Attention-aware Deep Reinforcement Learning for Video Face Recognition

Yongming Rao[1,2,3], Jiwen Lu[1,2,3]*, Jie Zhou[1,2,3]
[1]Department of Automation, Tsinghua University, Beijing, China
[2]State Key Lab of Intelligent Technologies and Systems, Beijing, China
[3]Tsinghua National Laboratory for Information Science and Technology (TNList), Beijing, China
raoyongming95@gmail.com; {lujiwen,jzhou}@tsinghua.edu.cn

Abstract

In this paper, we propose an attention-aware deep reinforcement learning (ADRL) method for video face recognition, which aims to discard the misleading and confounding frames and find the focuses of attentions in face videos for person recognition. We formulate the process of finding the attentions of videos as a Markov decision process and train the attention model through a deep reinforcement learning framework without using extra labels. Unlike existing attention models, our method takes information from both the image space and the feature space as the input to make better use of face information that is discarded in the feature learning process. Besides, our approach is attention-aware, which seeks different attentions of videos for the recognition of different pairs of videos. Our approach achieves very competitive video face recognition performance on three widely used video face datasets.

1. Introduction

Video face recognition has attracted great attention in computer vision over the past few years [4, 7, 8, 15, 24, 31, 32, 40, 41, 43]. There are many practical applications for video face recognition such as access control, video search and visual surveillance. Compared to still face recognition, videos can capture human faces from multiple views, which provide more useful information of a single face. However, video faces usually suffer from uncontrolled variations of poses, illuminations and *etc*., which leads to large intra-class distances. Hence, it is desirable to design a model to integrate information across frames and reduce intra-class distances for effective and robust video face recognition.

There have been a variety of studies on how to effectively integrate information across frames for video face representation [6, 18, 21, 28, 43]. These methods exploit video information from all frames, which is usually considered as equal importance. However, some features are misleading and confounding so that low quality frames may harm the performance of recognition. To address this, Yang *et al.* [43] proposed an attention-based method to find the weights of features by using the information from features themselves. However, the information of image quality is reduced in the feature learning process [40], where information from the feature space is not reliable enough to find the most important parts (precise focuses of attention) in videos.

In this work, we propose a new approach by introducing the Markov decision process (MDP) [3] to remove these misleading and confounding frames step by step with the

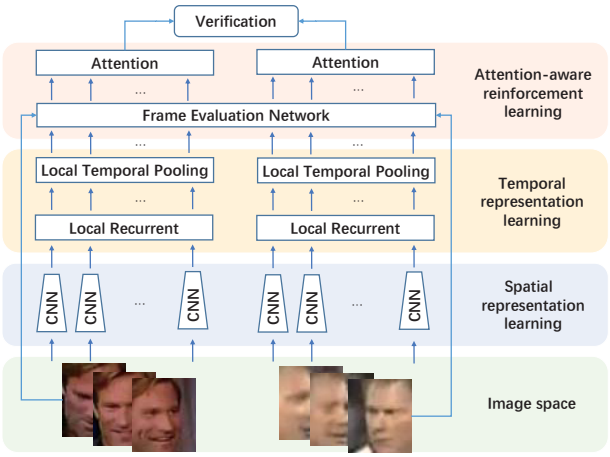


Figure 1. Flow-chart of our proposed method for video face recognition. Our approach takes a pair of face videos as the input and produces the temporal-spatial representations for each frame by using multiple stacked modules, including a convolutional neural network (CNN), a recurrent layer and a pooling layer with locality constraints, respectively. Then, a hard attention model with a frame evaluation network is trained by the proposed deep reinforcement learning method, which finds the attentions of the video pair for face verification.

*Corresponding author.

deep reinforcement learning method [30]. Instead of learning attentions only from the feature space, we compute the representation of videos by using the information from both the feature space and the image space. Our attention model is attention-aware because we take a pair of face videos instead of a single video as the input of the attention model because different situations may lead to different attentions in the recognition task. Motivated by the fact that convolutional neural network (CNN) has achieved state-of-the-art results on face recognition in recent years [31, 32, 35], we propose a local temporal representation for video face recognition by combining the CNN feature with recurrent layers with locality constraints to make better use of temporal information. Figure 1 shows the flow-chart of our proposed approach. Experimental results on three video face datasets show the effectiveness of the proposed ADRL.

2. Related Work

Video Face Recognition: Most existing video face recognition methods [6, 17, 18, 19, 20, 21, 26, 27, 28, 38, 43] usually consider each video as an image set and employ image set matching methods for video face recognition. These methods can be categorized into two classes: manifold-based and instance-based. For the first category, each video or image set is modeled as a manifold, and the similarity or distance between each video pair is computed by measuring the distance between manifolds. In previous works, many models have been used for manifold modeling such as affine hull [6], SPD models [18, 20], Grassmann manifolds [21] and n-order statistics [27, 38]. In these methods, image frames are considered of equal importance. When the number of low quality image frames increases, these models are easily misled. For the second category, each video or image set is modeled as a set of instances, which aims to exploit the relationship between instances in videos in the learning and recognition process. For example, Lu *et al.* trained a parametric model for discriminative representations of instances in each image set [26]. Sivic *et al.* employ a simple and effective thresholding method to get reliable features from video frames [33]. Yang *et al.* proposed an attention-based model to aggregate features of video frames [43]. Unlike these methods, our attention model is trained by using the information from both the manifold and instance levels, which are the representations of the whole video and single frames, respectively.

