal of Reinforcement Learning (RL) is to learn policies based on trial and error in a dynamic environment. In contrast to traditional supervised learning, reinforcement learning trains an agent by maximizing the accumulated reward from environment. Additional to its traditional ap-

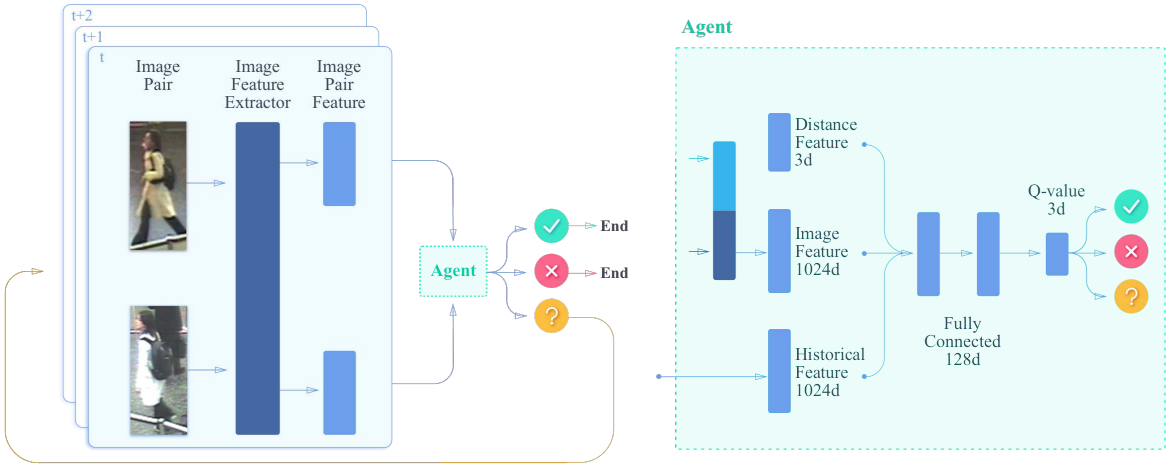

Figure 2: An illustration of our proposed method. Firstly we train an image level feature extractor (the left part) and then aggregate sequence level feature with an agent (the right part). The agent takes several kinds of features of one pair of images, and take one of three possible actions. If the taken action is "unsure", the above process is repeated again.

plications in control and robotics, recently RL has been successfully applied to a few computer vision tasks by treating them as a decision making process[3, 22, 10, 15, 12, 23]. Some closely related works include: In [10], the features for visual tracking problem are sorted by their costs, and then an agent is trained to decide whether current features are good enough to make accurate prediction. If not, it proceeds to the next feature. By this way, the agent saves unnecessary computation of expensive features. [12, 23] are two works which applied RL techniques to object detection task. In [12], the authors aimed to solve this task by limited budget which can be wall time, computing resources or etc. An agent is trained to learn a sequential policy for feature selection and stop before the cost budget is exhausted. While in [23] an agent is trained to learn whether to sample more image regions for better accuracy or stop the search. Our method shares the same spirit with these works, but tailored for multi-shot re-id problem.

3. Method

In this section, we will introduce our approach to multi-shot re-id problem. First, we will start with a formal formulation of this problem, and then present each component of our method. The overview of our method is depicted in Figure 2.

3.1. Formulation

In multi-shot re-id task, for each sequence in query identities, the goal is to rank all the gallery identities according to their similarities with the query identity. Given two sequences $(\mathcal{X}, \mathcal{Y}) = (\{x_1, \ldots, x_m\}, \{y_1, \ldots, y_n\})$, where x and y represent the images in $\mathcal{X}$ and $\mathcal{Y}$, respectively. Let $f(x)$ be a feature extractor that extracts discriminative features for each image x, and $g(\mathcal{X})$ be an aggregation function that aggregates image level features of $\mathcal{X}$ to sequence level feature. A similarity function $l(\cdot, \cdot)$ is designed to calculate the similarity between the query identity and gallery identity. According to the similarity computed by $l(\cdot, \cdot)$, we sort all the gallery identities for each query identity.

In the sequel, we will first present the details of our single image feature extractor $f(\cdot)$ in Sec. 3.2. It is built with a CNN trained with three different loss functions. Next, we elaborate our reinforcement learning based aggregation method $g(\cdot)$ and $l(\cdot, \cdot)$ in Sec. 3.3.

