Tracking Persons-of-Interest via Adaptive Discriminative Features

Shun Zhang[1], Yihong Gong[1], Jia-Bin Huang[2], Jongwoo Lim[3], Jinjun Wang[1], Narendra Ahuja[2] and Ming-Hsuan Yang[4]

[1]Xi'an Jiaotong University [2]University of Illinois, Urbana-Champaign
[3] Hanyang University [4]University of California, Merced
http://shunzhang.me.pn/papers/eccv2016/

Abstract. Multi-face tracking in unconstrained videos is a challenging problem as faces of one person often appear drastically different in multiple shots due to significant variations in scale, pose, expression, illumination, and make-up. Low-level features used in existing multi-target tracking methods are not effective for identifying faces with such large appearance variations. In this paper, we tackle this problem by learning discriminative, video-specific face features using convolutional neural networks (CNNs). Unlike existing CNN-based approaches that are only trained on large-scale face image datasets offline, we further adapt the pre-trained face CNN to specific videos using automatically discovered training samples from tracklets. Our network directly optimizes the embedding space so that the squared Euclidean distances correspond to a measure of semantic face similarity. This is technically realized by minimizing an improved triplet loss function. With the learned discriminative features, we apply the Hungarian algorithm to link tracklets within each shot and the hierarchical clustering algorithm to link tracklets across multiple shots to form final trajectories. We extensively evaluate the proposed algorithm on a set of TV sitcoms and music videos and demonstrate significant performance improvement over existing techniques.

1 Introduction

Multi-target tracking (MTT) aims at locating all targets of interest (e.g., faces, players, and cars), and inferring their trajectories in a video sequence over time while maintaining their identities. Multi-face tracking is one important domain of MTT that applies to numerous high-level video understanding tasks such as face recognition, content-based retrieval, surveillance, and group interaction analysis.

The goal of multi-face tracking in unconstrained scenarios is to track faces in videos that are generated from multiple moving cameras with different views or scenes as shown in Figure 1. Examples include automatic character tracking in movies, TV sitcoms, or music videos. It has attracted increased attention in recent years due to the fast growing popularity of such videos on the Internet. Unlike tracking in the *constrained* counterparts (e.g., a video from a single camera that is either fixed or moved slowly) where the main challenge is to deal with occlusions and intersections, multi-face tracking in *unconstrained* videos needs to address the following issues: (1) A video often consists of many shots. The contents of two neighboring shots may be dramatically different; (2) It entails dealing with re-identifying faces of people with large appearance

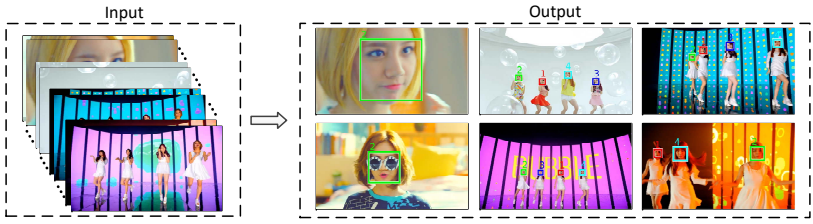


Fig. 1. We focus on tracking multiple faces according to their unknown identities in *unconstrained* videos, which consist of many shots from different cameras. The main challenge is to address large face appearance variations from different shots due to changes in pose, view angle, scale, makeup, illumination, camera motion and heavy occlusions.

variations due to changes in scale, pose, expression, illumination and make-up in different shots or scenes; and (3) The results of face detection may be unreliable when there are many uncontrolled factors, such as low resolution, occlusion, nonrigid deformation, motion blurring and complex background.

Multi-target tracking has been extensively studied in the literature with a prime focus on humans. The task is often cast as a data association problem [1,2,3,4,5] that integrates several cues such as appearance, position, motion, and size into an affinity model to link detections or tracklets (track fragments) into final trajectories. Such methods are effective when the targets are continuously detected and when the camera is either stationary or slowly moving. However, for unconstrained videos with many shot changes and intermittent appearance of targets, the data association problem becomes difficult because the assumptions such as appearance and size consistency, and continuous motion no longer hold. Therefore, the design of discriminative features plays a critical role in identifying faces *across* shots in unconstrained scenarios.

Existing MTT methods [3,4,5] generally use combinations of low-level features such as color histograms, Haar-like features, or HOG [6] to construct an appearance model for each target. However, these traditional features often are not sufficiently discriminative to identify faces with large appearance changes. For example, low-level features extracted from faces of two different persons under the same pose (e.g., frontal poses) are likely more similar than those extracted from faces of the same person under different poses (e.g., frontal and profile poses).

Recently, features extracted from a convolutional neural network (CNN) trained on a large-scale object recognition dataset have been successfully applied to a broad range of generic visual recognition tasks [7]. In particular, CNN-based features have shown impressive performance on face recognition and verification tasks [8,9,10,11]. These CNNs are often trained using large-scale face recognition datasets in a fully supervised manner and then serve as a feature extractor for unseen face images. Yet, they may not achieve good performance in unconstrained videos as the visual domains of faces in the training set and faces in given videos may be significantly different.