Deep Reinforcement Learning: Reinforcement learning has been originated from our understanding of humans' decision making process [25], which aims to enable the agent to decide the behavior from its experiences. Unlike conventional supervised machine learning methods, reinforcement learning is supervised through the reward signals of actions. Deep reinforcement learning [30] is a combination of deep learning and reinforcement learning, which has been used in various applications in recent years. For examples, Mnih *et al.* combined reinforcement learning with CNN and achieved the human-level performance in the Atari game [30]. Caicedo *et al.* introduced reinforcement learning for active object localization [5]. Zhang *et al.* employed reinforcement learning for vision control in robotics [45]. However, little progress has been made in reinforcement learning for visual recognition, especially in face recognition.

3. Proposed Approach

Figure 1 illustrates the flow-chart of our proposed approach. Our framework is composed of two parts: feature learning and attention learning. The feature learning part is a network which takes an entire video as the input. The network processes the whole video with a deep CNN model, a recurrent layer, and a temporal pooling layer to produce temporal representations of each frame in the video, respectively. The attention part is a frame evaluation network, which is designed to produce the values of frames. The values are used to find the most representative frames, which are the attentions of the video. In our work, we formulate the process of finding attentions in video pairs as a Markov decision process (MDP) and introduce a reinforcement learning method to train the evaluation network. The input information of the frame evaluation network comes from both the image space and the feature space. Moreover, we take the mutual relationship between both videos into the state evaluation of the MDP.

3.1. Temporal Representation Learning

Relationship between frames provides important hints for face recognition, and also extracts robust descriptors. Instead of taking the whole video into the recurrent layer as the input, we introduce a more flexible local bi-directional recurrent layer and a local temporal-pooling layer, which combine a few neighboring frames into a temporal representation from both directions and consider other frames as irrelevances.

Assume the video A containing N^A frames $X^A = [x_1^A, x_2^A, ..., x_{N^A}^A]$, $C_1(x)$ is a CNN feature representation, each frame x_i^A has a corresponding convolutional feature representation $f_i^A = C_1(x_i^A)$. We employ the widely used long short-term memory (LSTM) as the recurrent layer and mean-pooling strategy to combine features, so that the temporal representation of frame x_i^A becomes

$$h_i^A = \frac{1}{1+2r} \sum_{k=i-r}^{i+r} m_k^A, \tag{1}$$

and

$$[m_{i-r}^A, ..., m_i^A, ..., m_{i+r}^A] = \mathcal{R}([f_{i-r}^A, ..., f_i^A, ..., f_{i+r}^A]),$$

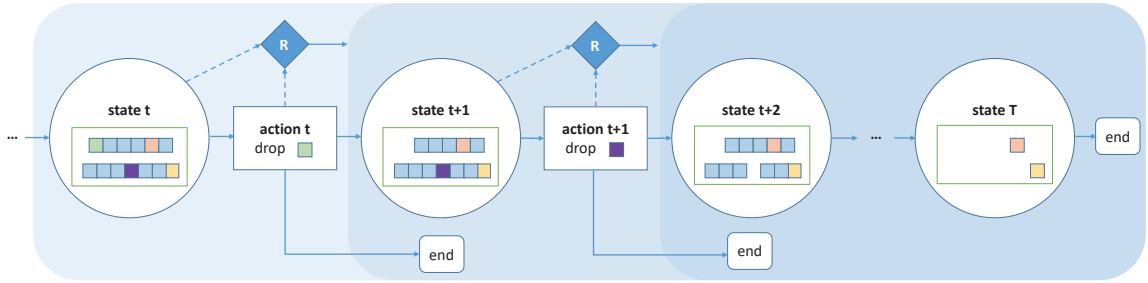


Figure 2. Markov decision process (MDP) of finding the focuses of attentions. States represent remaining frames after t steps, actions represent the decisions of dropping frames. Action a_t may lead to two states: state s_{t+1} and termination. Reward signal (R) is decided by the face recognition network C_1 depending on states and actions. States, actions, reward signals and terminations in MDP are illustrated by circles, rectangles, rhombuses and rounded rectangles, respectively.

where $\mathcal{R}$ is a bi-directional LSTM [11] that takes a sequence of features as the input and produces a sequence of activations $[m^A_{i-r}, ..., m^A_i, ..., m^A_{i+r}]$, r is the range of neighboring frames, h^A_i is the corresponding temporal representation. In the remaining text, we use C_1 to represent the CNN and temporal layers.

For each video, we extract the feature representations of each frame using our feature representation network, where the representation is a combination of the single frame feature and the inter-frame feature. In practice, the CNN model and the recurrent layer are trained separately. In other words, we employ the CNN model developed for still face recognition as convolutional feature extractor because sufficient labeled training samples can be used to train the model.