3.2. Image Level Feature Extraction

For single image feature extractor, a CNN is trained to embed an image into a latent space that preserves certain relationships of samples. To achieve this goal, we train a CNN with combination of three different kinds of loss functions: classification loss, pairwise verification loss [4] and triplet verification loss [26]. According to a recent work [32], multiple loss functions could better ensure the structure of the latent space and margins between samples. Particularly, we optimize large margin softmax loss[20] instead of softmax loss, since it demonstrates extraordinary performance in various classification and verification tasks.

Implementation details: We use two well-known network structures Inception-BN[11] and AlexNet [13] pre-trained on ILSVRC classification dataset[13] as base net-

works. We choose these two networks with different capacity and expression power to demonstrate the universality of our proposed aggregation method. In specific, we use the features from the last pooling layer as image level features. In training, we set the margin in triplet loss to 0.9. For large margin softmax, we set $\beta = 1000$, $\beta_{min} = 3$, and the margin as 3. For more details of these parameters, please refer to [20]. We optimize the network by momentum SGD optimizer with 320000 iterations. The learning rate is 0.01 and multiplied by 0.1 after 50000 and 75000 iterations, respectively.

As an important baseline, we simply use the average of l_2-normalized features from all the images as the feature for a sequence. Namely, the aggregation and similarity function is defined as:

$$g(\mathcal{X}) = \sum_{i}^{m} \frac{f(x_i)}{m}, \quad l(g(\mathcal{X}), g(\mathcal{Y})) = g(\mathcal{X}) \cdot g(\mathcal{Y}) \quad (1)$$

$\cdot$ representing inner product for two vectors. We rank all the gallery identities according to the value generated by $l(\cdot, \cdot)$.

3.3. Sequence Level Feature Aggregation

We formulate this problem as a Markov Decision Processes (MDP), described by $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R})$ as the states, actions, transitions and rewards. Each time step t, the agent will get a selected image pair from the two input sequences to observe a state $s_t \in \mathcal{S}$ and then choose an action $a_t \in \mathcal{A}$ from the experience it has learned. Next the agent will earn a reward $r_t \in \mathcal{R}$ from the environment in training. After that if the episode is not terminated, the agent will receive another image pair determined by state transition distribution $\mathcal{T}(s_{t+1}|s_t, a_t)$ and turn to the next state s_{t+1}. We will elaborate the details of them in the sequel.

Actions and Transitions: Initially, the agent is fed with an image pair selected from two selected sequences $\mathcal{X}$ and $\mathcal{Y}$. Note that we don't assume the order of the input and **randomly** form the pair from two sequences. We have three actions for the agent: *same*, *different* and *unsure*. The first two actions will terminate the current episode, and output the result immediately. We anticipate when the agent has collected enough information and is confident to make the decision, it stops early to avoid unnecessary computation. If the agent chooses to take action *unsure*, we will feed the agent another image pair.

Rewards: We define the rewards as follows:

1. $+1$, if a_t matches gt.
2. -1, if a_t differs from gt, or when $t = t_{max}$, a_t is still *unsure*.
3. r_p, if $t < t_{max}$, a_t is *unsure*.

Here t_{max} is defined as the maximum time step for each episode. gt is the ground truth. r_p is defined as a penalty (negative reward) or reward for the agent seeking for another image pair. If r_p is negative, it will be penalized for requesting more pairs; on the other hand, if r_p is positive, we encourage the agent to gather more pairs, and stop gathering when it has collected t_{max} pairs to avoid a penalty of -1. The value of r_p may strongly affect the agent's behavior. We will discuss its impact in Sec. 4.2.