In this paper, we aim to address this domain shift by adapting a pre-trained CNN to the *specific* videos. Due to the lack of manual annotations of target identities, we collect a large number of training samples of faces by exploiting spatio-temporal constraints of tracklets in an unsupervised manner. With these automatically discovered training samples, we adapt the pre-trained CNN with an improved triplet loss function, so the

squared Euclidean distance of the learned features reflects the semantic distance of face images. We incorporate these adaptive features into a hierarchical agglomerative clustering algorithm to link tracklets across multiple shots into final trajectories. We demonstrate the effectiveness of the learned features to identify characters in 7 long TV sitcom episodes and singers in 8 challenging music videos.

We make the following four contributions in this work:

- Unlike existing work that uses linear metric learning on hand-crafted features, we address the large appearance variations of faces in videos by learning video-specific features using the deep contrastive and triplet-based metric learning from automatically discovered constraints.
- We propose a new triplet loss function (SymTriplet). By visualizing and analyzing the gradient directions, we show that the SymTriplet simultaneously can pull positive pairs closer and push away negative samples from the positive pairs.
- Unlike prior work that often use face tracks with false positives manually removed, we take raw input video as the input and perform detection, tracking, clustering, and feature adaptation in a fully automatic way.
- We contribute a new dataset with 8 music videos from YouTube. We provide full annotations of 3,845 face tracklets and 117,598 face detections. The new dataset presents a new set of challenges (e.g., frequent shot/scene changes, large appearance variations, and rapid camera motion) that are crucial for developing multi-face tracking algorithms in unconstrained environments.

2 Related Work

Multi-target tracking. Multi-target tracking algorithms typically integrate appearance and motion cues into an affinity model to infer and link detections (or tracklets) into final trajectories [1,4,12,13,14]. However, MTT methods do not work well in unconstrained videos where abrupt changes across different shots/scenes occur as the assumptions of continuous appearance or motion no longer hold.

To identify targets across shots, the features need to be sufficiently discriminative to discern targets in various circumstances. Most existing MTT methods either use simple color histogram features [4,15,16,17,18] or hand-crafted features [12,19,20,21,22] as the appearance representation of targets. However, all these hand-crafted features are not tailored toward faces, and thus are less effective at handling the large appearance variations of faces in unconstrained scenarios.

Visual constraints in multi-target tracking. Several methods explore visual constraints from tracklets for improving tracking performance. These constraints can then be used implicitly for learning a cast-specific metric [21,23], explicitly for linking clusters [24], or for joint tracklets linking and clustering [18]. Some work [24,25,26] also exploit the visual constraints from tracklets for face clustering in videos. Note that some other methods use non-visual cues from external sources (e.g., script) [27,28,29] and exploit the weak supervision for improving face clustering and tracking. In this paper, we exploit visual constraints generated in a manner similar to [24,25,26]. Our method differs from previous methods in two major aspects. First, existing approaches often rely on hand-crafted features and then learn a linear transformation over the extracted features,

which may not be effective to capture the nonlinear manifold where face samples lie on. In this work, we apply a deep nonlinear metric learning method by adapting all layers of the CNN to learn discriminative face feature representations. Second, previous work often use face tracks with false positives manually removed [18,23,24,25,26]. In contrast, our approach takes raw input video as the input and perform detection, tracking, clustering, and feature adaptation in a fully automatic way.

CNN-based face representation learning. With advances in deep learning, recent face recognition and verification methods focus on learning identity-preserving feature representations. While the implementation details may differ, in general these CNN-based face representations (e.g., DeepID [8], DeepFace [30], FaceNet [10], VGG-Face [31]) are learned by training CNNs using large-scale face recognition datasets in a fully supervised manner. These CNNs then act as face feature extractors for face recognition, identification, and clustering tasks. Similar to [8,10,30,31], our approach learns feature representation in a purely data-driven manner. The main difference lies in that we further adapt the pre-trained features to a specific video, resulting in improved discriminative ability. Also, we introduce a new triplet-based loss function and demonstrate its effectiveness over the commonly used contrastive loss and triplet loss.

Long-term object tracking. The goal of long-term object tracking [32,33] is to track a specific target over time and re-detect it when the target leaves and re-enters the scene. These trackers perform well on various types of targets, including human faces. However, these online trackers are designed to handle scenes recorded by a stationary or slow-moving camera, and not effective in tracking faces in unconstrained videos because of the following two fundamental limitations. First, they are prone to target drift due to the noisy training samples collected during the online model update process. Second, the features employed by these trackers are rather low-level ones which are not sufficiently discriminative to re-identify faces across different shots/scenes. We tackle the first issue by processing offline, i.e., apply a face detector in every frame and then associate all detections/tracklets in the video. For the second issue, we present to learn discriminative and adaptive features to address face appearance variations.