3.2. Attention-aware Deep Reinforcement Learning

Face frames in videos are often of large variations in pose, illumination, expression and image quality. Hence, not all image frames in a video are helpful for recognition. In other words, some frames are valueless. It is desirable to consider using a subset of frames from each video to measuring the distance between videos to avoid the adverse effect from low-quality image frames.

Attention models have been widely used in various computer vision applications [29, 42]. In our approach, we consider the process of finding the focuses of attentions as the process of finding the most representative frames from video pairs. In previous works, a video is usually modeled as a manifold and the distance metric between manifolds is utilized to compute the distance between two videos. By introducing the attention model, we redefine the distance metric between manifolds as the distance between the attentions of videos:

$$\text{distance}(X^A, X^B) = \text{distance}(p^A, p^B),$$

where p^A and p^B are decided by a hard attention model:

$$p^A = \frac{\sum_{i=1}^{N^B} a^A_i h^A_i}{\sum_{i=1}^{N^B} a^A_i}, \quad p^B = \frac{\sum_{i=1}^{N^B} a^B_i h^B_i}{\sum_{i=1}^{N^B} a^B_i}, \tag{2}$$

$$a^A_i, a^B_i \in \{0, 1\}$$

where a^A_i and a^B_i are the weights of attentions suggesting whether corresponding frames are the focuses of attentions.

In many previous works [1, 9, 43], the weights of attentions are obtained by using the relationship between feature vectors. However, it is not appropriate for face recognition because one of the goals for feature learning is to minimize the intra-class distances [40], which aims to reduce the influence brought by frame situations like poses and expressions. Therefore, our attention model takes the information from frames directly to guarantee that we can find the precise attentions of videos. Moreover, the focuses of videos should be different for different videos pairs, so that a_i should be decided by information from both videos in our framework. Hence, we propose a deep model $Q_i = C_2(I_i, M_i)$ to evaluate the frame situation of x_i and decide the weights of attentions, which takes both information I_i from the image space and M_i from the feature space as the input and produces the value Q_i. To train C_2 without additional supervision, we consider C_1 as an *expert* in face recognition, and design an algorithm to *teach* C_2 by C_1 as the recognition performance of C_1 indicates the qualities of input frames.

There are two strategies for human to find the most representative frames from two videos. One is to directly grade each frame and then decide the most representative frames from each video. The other is to remove the worst frame step by step, and the remaining frames are the most representative ones. For the first strategy, it is difficult to choose a good value for Q_i and train C_2 with no extra labels. It is obvious that finding the worst frame among two videos is much easier than directly finding the most important parts.

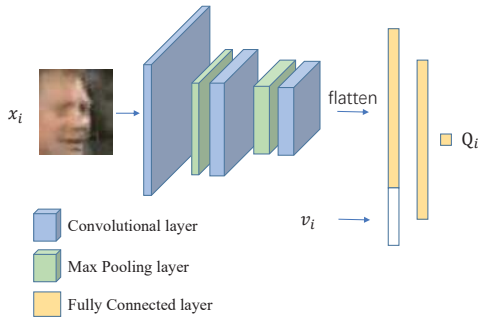


Figure 3. The architecture of frame evaluation network. The kernel sizes of convolution layers are 9×9, 4×4 and 3×3, respectively. Both the max pooling layers have kernel size 2×2 and stride 2. The feature dimension of the first two fully connected layers are set as 64. All hidden layers use PReLU [13] as the activation functions.

Therefore, we adopt the second strategy and formulate the process of finding attentions as a Markov decision process.

We denote the remaining frames after t times dropping as state s_t, the action of dropping frame as a_t. Dropping a frame may lead to two states: s_{t+1} and termination, where termination means that we already find a set of the most representative frames or there is only a pair of frames from each video. Evaluative feedback from environment r_i for (s_t, a_t) is decided by *expert* C_1. In practice, we use the cosine similarity computed by mean-pooling features as the metric for two different videos

$$S(X^A, X^B|s_t) = \cos(p^A|s_t, p^B|s_t), \tag{3}$$

where $p^A|s_t$ and $p^B|s_t$ are the mean-pooling of remaining feature vectors of video A and B at state s_t, respectively, which are formulated as:

$$a_i^A|s_t = \begin{cases} 1 & x_i^A \text{ remains at } s_t, \\ 0 & x_i^A \text{ has been dropped at } s_t. \end{cases} \tag{4}$$

We define r_i as the improvement in verification brought from action a_t at state s_t

$$r_t = l_{AB}(S(X^A, X^B|s_{t+1}) - S(X^A, X^B|s_t)). \tag{5}$$

where l_{AB} is the label either 1 or -1 denoting positive or negative pairs. Termination criteria defined by r_t is

$$r(s_t, a_t) < 0, \forall a_t. \tag{6}$$

The Markov decision process of finding attentions is shown in Figure 2.