States and Deep Q-learning: We use Deep Q-Learning [25] to find the optimal policy. For each state and action (s_t, a_t), $Q(s_t, a_t)$ represents the discounted accumulated rewards for the state and action. In training, we could iteratively update the Q function by:

$$Q(s_t, a_t) = r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}). \quad (2)$$

The state s_t for time step t in the episode consists of three parts. The first part is the observation o_t which is composed of the image features of current pair $(f(x), f(y))$ generated by the image feature extractor mentioned in Section 3.2, which is defined as $o_t = |f(x_t) - f(y_t)|$.The second part is a weighted average of the difference between historical image features of two sequences. This part makes the agent be aware of the previous image pairs it has already seen before. In specific, for each observation o_t the weight w_t is defined as:

$$w_t = 1.0 - \frac{e^{Q_u}}{e^{Q_s} + e^{Q_d} + e^{Q_u}} \quad (3)$$

where Q_u is short for $Q(s_t, a_t = unsure)$, and vice versa. The weight decreases as Q_u increases, as higher Q_u may indicate that current pair of images are hard to distinguish. The aggregated features should be affected as small as possible. As a result, h_t is the weighted average of the historical features for $t > 1$:

$$h_t = \frac{\sum_{i=1}^{t-1} w_i \times o_i}{\sum_{i=1}^{t-1} w_i}. \quad (4)$$

$h_t = o_t$ when $t = 1$. [1] Note that though the Q function is not specifically trained for sample weighting, it still reflects the importance of each frame. We leave end-to-end learning of the weights as our future work.

We also augment the image features with hand-crafted features for better discrimination. For each time step t, we calculate the distance $\|f(x_i) - f(y_j)\|_2^2$ for all $1 \leq i, j < t$,

[1] Note that since $h_t = 0$ implies $f(x) = f(y)$, it will introduce a strong bias to make the agent to choose "same" leading a poor performance if we set $h_t = 0$ when $t = 1$.

and then add the maximum, minimum and mean of them to the input, which results in 3 dimension extra features. [2]

The structure of the Q-network is shown in Fig.2. We simply use a two layer fully connected network as the Q function. Each fully connected layer has 128 outputs and is followed by an ReLU activation function.

Testing: For each query video sequences we play one episode and take the difference of the Q-value of action *same* and *different* at the terminal step as the final ranking score. Note that the Q-net essentially combines aggregation function $g(\cdot)$ and similarity function $l(\cdot, \cdot)$.

Implementation details: In training phase, for each episode we randomly choose positive or negative sequence pairs with ratio $1:1$. We feed the weighted historical features, features of current step and hand-crafted distance features into the Q-Net. The whole net along with the single image feature extractor is trained end-to-end except for fixing the first two stages of the base networks.

We train the Q-Net for 20 epochs by momentum SGD optimizer, 100000 iterations for each epoch. We use ϵ-greedy learning[27] as the exploration strategy and anneal ϵ linearly from 1 to 0.1 in the first 10 epochs. Learning rate is set to 0.0001, discount factor $\gamma = 0.9$ and batch size is 16. Experience replay is used and the memory buffer size is set to 5000. It takes 5.502 and 2.613 ms per episode for Inception-BN and Alexnet to verify a single pair of sequences on a Maxwell Titan X GPU. All these runtimes include the time of both image level feature extractor and Q-Net.

4. Experiments

In this section, we will present the results of our method on three open benchmarks, and compare it with other state-of-the-art methods. We will first introduce the datasets and evaluation metric used, and then present the ablation analyses of our method. After comparisons with other methods, we will also present some qualitative results to interpret the mechanism of our methods.

4.1. Evaluation Settings

We evaluate our algorithm with three most commonly used public datasets for multi-shot re-id problem: iLIDS-VID[29], PRID2011[9] and MARS[37]. For iLIDS-VID and PRID2011 dataset, following the setting in [24] we randomly split the dataset half-half for training and testing and average the results of 10 runs to make the evaluation stable. For MARS dataset, we follow the setting by the authors of the dataset. 625 identities are used for training, and the rest are used for testing. In testing, 1980 tracklets are preserved for query sets, while the rests are used as gallery sets.

To evaluate performance for each algorithm, we report the Cumulative Matching Characteristic (CMC) metric. It represents the expectation of the true matching hits in the first top-n ranking. Here we use $n \in \{1, 5, 10, 20\}$ in the evaluations.

4.2. Ablation Studies

Before comparing our models with previous works, we first conduct ablation studies of some important factors of our method. The results are listed in Table 1 and Table 2 for different settings and datasets. As a baseline, we calculate the averagely pooled features mentioned in Equation 1. The results of baseline method using all frames are listed in **All frames** rows.