3 Algorithmic Overview

Our goal is to track multiple faces across many shots in an unconstrained video while maintaining identities of persons. To achieve this, we learn discriminative face appearance features that are adapted to the appearance variations presented in the specific videos. We then use a hierarchical clustering algorithm to link tracklets across shots into final trajectories. We illustrate the four main steps of our algorithm in Figure 2:

(a) **Pre-training**: We pre-train a CNN model based on the AlexNet [34] using an external face recognition dataset to learn identity-preserving features (Section 6.1).
(b) **Automatic training sample discovery**: We detect shot changes and divide the video into non-overlapping shots. Within each shot, we apply a face detector and link adjacent detections into short tracklets. We discover a large collection of face pairs or face triplets from tracklets based on spatio-temporal constraints (Section 4.1).
(c) **Adaptive feature learning**: We adapt the pre-trained CNN using the automatically discovered training samples to address large appearance changes of the imaged

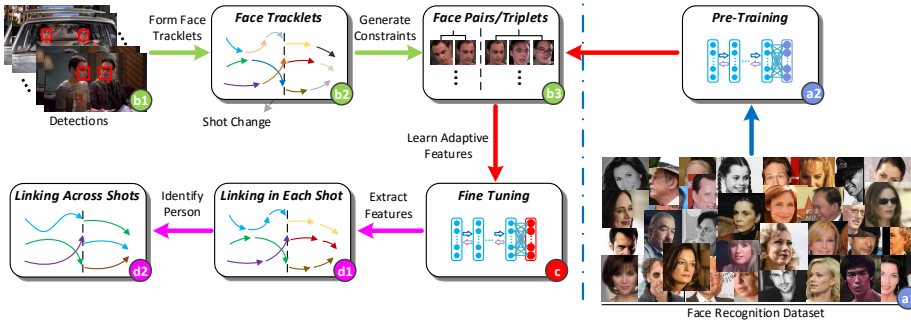


Fig. 2. Our method of tracking faces in unconstrained videos has four main steps. (a) Pre-training a CNN on a large-scale face recognition dataset. (b) Generating face pairs or face triplets from the tracklets in a specific video. (c) Adapting the pre-trained CNN to learn video-specific features. (d) Linking tracklets in each shot and then across shots to form the final face trajectories.

faces presented in a specific video (Section 4.2). For adapting the CNN, we first introduce two types of loss functions for optimizing the embedding space: the contrastive loss and the triplet loss. Moreover, we present a new triplet loss to improve the discriminative ability of learned features (Section 4.3).

(d) **Linking tracklets**: For each shot, we use conventional multi-face tracking methods to link tracklets into short trajectories. We use a hierarchically clustering algorithm to link trajectories across shots. We assign the tracklets in each cluster with the same identity (Section 5).

4 Learning Adaptive Discriminative Features

We present the algorithmic details of our unsupervised learning approach of discriminative, video-specific features, including training sample discovery, adaptive feature learning, and the improved triplet loss function.

4.1 Automatic training sample discovery

Shot detection and tracklets linking. We first use a publicly available shot change detection method to divide the input video into non-overlapping shots.[1] We then use an off-the-shelf face detector [35] to locate faces in each frame. Given face detections, we use a two-threshold strategy [16] to generate tracklets within each shot by linking the detected faces in adjacent frames based on their similarities in appearances, positions, and scales of the bounding boxes. Note that the two-threshold strategy for linking detections could be replaced by more sophisticated methods, e.g., tracking using particle filters [13,36]. We discard all tracklets shorter than five frames. Our face tracklets are conservative with limited temporal spans up to the length of each shot.

Spatio-temporal constraints. Given a set of tracklets, we can discover a large collection of positive and negative training sample pairs belonging to the same/different

[1] http://sourceforge.net/projects/shot-change/

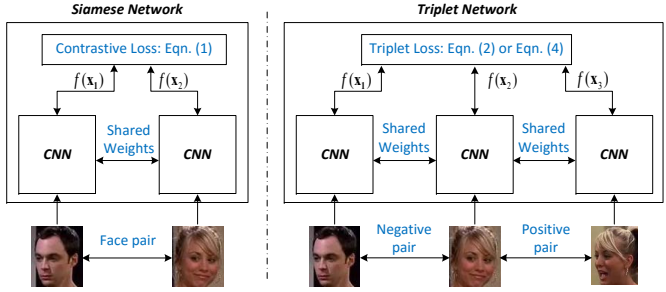


Fig. 3. Illustration of the Siamese network (left) with pairs as inputs and the triplet network (right) with triplets as inputs for Adaptive discriminative feature learning. The Siamese network consists of two CNNs and uses a contrastive loss, while the Triplet network consists of three CNNs and uses a triplet loss. The CNNs in each network share the same architectures and parameters, and are initialized with the pre-trained parameters on the large-scale face recognition dataset.

persons: (1) all pairs of faces in one tracklet are from one person and (2) two face tracklets that appear in the same frame contain faces of different persons.

Let $\mathbf{T}^i = \{\mathbf{x}_1^i, \ldots, \mathbf{x}_{n_i}^i\}$ denote the i^{th} face tracklet of length n_i. We generate a set of positive pairs $\mathbf{P}^+$ by collecting all within-tracklet face pairs: $\mathbf{P}^+ = \{(\mathbf{x}_k^i, \mathbf{x}_l^i)\}$, s.t. $\forall k, l = 1, ..., n_i,\ k \neq l$. Similarly, if tracklets $\mathbf{T}^i$ and $\mathbf{T}^j$ overlap in some frames, we can generate a set of negative pairs $\mathbf{N}^-$ by collecting all between-tracklet face pairs: $\mathbf{N}^- = \{(\mathbf{x}_k^i, \mathbf{x}_l^j)\}$, s.t. $\forall k = 1, ..., n_i,\ \forall l = 1, ..., n_j$.