The key step of the Markov decision model is to decide the best action at certain states (the decision policy). By introducing the Q-learning method [30, 39], we define Q_i as the expectation value of action a_t at s_t, where the action a_t drops the frame x_i. The policy is defined as $\pi = \text{argmax}_{a_i} Q_i$. Therefore, Q_i can be rewritten as

$$Q_i = Q(s_t, a_t) = \max_{\pi} \mathbb{E}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + ...|\pi]. \tag{7}$$

where γ is the discount factor in Q-learning, which takes a trade-off between the immediate reward and the prediction of feature reward. As introduced by [30], we employ a deep neural network to estimate $Q^*(s_t, a_t) = C_2(s_t, a_t)$. If the estimation is good enough, we can regard $Q^*(s_t, a_t)$ as $Q(s_t, a_t)$.

There are two ways to design the architecture of the deep Q-network (DQN) C_2, one is taking the state s_t as the input and producing the Q-value of all possible actions, which is used in [30]. The other is taking both the state s_t and the action a_t as the input and producing single Q-value of the action.

During the process of frame dropping, the number of possible actions is changing (decreasing) and it is difficult to describe the state s_i as the input of neural network while the action a_i is clearer as the frame will be dropped. In order to attenuate the effect brought from our description of states, we adopt the second architecture as the frame evaluation network. As shown in Figure 3, our frame evaluation network takes the dropping frame x_i and a vector v_i which describe the geometry relationship among the dropping frames and two videos in the feature space. Frame evaluation network firstly represents x_i as a deep feature by a convolutional network, and then concatenates the feature with vector v_i and takes the concatenated feature into a fully connected network to produce $Q(s_t, a_t)$. In practice, v_i is composed of 4 parts which are the 1-order and 2-order statistics of each videos respectively. For video A, the 1-order statistic feature is computed as

$$v_{\text{1-order}}^A = \tanh(W_1(p^A|s_t - h_a) + b_1) \tag{8}$$

where h_a is the feature vector of the dropping frame.

The 2-order statistic feature is computed as

$$v_{\text{2-order}}^A = \tanh(W_2\sigma^A|s_t + b_2) \tag{9}$$

where $\sigma^A|s_t$ is the variance of features of remaining frames in each dimension. Weights W_1, W_2, b_1 and b_2 are shared for all videos.

According to the Bellman equation [2], we apply the Q-learning method to train the frame evaluation network by

$$\min_{\theta} \mathcal{H} = \mathbb{E}[r(s_i, a_i) + \gamma \max_{a_{i+1}} Q(s_{i+1}, a_{i+1}) - Q(s_i, a_i)]^2, \tag{10}$$

where θ is the set of weights in frame evaluation network C_2. In our proposed framework, we create any states as the training data, so that we constitute mini-batch by random

Algorithm 1 ADRL

Input: Temporal representation $\{h_i^A\}$ of training set $\{X^A\}$, list of labeled video pairs $list = \{(A, B, l_{AB})\}$
Output: Weights of frame evaluation network θ

initialize θ with small random values
for $i \leftarrow 1, 2, ..., M$ **do**
  Sample random minibatch $\{(A, B, l_{AB})\}$ from $list$
  Create random states and ϵ-greedy actions $\{s_t, a_t\}$
  Compute corresponding rewards $\{r_t\}$ with $\{h_i^A\}$
  for s_t, a_t in $\{s_t, a_t\}$ **do**
    if termination **then**
      $y_t \leftarrow r_t$
    else
      $y_t \leftarrow r_t + \gamma\, max_{a_{t+1}}\, Q(s_{t+1}, a_{t+1})$
    end if
  end for
  Update θ with gradient $\bigtriangledown_\theta \mathbb{E}_{batch}[Q(s_t, a_t) - y_t]^2$
end for
return θ

Algorithm 2 Frame Dropping with Thresholding

Input: Temporal representations $\{h_i^A\}$ and $\{h_j^B\}$, videos X^A and X^B of test pair (A, B), threshold th
Output: Terminated state s_T

$t \leftarrow 0$
while True **do**
  with constraints $\#X^A \geq th$ and $\#X^B \geq th$
    Find all possible actions $T = \{a_i\}$
  $a_t \leftarrow \text{argmax}_{a_t \in T}\, Q(s_t, a_t)$
  Update s_t to s_{t+1} according to a_t
  Update X^A and X^B according to a_t
  Compute $Qmax \leftarrow \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})$
  if $Qmax < 0$ or $min(\#X^A, \#X^B) \leq th$ **then**
    $s_T \leftarrow s_{t+1}$
    break
  else
    $t \leftarrow t + 1$
  end if
end while
return s_T

states and ϵ-greedy actions that select actions following π with probability ϵ and random actions with probability $1-\epsilon$.

The details of the proposed attention-aware deep reinforcement learning (ADRL) method are summarized in **Algorithm 1**.

3.3. Verification

Given a test video pair (A, B) where x_i^A is the i-th frame in video A and x_j^B is the j-th frame in video B, we first employ C_1 to compute their temporal representations $\{h_i^A\}$ and $\{h_j^B\}$, and then use the frame evaluation network C_2 to find the attentions of each video by the aforementioned step by step frame dropping process with frame number threshold th. Assume the process is terminated at s_T, the similarity of this pair of video pair can be calculated by the cosine similarity between $p^A|s_T$ and $p^B|s_T$.