First let's discuss an important parameter of our model: the reward for unsure action r_p. We show the statistics of how many images are used (which is double of the time steps) in each episode in Fig. 3 and corresponding CMC rank 1 in Table 1 and Table 2. When r_p is small (negative), the agent will stop early and verify the identities with fewer images. When r_p is big (positive), the agent will be encouraged to be more cautious, requesting more image pairs for better performance. This will help the agent postpone its decision to avoid mistakes caused by imperfect quality like occlusions. Among all different values of r_p, we found that $r_p = 0.2$ gives us the most remarkable performance.

We compare the CMC Rank 1 results of our proposed models with baseline methods in Figure 4. The dashed green line denotes the **All frames** setting in Table 1 and Table 2, while the blue stars denotes the setting that we randomly sample pairs from the tracks, and then averagely pool their features to a track level feature. We vary the number of images sampled to generate the curve. And the yellow squares show the CMC Rank 1 performance of our model with different values of r_p. We then take a close look of the analysis of the number of images used in these two networks. Not surprisingly, our method uses notably less number of images. Particularly, we can outperform the **All frames** baselines using only 3% to 4% images. We owe the reason to that the average pooling of all the frames may be easily contaminated by some imperfect frames.

In Figure 4, we also compare the CMC Rank 1 results of our model with different choices of the maximum time step t_{max}. We take three different choices: $t_{max} = 4$ (red triangles), $t_{max} = 8$ (yellow squares) and $t_{max} = 16$ (seafoam blue pentagons) and see how CMC Rank 1 changes with different values of r_p. Comparing among three settings, we find that $t_{max} = 8$ gives the best trade-off between number of images used and performance.

Next, we compare across different datasets. There are tons of occlusions in iLIDS-VID and MARS datasets. Mo-

[2] Here we don't make time step t as a part of the state-space. Since the feature extractor fits better in the training set, the agent uses less time steps to verify samples in training set compared with that in testing set, adding t to the state-space will cause overfitting issues.

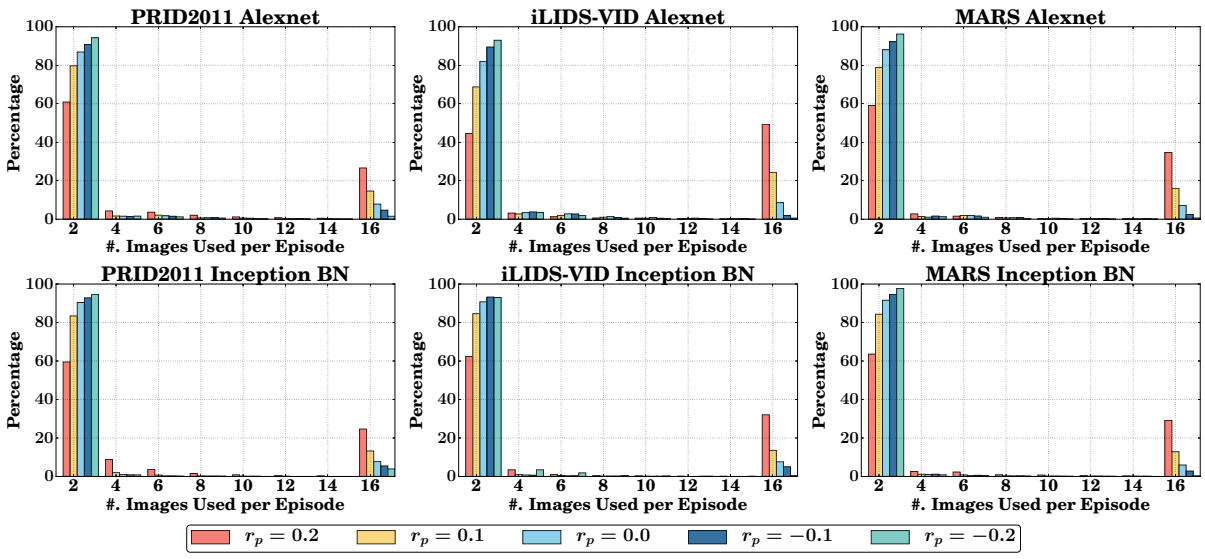

Figure 3: Statistics of the number of images used in each episode of our model with different reward for action *unsure*.