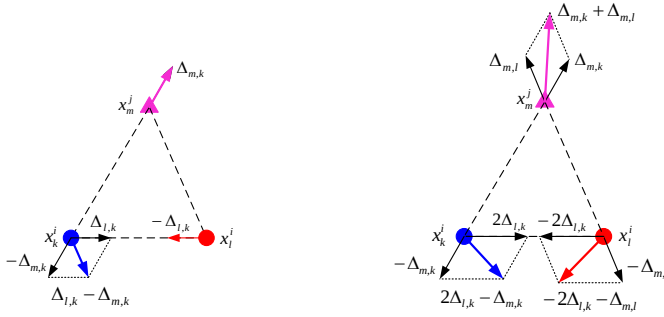
4.2 Adaptive discriminative feature learning

With the automatically discovered training pairs, we aim to optimize the embedding function $\mathbf{f}(\cdot)$ based on the CNN so that the squared Euclidean distance in the embedding space $D(\mathbf{f}(\mathbf{x}_1), \mathbf{f}(\mathbf{x}_2))$ reflects the semantic similarity of two face images $\mathbf{x}_1$ and $\mathbf{x}_2$. We set the feature dimension $\mathbf{f}(\cdot)$ as 64 in all our experiments. In what follows, we first describe two commonly used loss functions for optimizing the embedding space: (1) contrastive loss and (2) triplet loss, and then present an improved loss function for feature learning.

Contrastive loss. The Siamese network [37,38] consists of two identical CNNs with shared architecture and parameters as shown in Figure 3(left). Minimizing the contrastive loss function encourages small distance of two images of the same person and large distance otherwise. Denote $(\mathbf{x}_1, \mathbf{x}_2) \in \{\mathbf{P}^+, \mathbf{N}^-\}$ as a pair of training images generated with the spatio-temporal constraints. Following [37,38], the contrastive loss function is of the form:

$$L_p = \begin{cases} \frac{1}{2} D(\mathbf{f}(\mathbf{x}_1), \mathbf{f}(\mathbf{x}_2)) & \text{if } (\mathbf{x}_1, \mathbf{x}_2) \in \mathbf{P}^+ \\ \frac{1}{2} \max\left[0, \tau - D(\mathbf{f}(\mathbf{x}_1), \mathbf{f}(\mathbf{x}_2))\right] & \text{if } (\mathbf{x}_1, \mathbf{x}_2) \in \mathbf{N}^- \end{cases}, \tag{1}$$

where τ ($\tau = 1$ in all our experiments) is the margin. Intuitively, if $\mathbf{x}_1$ and $\mathbf{x}_2$ are from the same person, the loss is $\frac{1}{2} D(\mathbf{f}(\mathbf{x}_1), \mathbf{f}(\mathbf{x}_2))$ and we aim to decrease $D(\mathbf{f}(\mathbf{x}_1), \mathbf{f}(\mathbf{x}_2))$. Otherwise, the loss is $\frac{1}{2} \max[0, \tau - D(\mathbf{f}(\mathbf{x}_1), \mathbf{f}(\mathbf{x}_2))]$ and we increase $D(\mathbf{f}(\mathbf{x}_1), \mathbf{f}(\mathbf{x}_2))$ until it is larger than the margin τ.



(a) Original triplet loss (b) SymTriplet loss

Fig. 4. Illustration of the negative partial gradient direction to the triplet sample. (a) the original triplet loss; (b) the SymTriplet loss. The triplet samples $\mathbf{x}_k^i$, $\mathbf{x}_l^i$ and $\mathbf{x}_m^j$ are highlighted with blue, red and magenta, relatively. The circles denote faces from the same person whereas the triangle denotes a different person. The gradient directions are color-coded.

Triplet loss. The Triplet-based network [10] consists of three identical CNNs with shared architecture and parameters as shown in Figure 3(right). One triplet consists of two face images of one person and one face image from another person. We can generate a set of triplets $\mathbf{S}$ from two tracklets $\mathbf{T}^i$ and $\mathbf{T}^j$ that overlap in some frames: $\mathbf{S} = \{(\mathbf{x}_k^i, \mathbf{x}_l^i, \mathbf{x}_m^j)\}$, s.t. $\forall k, l = 1, ..., n_i,\ k \neq l,\ \forall m = 1, ..., n_j$. Here we aim to ensure that the embedded distance of the positive pair $(\mathbf{x}_k^i, \mathbf{x}_l^i)$ is closer than that of the negative pair $(\mathbf{x}_k^i, \mathbf{x}_m^j)$ by a distance margin α ($\alpha = 1$). For one triplet, the triplet loss is of the form:

$$L_t = \frac{1}{2}\max\left(0,\ D(\mathbf{f}(\mathbf{x}_k^i), \mathbf{f}(\mathbf{x}_l^i)) - D(\mathbf{f}(\mathbf{x}_k^i), \mathbf{f}(\mathbf{x}_m^j)) + \alpha\right). \tag{2}$$