The step by step frame dropping algorithm with thresholding is summarized in **Algorithm 2**.

3.4. Implementation Details

To train the recurrent layer, we adopt the triplet loss [32] with the cosine similarity to minimize the following objective:

$$min\, \mathcal{L} = \mathbb{E}_h[\max(0, \alpha - \cos(h, h_p) + \cos(h, h_n))], \quad (11)$$

where h is the temporal representation of an anchor., h_p and h_n are the temporal representations of the positive and negative samples, respectively. Given the h from the video X^A, we simply choose a random frame from another video of the same subject in the training set to extract h_p and a random frame from videos of different subjects in the training set to extract h_n. TO train the recurrent layer, we use the standard stochastic gradient descent (SGD) and set the learning rate, momentum and α as 0.001, 0 and 0.4, respectively.

For the frame evaluation network, we use the RMSprop solver [36] and set the learning rate and RMS decay as 10^{-6} and 0.9, respectively. To improve the stability of the training stage, we clip gradient to the range between -0.5 to 0.5 and set the size of mini-batch as 64. The ϵ-greedy probability ϵ is annealed linearly from 1.0 to 0.1 in the first $1,000$ batches and fixed at 0.1 thereafter, as [30]. We set the discount factor of MDP γ and frame dropping threshold th as 0.98 and 30, respectively.

3.5. Discussion

Hard Attention vs. Soft Attention: Different from the soft attention model, hard attention model may destroy the structure of videos. In other words, the relationship among frames is ignored when using the hard attention model. Therefore, we add the temporal representation learning part to our proposed framework to help the framework take both the local and global information in video into account. Moreover, soft attention model is differentiable, thus attention model together with feature representation network can be trained end-to-end by using the standard SGD algorithm. To address the problem of undifferentiable hard attention model, we employ the deep reinforcement method and design an evaluative supervision signal by C_1 to train the attention model without extra labels. Furthermore, step-by-step frame evaluation is more effective than deciding the qualities of all frames directly. Experiments in Section 4.2 also demonstrate this point.

Attention Model vs. Manifold Model: Unlike still face recognition, video face recognition usually has more information as well as noise of the subjects. Unlike conventional manifold models that take the entire video into account, the attention model can discard valueless information of videos, which can denoise the input video effectively. Therefore, our attention model is more robust and effective in practical video face recognition applications.

4. Experiments

We conducted experiments on three widely used datasets including the YouTube Face dataset (YTF) [41], Point-and-Shoot Challenge (PaSC) [4] and Youtube celebrities dataset (YTC) [23] to evaluate our proposed ADRL method, and compared it with state-of-the-art video face recognition methods. The following describes the details of the experiments and results.

4.1. Experiments Settings

In our experiments, we used the still face recognition network model provided by authors of [40], which is a residual convolutional network [14] trained by the joint signals of center loss and softmax proposed in their work. As suggested in their work, we employed the recently proposed algorithm MTCNN [46] to detect 5 points landmarks for faces in frames and images. We used the provided landmarks if detection fails in the testing stage. Faces in frames are aligned by similarity transformation according to the landmarks and cropped to 112×96 to remove the background information.

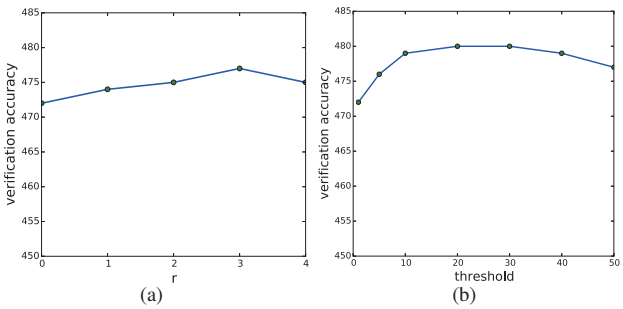
For the verification task on the YTF and PaSC datasets, we used the cosine similarity and threshold comparison as described in Algorithm 2, where thresholds are computed from the training set. For the identification task on the YTC dataset, we computed the cosine similarity between examples in training set and examples in testing set and decided the categories according to the nearest neighbor rule.

Many previous works [31, 40] used the mean-pooling of CNN features as the representation of the video, so we set the mean-pooling of the still face recognition network as the baseline in our experiments. The performance of our approach was evaluated by comparisons with state-of-the-art methods and the baseline method.

To further show the effectiveness of our method, we have conducted additional experiments by fine-tuning CNN model following [10], which we referred to as *ADRL-finetune*. Our CNN model was fine-tuned on training set of the corresponding video face dataset as [10] and supervised by the triplet loss with the learning rate 0.001. All other settings remained unchanged.

4.2. Results on YouTube Face Dataset

We first evaluated our method on the YTF dataset, which contains 3,425 videos of 1,595 different subjects. There



(a) (b)

Figure 4. The verification accuracy under varying (a) r and (b) th on the YTF dataset split 1.