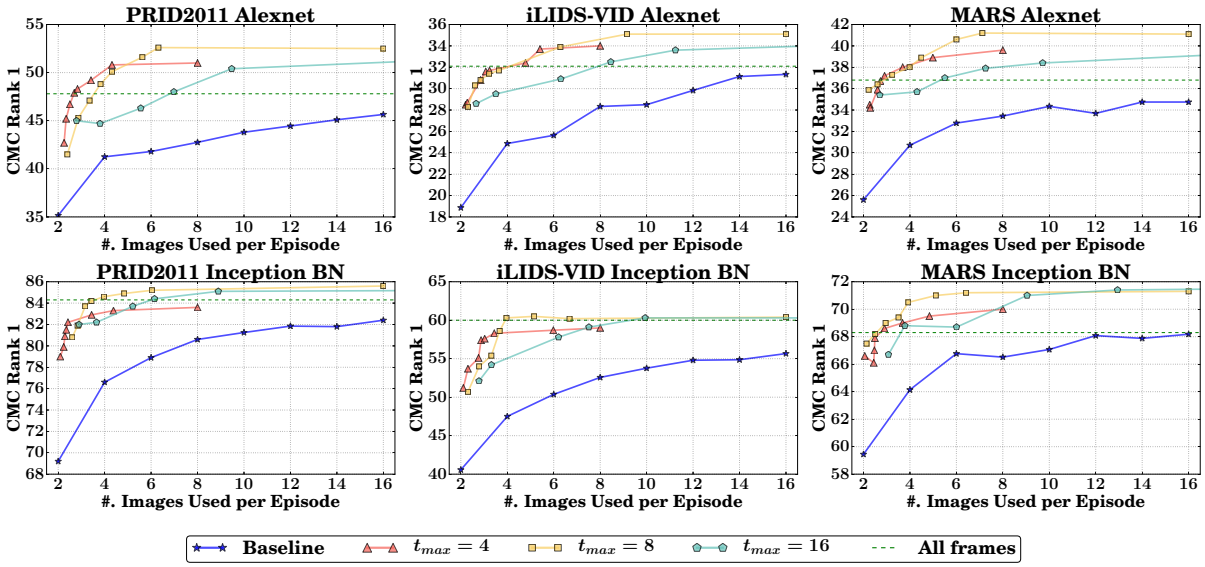

Figure 4: CMC Rank 1 results for our model compared with baseline.

reover, there are many mislabeled samples in MARS since the bounding boxes of MARS dataset are machine generated. PRID2011 dataset is much easier compared with the other two datasets. We find that the agent tends to ask for more images in iLIDS-VID and MARS dataset than PRID2011 dataset under the same setting. These two findings coincide with our anticipated behavior of the agent.

Finally there are some more settings worthy trying. We put these experiment results in Table 1 and Table 2 with $r_p = 0.2$ and $t_{max} = 8$ if not specially mentioned.

- **No handcrafted features**: We learn the policy without the 3 dimensions handcrafted distance features, only with image level features and historical information. CMC Rank 1 drops a lot and the agent will tend to make a quicker choice.

- **DRQN**: We try to replace the last fc layer with a LSTM layer as in [8] to gather historical features instead of the method we described in 3.2. The results are worse compared with our proposed method.

Dataset	PRID2011		iLIDS-VID		MARS	
Settings	CMC1	#.of Images	CMC1	#.of Images	CMC1	#.of Images
All frames	84.3	200.000	60.0	146.000	68.3	111.838
$r_p = 0.2$	**85.2**	6.035	60.2	6.681	**71.2**	6.417
$r_p = 0.1$	84.6	3.970	**60.3**	3.966	70.5	3.931
$r_p = 0$	83.7	3.162	55.4	3.134	69.0	2.952
$r_p = -0.1$	81.9	2.835	54.0	2.789	68.2	2.507
$r_p = -0.2$	80.8	**2.605**	50.7	**2.307**	67.5	**2.130**
No handcrafted features	83.5	5.679	57.8	5.934	69.2	6.103
DRQN	83.2	4.314	59.8	5.109	69.9	4.577
Sequential	84.1	7.549	59.7	7.021	70.5	6.591
Video fine-tune	84.7	16.000	60.2	16.000	70.7	16.000

Table 1: Test results for our model based on Inception BN image feature extractor.