4.3 Improved triplet loss

We observe that in one triplet ($\mathbf{x}_k^i$, $\mathbf{x}_l^i$ and $\mathbf{x}_m^j$) there are three distances between each pair of samples. The conventional triplet loss in (2), however, takes only two of the three distances into consideration: $D(\mathbf{f}(\mathbf{x}_k^i), \mathbf{f}(\mathbf{x}_l^i))$ and $D(\mathbf{f}(\mathbf{x}_k^i), \mathbf{f}(\mathbf{x}_m^j))$. We illustrate the problem of the original triplet loss by analyzing the gradients of the loss function. We first denote the difference vector between the triplet (k, l, m), e.g., $\Delta_{l,k} = \mathbf{f}(\mathbf{x}_l^i) - \mathbf{f}(\mathbf{x}_k^i)$. If the triplet loss in (2) is non-zero, we can compute the gradients as

$$\frac{\partial L_t}{\partial \mathbf{f}(\mathbf{x}_k^i)} = -(\Delta_{l,k} - \Delta_{m,k}),\quad \frac{\partial L_t}{\partial \mathbf{f}(\mathbf{x}_l^i)} = \Delta_{l,k},\quad \frac{\partial L_t}{\partial \mathbf{f}(\mathbf{x}_m^j)} = -\Delta_{m,k}. \tag{3}$$

Fig. 4(a) visualizes the negative gradient directions for each sample. There are two issues with the original triplet loss. First, the negative point $\mathbf{x}_m^j$ is only pushed away from $\mathbf{x}_k^i$, not both of $\mathbf{x}_k^i$ and $\mathbf{x}_l^i$. Second, the positive pair $(\mathbf{x}_k^i, \mathbf{x}_l^i)$ do not move consistently. For example, $\mathbf{x}_l^i$ only move in the direction between $\mathbf{x}_k^i$ and $\mathbf{x}_l^i$, while $\mathbf{x}_k^i$ would move in the direction with a certain angle.

To address this issue, we propose an improved triplet-based loss function (SymTriplet) by considering all three distances simultaneously. We define the SymTriplet loss as:

$$L_s = \max\left[0,\ D(\mathbf{f}(\mathbf{x}_k^i), \mathbf{f}(\mathbf{x}_l^i)) - \frac{1}{2}(D(\mathbf{f}(\mathbf{x}_k^i), \mathbf{f}(\mathbf{x}_m^j)) + D(\mathbf{f}(\mathbf{x}_l^i), \mathbf{f}(\mathbf{x}_m^j))) + \alpha\right], \tag{4}$$

where α is the distance margin. The gradients of the SymTriplet loss are

$$\frac{\partial L_s}{\partial \mathbf{f}(\mathbf{x}_k^i)} = -(2\Delta_{l,k} - \Delta_{m,k}), \quad \frac{\partial L_s}{\partial \mathbf{f}(\mathbf{x}_l^i)} = 2\Delta_{l,k} + \Delta_{m,l}, \quad \frac{\partial L_s}{\partial \mathbf{f}(\mathbf{x}_m^j)} = -(\Delta_{m,k} + \Delta_{m,l}). \quad (5)$$

We visualize the negative gradient directions in Fig. 4(b). We show that the proposed SymTriplet loss directly optimize the embedding space so that the positive pair are pulled closer to each other and the negative sample ($\mathbf{x}_m^j$) is pulled away from the two positive samples ($\mathbf{x}_k^i, \mathbf{x}_l^i$). This property allows us to improve the discriminative ability of the learned features.

5 Multi-face Tracking via Hierarchical Tracklet Linking

We follow a two-step procedure to associate face tracklets generated in Section 4.1 with the learned features: (1) linking the face tracklets within each shot into shot-level tracklets, and (2) merging shot-level tracklets across multiple shots into final trajectories.

Linking tracklets within each shot. We use conventional multi-target tracking algorithms to solve the data association problem of face tracklets within each shot. The appearance of each detection is represented as a feature descriptor extracted from the learned Siamese/Triplet network. The linking probabilities between two tracklets are measured based on temporal, kinematic and appearance information. We then use the Hungarian algorithm to find a global optimum [16,39]. The tracklets with the same label are linked into shot-level tracklets.

Linking tracklets across shots. We use a simple Hierarchical Agglomerative Clustering algorithm with a stopping threshold to link tracklets across shots with the learned appearance features. We explicitly enforce the spatio-temporal constraints by setting the distances between tracklets which have overlapped frames to infinity. We iteratively merge tracklets until the smallest distance is larger than a predefined threshold θ. We present the detailed linking processes in the supplementary material.

6 Experimental Results

In this section, we first describe the implementation details, datasets, and evaluation metrics. We then compare the proposed algorithm with the state-of-the-art methods. More experimental results and videos are available in the supplementary material.

6.1 Implementation Details

Pre-training: We adopt the AlexNet [34] architecture. We replace the original 1,000 node output layer with K nodes where each node corresponds to a specific person. Using the CASIA-WebFace dataset [40], we select $K = 9,427$ persons, 80% of the images (431,300 images) for training and the rest 20% (47,140 images) as the validation set. We resize all face images to $227 \times 227 \times 3$ pixels. We use stochastic gradient descent with an initial learning rate of 0.01 that decreases by a factor of 10 for every 20,000 iterations. We use the Caffe [41] framework to train the network.