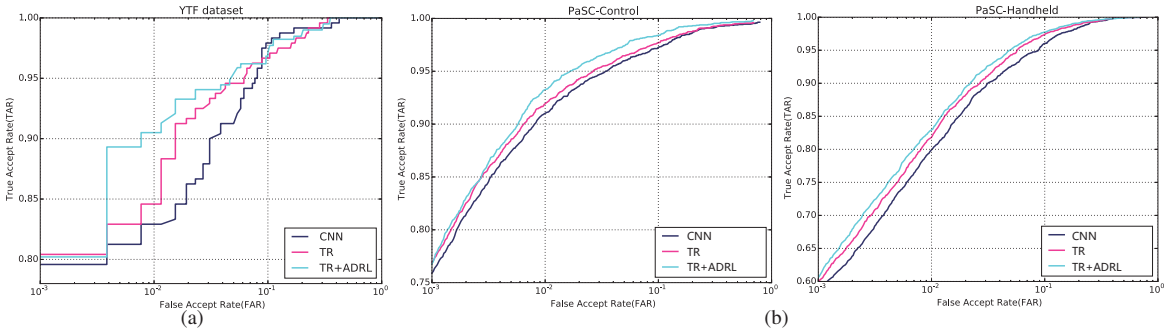
Table 1. Comparisons of the average verification accuracy with the state-of-the-art results on the YTF dataset.

Method	Accuracy	Year
LM3L [16]	81.3 ± 1.2	2014
DDML [15]	82.3 ± 1.2	2014
EigenPEP [24]	84.8 ± 1.4	2014
DeepFace-single [35]	91.4 ± 1.1	2015
DeepID2+ [34]	93.2 ± 0.2	2015
FaceNet [32]	95.12 ± 0.39	2015
Deep FR [31]	97.3	2015
NAN [43]	95.72 ± 0.64	2016
Wen *et al.* [40]	94.9	2016
TBE-CNN [10]	94.96 ± 0.31	2017
ADRL	95.96 ± 0.59	
ADRL-finetune	96.52 ± 0.54	

are many challenging videos in this dataset, including amateur photography, occlusions, problematic lighting, pose and motion blur. The length of videos in this dataset vary from 48 to 6,070 frames, and the average length of all videos is 181.3 frames. We followed the standard verification protocol and tested our approach for unconstrained face verification with 5,000 given video pairs and 10 splits, where each split has around 250 intra-personal pairs and around 250 inter-personal pairs. For the YTF dataset, we set the range of contiguous frames r as 3.

Comparison with the state-of-the-art: We compared our method with nine state-of-the-art face recognition methods, which are presented in Table 1. We see that our proposed ADRL method outperforms all other state-of-the-art methods except the deep FR method. The reason is that the deep FR method benefits a lot from front face selection and triplet loss embedding with carefully selected triplets. Compared to their work, our embedding method is more easy to implement and our faces selection model can be trained without extra labels.

Comparison with the other attention-based model: Different from still face recognition, NAN is an attention-based framework designed for video face recognition. We



(a) (b)

Figure 6. The ROC curve of our proposed method on the (a) YTF dataset split 1 and (b) PaSC.

Table 2. Comparisons of the average verification accuracy (%) with the other attention-based model on the YTF dataset. CNN is the result of mean-pooling CNN feature, TR is the result of temporal representation learning.

Method	Accuracy
NAN-CNN	95.20 ± 0.76
NAN	95.72 ± 0.64
Ours-CNN	93.64 ± 1.07
Ours-CNN (finetuned)	94.12 ± 0.76
Ours-TR	94.78 ± 0.85
Ours-TR+ADRL	95.96 ± 0.59
Ours-TR+ADRL (finetuned)	96.52 ± 0.54

compared our method with NAN in Table 2. We see that our ADRL outperforms NAN on both the verification accuracy and the standard deviation. We also compared our methods with our baseline CNN and presented the ROC curve in Figure 6. The TR and ADRL methods improve the baseline method by 1.18% and 2.32%, and reduce the error rate of the baseline method by 18.6% and 36.5%, respectively. In their work, they used a more powerful baseline model, and improved less compared to our proposed method because they didn't make use of the temporal and structural information in video pairs, and their attention model is designed only in the feature space. Besides, their model directly decides the qualities of the input video frames by using feature vectors. Their method is faster but less effective compared to our method. When we fine-tuned our model, the performance of our CNN model approaches their CNN model, and ADRL can further boost the final performance and outperform NAN more.

Analysis on temporal representation learning: Temporal representation takes the relationship between video frames into consideration. We tested our TR model varying different r from 0 to 4 on the YTF dataset split 1. When $r = 0$, TR leaning is the same as triplet loss embedding proposed in [32]. We presented the results in Figure 4(a). We see that the proposed TR learning method is better than

Figure 5. Examples from the YTF dataset. Faces in left column are the first three frames dropped from videos, faces in right column are the remaining frames that have the smallest Q, faces are sorted by Q.

previous embedding learning methods for video face recognition.

Table 3. Comparisons of the verification rate (%) with the state-of-the-art methods on the PaSC at a false accept rate(FAR) of 0.01.