Dataset	PRID2011		iLIDS-VID		MARS	
Settings	CMC1	#.of Images	CMC1	#.of Images	CMC1	#.of Images
All frames	47.8	200.000	32.1	146.000	36.8	111.838
$r_p = 0.2$	**52.6**	6.316	**35.1**	9.154	**41.2**	7.119
$r_p = 0.1$	50.1	4.317	33.3	5.722	38.9	4.491
$r_p = 0$	47.1	3.349	31.7	3.637	37.3	3.238
$r_p = -0.1$	45.3	2.870	30.3	2.614	36.4	2.604
$r_p = -0.2$	41.5	**2.394**	28.3	**2.307**	35.9	**2.221**
No handcrafted features	48.2	5.931	32.4	7.793	37.3	6.645
DRQN	48.7	3.291	33.0	6.119	40.2	5.716
Sequential	51.4	7.834	34.1	9.318	40.8	7.423
Video fine-tune	50.3	16.000	32.7	16.000	40.0	16.000

Table 2: Test results for our model based on Alexnet image feature extractor.

Dataset	PRID2011				iLIDS-VID				MARS			
CMC Rank	1	5	10	20	1	5	10	20	1	5	10	20
RNN-CNN[24]	70	90	95	97	58	**87**	91	96	40	64	70	77
ASTPN[33]	77	95	**99**	99	**62**	86	**94**	**98**	44	70	74	81
Two-Stream[5]	78	94	97	99	60	86	93	97	-	-	-	-
CNN+XQDA[38]	77.9	93.5	-	99.3	53.0	81.4	-	95.1	65.3	82.0	-	89.0
Alexnet (All frames)	47.8	74.4	83.6	91.2	32.1	59.0	70.0	80.6	36.8	53.1	61.6	68.8
Alexnet + Ours	52.6	81.3	88.4	96.3	35.1	61.3	72.1	84.0	41.2	55.6	63.1	73.3
Inception-BN (All frames)	84.3	96.5	98.8	**99.7**	60.0	85.4	92.0	96.3	68.3	83.5	88.0	90.8
Inception-BN + Ours	**85.2**	**97.1**	98.9	99.6	60.2	84.7	91.7	95.2	**71.2**	**85.7**	**91.8**	**94.3**
QAN[21]	90.3	98.2	99.3	100	68.0	86.8	95.4	97.4	-	-	-	-
STRN[39]	79.4	94.4	-	99.3	55.2	86.5	-	97.0	70.6	90.0	-	97.6

Table 3: Comparisons with other state-of-the-art methods. Please note that the results in last two rows are not directly comparable due to different setting. For more details, please refer to the text.

- **Sequential**: Instead of feeding the agent with random ordered images, we try to provide the images sequentially started from the beginning of the sequences. The results are worse compared with random order.
- **Video fine-tune**: Here we randomly sample 8 images from each sequence, averagely pool the features and use this sequence level feature to fine-tune the CNN as described in Sec. 3.2. This model gets a slightly worse CMC Rank 1 performance, but uses more images.

4.3. Comparisons with State-of-the-art Methods

Table 3 summarizes the CMC results of our model and other state-of-the-art multi-shot re-id methods. Here we use the setting of $r_p = 0.2$ since this setting is the most accu-

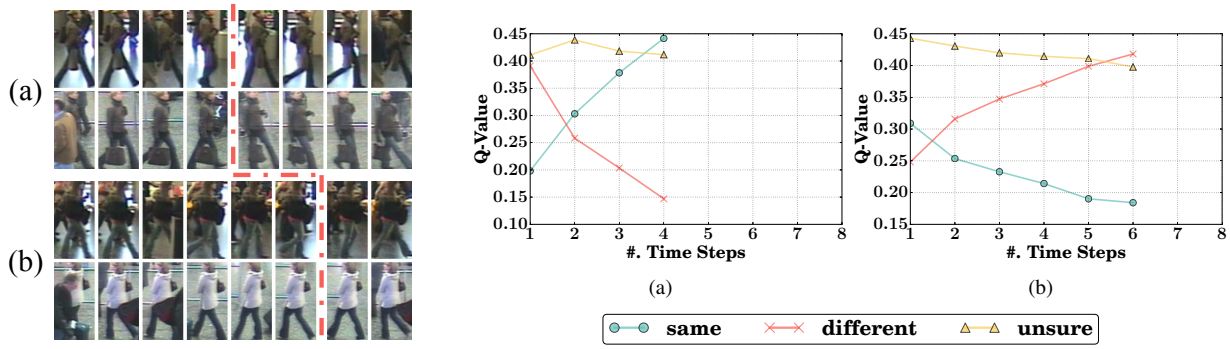

Figure 5: Some example episodes generated by our model. All the sampled images for each identity are listed on the left with a red dashed line splits used images and unused images. On the right side, normalized Q values for each example are shown.