CNN fine-tuning: We adapt the pre-trained CNN with the Siamese/Triplet network. We replace the classification layer in the pre-trained network with 64 output nodes for feature embedding. As the NVIDIA GT980Ti GPU used in our experiments has only 6GB memory, we find that it does not have sufficient memory to train the VGG-Face model. We set a fixed learning rate to 0.0001 for fine-tuning.
Linking tracklets: For features from the Siamese network, we empirically set the threshold of the HAC algorithm as $\theta = 0.4$. For features from the Triplet network, we use $\theta = 5$ for both the original triplet loss and the improved triplet loss.

6.2 Datasets

We evaluate the proposed algorithm on three types of videos containing multiple persons: (1) a constrained video in a laboratory setting: Frontal [18], (2) TV sitcoms: The Big Bang Theory (BBT) dataset [18,24], and (3) music videos from YouTube.
Frontal video. This is a short video in a constrained scene taken indoors with a fixed camera. Four persons facing the camera move around and occlude each other.
BBT dataset. We select the first 7 episodes from Season 1 of the Big Bang Theory TV Sitcom (referred as BBT01-07). Each video is about 23 minutes long with the main cast of 5-13 people and is taken mostly indoors. The main difficulty lies in identifying faces of the same person from frequent changes of camera views and scenes, where faces have large appearance variations in viewing angle, pose, scale, and illumination.
Music video dataset. We introduce a new set of 8 music videos from YouTube. These videos present a new set of challenges, e.g., frequent shot/scene changes, large appearance variations, and rapid camera motion. We believe that these challenges are crucial for developing robust multi-face tracking algorithms in unconstrained environments. Three of the sequences (T-ARA, WESTLIFE and PUSSYCAT DOLLS) are live vocal concert recordings from multiple cameras with different views. The other sequences (BRUNO MARS, APINK, HELLO BUBBLE, DARLING and GIRLS ALOUD) are MTV videos. Faces in these videos often undergo large appearance variations due to changes in pose, scale, makeup, illumination, camera motion, and occlusions.

6.3 Evaluation metrics

We evaluate the proposed method in two aspects. First, to evaluate the effectiveness of the learned video-specific features, we use the bottom-up HAC algorithm to merge pairs of tracklets until all tracklets have been merged into the ideal number of clusters (i.e., the actual number of people in the video). We measure the quality of clustering using the weighted purity: $W = \frac{1}{M}\sum_c m_c p_c$, where each cluster c contains m_c elements and its purity p_c is measured as a fraction of the largest number of faces from the same person to m_c, and M denotes the total number of faces in the video. Second, we evaluate the method with the metric set commonly used in multi-target tracking [42], including Recall, Precision, F1, FAF, IDS, Frag, MOTA and MOTP.

6.4 Evaluation on features

We evaluate the proposed adaptive features against several alternatives:

Table 1. Clustering results on 7 BBT videos and 8 music videos. The weighted purity of each video is measured on the ideal number of clusters.

BBT dataset								
Episodes	BBT01	BBT02	BBT03	BBT04	BBT05	BBT06	BBT07	
HOG	0.37	0.32	0.38	0.35	0.29	0.26	0.31	
AlexNet	0.47	0.32	0.45	0.35	0.29	0.26	0.45	
Pre-trained	0.62	0.72	0.73	0.57	0.52	0.52	0.61	
VGG-Face	0.91	0.85	0.83	0.54	0.65	0.46	0.82	
Ours-Siamese	**0.94**	**0.95**	0.87	0.74	0.70	0.70	0.89	
Ours-Triplet	**0.94**	**0.95**	**0.92**	0.74	0.68	0.70	0.89	
Ours-SymTriplet	**0.94**	**0.95**	**0.92**	**0.78**	**0.85**	**0.75**	**0.91**	
Ours-SymTriplet-BBT02	0.90	**0.95**	0.87	0.74	0.79	0.67	0.88	

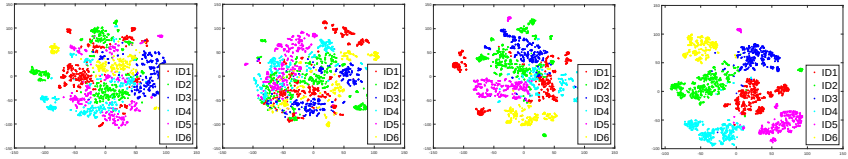
Music dataset								
Videos	T-ara	Pussycat Dolls	Bruno Mars	Hello Bubble	Darling	Apink	Westlife	Girls Aloud
HOG	0.22	0.28	0.36	0.35	0.19	0.20	0.27	0.29
AlexNet	0.25	0.31	0.36	0.31	0.18	0.22	0.37	0.30
Pre-trained	0.31	0.31	0.50	0.34	0.24	0.29	0.37	0.33
VGG-Face	0.23	0.46	0.44	0.29	0.20	0.24	0.27	0.31
Ours-Siamese	**0.69**	0.77	0.88	0.54	0.46	0.48	0.54	0.67
Ours-Triplet	0.68	0.77	0.83	0.60	0.49	0.60	0.52	0.67
Ours-SymTriplet	**0.69**	**0.78**	**0.90**	**0.64**	**0.70**	**0.72**	**0.56**	**0.69**

- Ours-SymTriplet: a 64-D feature trained with the SymTriplet loss.
- Ours-Triplet: a 64-D feature trained with the original triplet loss.
- Ours-Siamese: a 64-D feature trained with the contrastive loss.
- HOG: a conventional hand-crafted feature with 4,356 dimensions.
- AlexNet: a generic feature representation with 4,096 dimensions from the AlexNet.
- Pre-trained: a 4,096-dimensional face feature from the AlexNet architecture trained on the WebFace dataset.
- VGG-Face [31]: a publicly available face descriptor with 4,096 dimensions.