Method	Control	Handheld
PittPatt	48.00	38.00
DeepO2P [22]	68.76	60.14
VGGFace	78.82	68.24
SPDNet [18]	80.12	72.83
GrNet [21]	80.52	72.76
TBE-CNN [10]	97.80	96.12
Ours-CNN	91.02	79.91
Ours-CNN (finetuned)	93.76	91.34
Ours-TR	91.92	82.43
Ours-ADRL	93.13	83.69
Ours-ADRL (finetuned)	95.67	93.78

Analysis on deep reinforcement learning: To show the robustness of our proposed method, we tested our attention model at varying values of th from 1 to 50 on the YTF dataset split 1, and the results are shown in Figure 4(b). We see that our method performs stably over a wide range of th from 10 to 40.

Figure 5 shows the examples of our MDP results. We see that our attention model learns to find the low quality frames from video and keep relatively high quality frames.

4.3. Results on Point-and-Shoot Challenge

Point-and-Shoot Challenge (PaSC) is a standard video face dataset, which contains 2,802 videos of 265 subjects balanced with varied factors such as the distance to the camera, viewpoints, the sensor types and *etc*. Two halves of the dataset are taken by control and handheld cameras respectively. Compared to the YTF dataset, PaSC is more challenging because faces in this dataset have full pose variation. For the PaSC dataset, we set the range of contiguous frames r as 1 because of the large pose variation.

Since the PaSC dataset does not have the training set, we directly used our models trained on the YTF dataset to evaluate our method on PaSC and compared them with other methods. Following the standard protocol, we reported the results at a false accept rate of 0.01, which are presented in Table 3. Figure 6 shows the ROC curves of our proposed methods. We used the results of PittPatt presented in [43] and the results of DeepO2P [22] and VGGFace presented in [18]. We can see that our proposed methods achieve very competitive performance compared state-of-the-art methods on the PaSC dataset without highly-engineered model.

4.4. Results on Youtube Celebrities

We used the the YouTube Celebrities (YTC) dataset to evaluate the performance of our methods on the video face classification task. This dataset contains 1,910 videos of 47 subjects and the number of frames varies from 8 to 400. We followed the protocol of the standard ten-fold cross validation. For each subject in each fold, we conducted experiments by selecting 3 videos for training and 6 videos for testing randomly. In our experiments, we used the model trained on the YTF dataset with $r = 3$. Table 4 shows the results of different methods in our experiments. We see that our proposed methods outperform other state-of-the-art methods on both the classification accuracy and the standard deviation, which clearly show that our methods are more effective and robust.

Table 4. Comparisons of the classification accuracy (%) with the other compared state-of-the-art methods on the YTC dataset.

Method	Accuracy
MDA [37]	67.2 ± 4.0
LMKML [28]	70.31 ± 2.52
MMDML [26]	78.5 ± 2.8
GJRNP [44]	81.3 ± 2.0
DRM-WV [12]	88.32 ± 2.14
Ours-CNN	96.88 ± 0.99
Ours-TR	97.13 ± 0.52
Ours-ADRL	97.82 ± 0.51

5. Conclusion

In this paper, we have presented a new attention-aware deep reinforcement learning (ADRL) method for video face recognition, which aims to discard the misleading and confounding frames and find the focuses of attention in video. Our method achieves very competitive performance of video face recognition on the widely used YTF, PaSC and YTC datasets.

While our method is designed for video face recognition, it can also be applied in other computer vision tasks, especially for other video-based visual recognition applications such as video action recognition, event detection and visual tracking, which is an interesting future work.

Acknowledgements

We would like to thank anonymous reviewers for their insightful comments and helpful advice. This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1001004, the National Natural Science Foundation of China under Grants 61672306, 61572271, 61527808, 61373074 and 61373090, the National 1000 Young Talents Plan Program, the National Basic Research Program of China under Grant 2014CB349304, the Ministry of Education of China under Grant 20120002110033, and the Tsinghua University Initiative Scientific Research Program.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 3

[2] R. Bellman. Dynamic programming and lagrange multipliers. *PNAS*, 42(10):767–769, 1956. 4

[3] R. Bellman. A markovian decision process. *Indiana Univ. Math. J.*, 6:679–684, 1957. 1

[4] J. R. Beveridge, P. J. Phillips, D. S. Bolme, B. A. Draper, G. H. Givens, Y. M. Lui, M. N. Teli, H. Zhang, W. T. Scruggs, K. W. Bowyer, et al. The challenge of face recognition from digital point-and-shoot cameras. In *BTAS*, pages 1–8, 2013. 1, 6

[5] J. C. Caicedo and S. Lazebnik. Active object localization with deep reinforcement learning. In *ICCV*, pages 2488–2496, 2015. 2

[6] H. Cevikalp and B. Triggs. Face recognition based on image sets. In *CVPR*, pages 2567–2573, 2010. 1, 2

[7] J.-C. Chen, V. M. Patel, and R. Chellappa. Unconstrained face verification using deep cnn features. In *WACV*, pages 1–9, 2016. 1