rate according to the evaluations in previous section. CNN-RNN[24], ASTPN[33], STRN[39] and Two-Stream[5] are four different methods based on RNN time series model and more advanced attention mechanism. CNN-XQDA[38] and QAN[21] train discriminative embeddings of images and apply different pooling methods. Among them, CNN-RNN[24], ASTPN[33] and Two-Stream[5] use both image and explicit motion features (optical flow) as inputs for deep neural network.

Here QAN[21] uses their own extra data for training. STRN[39] uses MARS pre-trained model to train PRID2011 and iLIDS-VID. Therefore, their methods cannot be fairly compared with other methods. We just list their results for reference.

For PRID2011 dataset, our method outperforms all other methods, improves the CMC Rank 1 about 5% compared with best state-of-the-art methods. For iLIDS-VID and MARS dataset, our results are at least comparable or even better than the compared methods. For CMC Rank 5, 10 and 20, the trends are similar to Rank 1.

Note that all the other methods use all images for verification. *Our proposed model uses only 3% to 6% images for each track pairs on average to obtain these encouraging performance.*

4.4. Qualitative Results

In Figure 5, two representative episodes are shown. We can see the change of the Q values for the agent in dynamic environment. Softmax function is applied to normalize the Q values. (a) shows an example episode with the same person, while (b) shows one with different persons. These two episodes end with different length. Severe occlusions happen in the early pairs of (a) and (b). After the occlusions disappear, the agent gradually collects information and corrects its decisions. After fed with several image pairs of better quality, the agent is confident enough to make the correct choices eventually.

5. Conclusion

In this paper we have introduced a novel approach for multi-shot pedestrian re-identification problem by casting it as a pair by pair decision making process. Thanks to reinforcement learning, we could train an agent for such task. Specifically, it receives image pairs sequentially, and output one of the three actions: *same, different* or *unsure*. By early stop or decision postponing, the agent could adjust the budget needs to make confident decision according to the difficulties of the tracks.

We have tested our method on three different multi-shot pedestrian re-id datasets. Experimental results have shown our model can yield competitive or even better results with state-of-the-art methods using only 3% to 6% of images. Furthermore, the Q values outputted by the agent is a good indicator of the difficulty of image pairs, which makes our decision process is more interpretable.

Currently, the weight for each frame is determined by the Q value heuristically, which means the weight is not guided fully by the final objective function. More advanced mechanism such as attention can be easily incorporated into our framework. We leave this as our future work.

6. Acknowledgement

The work was supported in part by the National Basic Research Program of China (Grant No. 2015CB856004), the Key Basic Research Program of Shanghai Municipality, China (15JC1400103,16JC1402800)

References

[1] E. Ahmed, M. J. Jones, and T. K. Marks. An improved deep learning architecture for person re-identification. In *CVPR*, 2015. 1, 2

[2] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a Mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, 6:937–965, 2005. 2

[3] J. C. Caicedo and S. Lazebnik. Active object localization with deep reinforcement learning. In *ICCV*, 2015. 3

[4] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005. 2, 3

[5] D. Chung, K. Tahboub, and E. J. Delp. A two stream Siamese convolutional neural network for person re-identification. In *ICCV*, 2017. 1, 2, 7, 8

[6] A. Das, R. Panda, and A. K. Roy-Chowdhury. Continuous adaptation of multi-camera person identification models through sparse non-redundant representative selection. *Computer Vision and Image Understanding*, 156:66–78, 2017. 2

[7] S. Ding, L. Lin, G. Wang, and H. Chao. Deep feature learning with relative distance comparison for person re-identification. *Pattern Recognition*, 48(10):2993–3003, 2015. 1, 2

[8] M. J. Hausknecht and P. Stone. Deep recurrent Q-Learning for partially observable MDPs. In *AAAI*, 2015. 6

[9] M. Hirzer, C. Beleznai, P. M. Roth, and H. Bischof. Person re-identification by descriptive and discriminative classification. In *SCIA*, 2011. 5