We note that our 64-D features are more compact than all other baseline features.

Clustering purity. We quantitatively evaluate the above features on the all videos. Table 1 shows the performance of different features on 7 BBT sequences and 8 music videos. We show that identity-preserving features (Pre-trained and VGG-Face) trained on face datasets offline achieve better performance over generic feature representation (e.g., AlexNet and HOG). Our video-specific features trained with Siamese and Triplet networks achieve superior performance to other alternatives, highlighting the importance of learning adaptive features. Using the proposed Symmetric Triplet loss function, Ours-SymTriplet achieves the best performance. For example, in the DARING sequence, Ours-SymTriplet achieves the highest weighted purity of 0.70, significantly outperforming other features, e.g., Ours-Siamese: 0.46, Ours-Triplet: 0.49, and VGG-Face: 0.20. Overall, our results are more than twice as accurate as VGG-Face in music videos. For the BBT dataset, Ours-SymTriplet consistently outperform all other alternatives.

The t-SNE visualization. In Figure 5, we extract the features using HOG, AlexNet, Pre-trained and Ours-SymTriplet on the T-ARA video, and visualize them in 2D using the t-SNE algorithm [43]. For HOG there exists no clear cluster structures, faces of the same person are scattered into many different places. It shows that the conventional features are not effective for faces with large appearance variations. Compared to HOG, the AlexNet and Pre-trained features increase inter-person distances, but the clusters of



HOG (4356-D) AlexNet (4096-D) Pre-trained (4096-D) Ours-SymTriplet (64-D)

Fig. 5. 2D tSNE visualization of all face features from the proposed fine-tuned CNN for adapting video-specific variations, compared with HOG, AlexNet, and Pre-trained features. T-ARA has 6 main casts. The faces of different people are color coded.

the same person do not group together. Our adaptive features form tighter clusters for the same person and greater separation for different persons.

6.5 Multi-face tracking

We compare the proposed algorithm with several state-of-the-art MTT trackers, including modified versions of TLD [32], ADMM [44], IHTLS [45], and Wu et al. [18,24]. Note that TLD [32] is a long-term single-target tracker which can re-detect targets of interest when targets leave and re-enter a scene. We implement two multi-face tracking methods with TLD. The first method is called mTLD. On each sequence, we run multiple TLD trackers for all targets, and each TLD tracker is initialized with the ground truth bounding box in the first frame. For the second method, we integrate mTLD into our framework (referred as Ours-mTLD). mTLD is used to generate shot-level trajectories within each shot instead of the two-threshold and Hungarian algorithms. At the beginning of each shot, we initialize TLD trackers with untracked detections, and link the detections in the following frames according to their overlap scores with TLD outputs.

Table 2 shows quantitative results of the proposed algorithm and the mTLD [32], ADMM [44] and IHTLS [45] on the BBT and music video datasets. We also show the tracking results with the Pre-trained features instead of the proposed video-specific features. Note that the results shown in Table 2 are the *overall* evaluation. We leave the results on each individual sequence to the supplementary material.

The mTLD method does not perform well in terms of recall, precision, F1, and MOTA metrics on both datasets. We attribute the poor performance to its tendency to drift and the use of low-level features. The ADMM [44] and IHTLS [45] often produce many identity switches and fragments because they fail to re-identify persons when abrupt camera motions or shot changes occur. The Pre-trained features are not effective to identify faces in different shots, and achieve very low MOTA. Ours-mTLD has more IDS and Frag than Ours-SymTriplet. The main reason is that the shot-level trajectories by mTLD are shorter and noisier than the original trajectories, since TLD trackers sometimes drift or do not output tracking results when there are large appearance changes. Ours-SymTriplet performs better in terms of precision, F1, and MOTA metrics, with significantly fewer identity switches and fragments than the competing algorithms.

Figure 6 shows sample tracking results of our algorithm with Ours-SymTriplet features on all eight music videos and three BBT sequences (BBT01, BBT02, and BBT05). The numbers and the colors indicate the inferred identities of the targets. The