[8] J.-C. Chen, R. Ranjan, A. Kumar, C.-H. Chen, V. M. Patel, and R. Chellappa. An end-to-end system for unconstrained face verification with deep convolutional neural networks. In *ICCVW*, pages 118–126, 2015. 1

[9] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman. Lip reading sentences in the wild. *arXiv preprint arXiv:1611.05358*, 2016. 3

[10] C. Ding and D. Tao. Trunk-branch ensemble convolutional neural networks for video-based face recognition. *PAMI*, 2017. 6, 8

[11] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *NN*, 18(5):602–610, 2005. 3

[12] M. Hayat, M. Bennamoun, and S. An. Deep reconstruction models for image set classification. *PAMI*, 37(4):713–727, 2015. 8

[13] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, pages 1026–1034, 2015. 4

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6

[15] J. Hu, J. Lu, and Y.-P. Tan. Discriminative deep metric learning for face verification in the wild. In *CVPR*, pages 1875–1882, 2014. 1, 6

[16] J. Hu, J. Lu, J. Yuan, and Y.-P. Tan. Large margin multi-metric learning for face and kinship verification in the wild. In *ACCV*, pages 252–267, 2014. 6

[17] Y. Hu, A. S. Mian, and R. Owens. Sparse approximated nearest points for image set classification. In *CVPR*, pages 121–128, 2011. 2

[18] Z. Huang and L. Van Gool. A riemannian network for spd matrix learning. *arXiv preprint arXiv:1608.04233*, 2016. 1, 2, 8

[19] Z. Huang, R. Wang, S. Shan, and X. Chen. Learning euclidean-to-riemannian metric for point-to-set classification. In *CVPR*, pages 1677–1684, 2014. 2

[20] Z. Huang, R. Wang, S. Shan, X. Li, and X. Chen. Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification. In *ICML*, pages 720–729, 2015. 2

[21] Z. Huang, J. Wu, and L. Van Gool. Building deep networks on grassmann manifolds. *arXiv preprint arXiv:1611.05742*, 2016. 1, 2, 8

[22] C. Ionescu, O. Vantzos, and C. Sminchisescu. Matrix backpropagation for deep networks with structured layers. In *ICCV*, pages 2965–2973, 2015. 8

[23] M. Kim, S. Kumar, V. Pavlovic, and H. Rowley. Face tracking and recognition with visual constraints in real-world videos. In *CVPR*, pages 1–8, 2008. 6

[24] H. Li, G. Hua, X. Shen, Z. Lin, and J. Brandt. Eigen-pep for video face recognition. In *ACCV*, pages 17–33, 2014. 1, 6

[25] M. L. Littman. Reinforcement learning improves behaviour from evaluative feedback. *Nature*, 521(7553):445–451, 2015. 2

[26] J. Lu, G. Wang, W. Deng, P. Moulin, and J. Zhou. Multi-manifold deep metric learning for image set classification. In *CVPR*, pages 1137–1145, 2015. 2, 8

[27] J. Lu, G. Wang, and P. Moulin. Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning. In *ICCV*, pages 329–336, 2013. 2

[28] J. Lu, G. Wang, and P. Moulin. Localized multifeature metric learning for image-set-based face recognition. *TCSVT*, 26(3):529–540, 2016. 1, 2, 8

[29] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu. Recurrent models of visual attention. In *NIPS*, pages 2204–2212, 2014. 3

[30] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. 2, 4, 5

[31] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, volume 1, page 6, 2015. 1, 2, 6

[32] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015. 1, 2, 5, 6, 7

[33] J. Sivic, M. Everingham, and A. Zisserman. who are you?-learning person specific classifiers from video. In *CVPR*, pages 1145–1152, 2009. 2

[34] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. In *CVPR*, pages 2892–2900, 2015. 6

[35] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, pages 1701–1708, 2014. 2, 6

[36] T. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4(2), 2012. 5

[37] R. Wang and X. Chen. Manifold discriminant analysis. In *CVPR*, pages 429–436, 2009. 8

[38] R. Wang, H. Guo, L. S. Davis, and Q. Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In *CVPR*, pages 2496–2503, 2012. 2

[39] C. J. C. H. Watkins and P. Dayan. Q-learning. *ML*, 8(3):279–292, 1992. 4

[40] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, pages 499–515, 2016. 1, 3, 6

[41] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *CVPR*, pages 529–534, 2011. 1, 6

[42] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, pages 2048–2057, 2015. 3

[43] J. Yang, P. Ren, D. Chen, F. Wen, H. Li, and G. Hua. Neural aggregation network for video face recognition. *arXiv preprint arXiv:1603.05474*, 2016. 1, 2, 3, 6, 8

[44] M. Yang, X. Wang, W. Liu, and L. Shen. Joint regularized nearest points for image set based face recognition. *IVC*, 2016. 8

[45] F. Zhang, J. Leitner, M. Milford, B. Upcroft, and P. Corke. Towards vision-based deep reinforcement learning for robotic motion control. *arXiv preprint arXiv:1511.03791*, 2015. 2

[46] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *SPL*, 23(10):1499–1503, 2016. 6