[10] C. Huang, S. Lucey, and D. Ramanan. Learning policies for adaptive tracking with deep feature cascades. In *ICCV*, 2017. 3

[11] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 3

[12] S. Karayev, M. Fritz, and T. Darrell. Anytime recognition of objects and scenes. In *CVPR*, 2014. 3

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 3

[14] I. Kviatkovsky, A. Adam, and E. Rivlin. Color invariants for person reidentification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1622–1634, 2013. 2

[15] X. Lan, H. Wang, S. Gong, and X. Zhu. Identity alignment by noisy pixel removal. In *BMVC*, 2017. 3

[16] W. Li, R. Zhao, T. Xiao, and X. Wang. DeepReID: Deep filter pairing neural network for person re-identification. In *CVPR*, 2014. 1, 2

[17] W. Li, X. Zhu, and S. Gong. Person re-identification by deep joint learning of multi-loss classification. In *IJCAI*, 2017. 1, 2

[18] Z. Li, S. Chang, F. Liang, T. S. Huang, L. Cao, and J. R. Smith. Learning locally-adaptive decision functions for person verification. In *CVPR*, 2013. 2

[19] K. Liu, B. Ma, W. Zhang, and R. Huang. A spatio-temporal appearance representation for viceo-based pedestrian re-identification. In *ICCV*, 2015. 2

[20] W. Liu, Y. Wen, Z. Yu, and M. Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, 2016. 3, 4

[21] Y. Liu, J. Yan, and W. Ouyang. Quality aware network for set to set recognition. In *CVPR*, 2017. 2, 7, 8

[22] M. Malmir, K. Sikka, D. Forster, I. R. Fasel, J. R. Movellan, and G. W. Cottrell. Deep active object recognition by joint label and action prediction. *Computer Vision and Image Understanding*, 156:128–137, 2017. 3

[23] S. Mathe, A. Pirinen, and C. Sminchisescu. Reinforcement learning for visual object detection. In *CVPR*, 2016. 3

[24] N. McLaughlin, J. M. del Rincón, and P. C. Miller. Recurrent convolutional network for video-based person re-identification. In *CVPR*, 2016. 1, 2, 5, 7, 8

[25] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. A. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. 4

[26] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 2, 3

[27] R. S. Sutton and A. G. Barto. Reinforcement learning: An introduction. *IEEE Transactions on Neural Networks*, 9(5):1054–1054, 1998. 5

[28] F. Wang, W. Zuo, L. Lin, D. Zhang, and L. Zhang. Joint learning of single-image and cross-image representations for person re-identification. In *CVPR*, 2016. 1, 2

[29] T. Wang, S. Gong, X. Zhu, and S. Wang. Person re-identification by video ranking. In *ECCV*, 2014. 5

[30] X. Wang, G. Doretto, T. Sebastian, J. Rittscher, and P. H. Tu. Shape and appearance context modeling. In *ICCV*, 2007. 2

[31] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009. 2

[32] T. Xiao, H. Li, W. Ouyang, and X. Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *CVPR*, 2016. 3

[33] S. Xu, Y. Cheng, K. Gu, Y. Yang, S. Chang, and P. Zhou. Jointly attentive spatial-temporal pooling networks for video-based person re-identification. In *ICCV*, 2017. 2, 7, 8

[34] H. Zhao, M. Tian, S. Sun, J. Shao, J. Yan, S. Yi, X. Wang, and X. Tang. Spindle net: Person re-identification with human body region guided feature decomposition and fusion. In *CVPR*, 2017. 2

[35] L. Zhao, X. Li, Y. Zhuang, and J. Wang. Deeply-learned part-aligned representations for person re-identification. In *ICCV*, 2017. 2

[36] R. Zhao, W. Ouyang, and X. Wang. Learning mid-level filters for person re-identification. In *CVPR*, 2014. 2

[37] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian. MARS: A video benchmark for large-scale person re-identification. In *ECCV*, 2016. 1, 5

[38] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *ICCV*, 2015. 1, 2, 7, 8

[39] Z. Zhou, Y. Huang, W. Wang, L. Wang, and T. Tan. See the forest for the trees: Joint spatial and temporal recurrent neural networks for video-based person re-identification. In *CVPR*, 2017. 1, 2, 7, 8