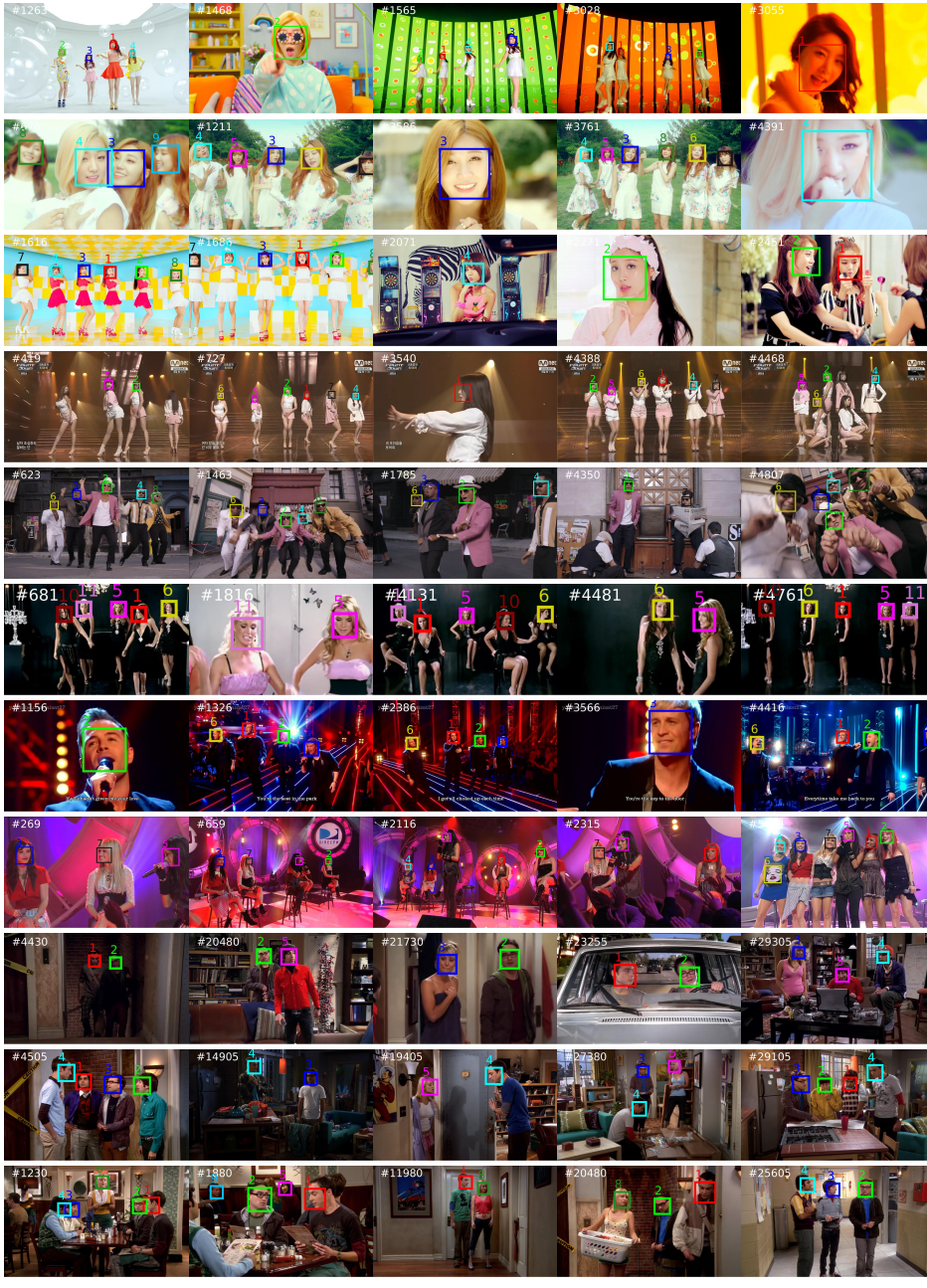


Fig. 6. Sample tracking results of the proposed algorithm. Shown from the top to bottom are HELLO BUBBLE, APINK, DARLING, T-ARA, BRUNO MARS, GIRLS ALOUD, WESTLIFE, PUSSYCAT DOLLS, BBT01, BBT02 and BBT05. The faces of the different people are color coded.

Table 2. Quantitative comparison with other state-of-the-art multi-target tracking methods on the BBT and music video datasets. The **best** and second-best values are highlighted with the bold and underline, respectively.

BBT dataset								
Method	**Recall↑**	**Precision↑**	**F1↑**	**FAF↓**	**IDS↓**	**Frag↓**	**MOTA↑**	**MOTP↑**
mTLD [32]	1.1%	8.1%	1.9%	**0.18**	**8**	**83**	-11.2%	73.2%
ADMM [44]	**78.3%**	56.8%	65.8%	0.49	2709	4623	39.5%	72.7%
IHTLS [45]	77.7%	63.4%	69.8%	0.49	2648	4496	39.2%	72.7%
Pre-trained	45.0%	76.8%	56.8%	0.19	908	2435	30.0%	**77.9%**
Ours-mTLD	63.7%	78.8%	70.5%	0.24	1224	3487	44.6%	77.6%
Ours-Siamese	74.5%	**81.4%**	77.8%	0.24	884	4051	56.1%	77.4%
Ours-Triplet	76.2%	80.2%	78.1%	0.27	944	4223	55.8%	77.3%
Ours-SymTriplet	76.6%	81.0%	**78.7%**	0.26	846	4261	**57.2%**	77.2%
Music video dataset								
Method	**Recall↑**	**Precision↑**	**F1↑**	**FAF↓**	**IDS↓**	**Frag↓**	**MOTA↑**	**MOTP↑**
mTLD [32]	9.7%	36.1%	15.3%	0.39	**280**	**621**	-7.7%	68.4%
ADMM [44]	**75.5%**	61.8%	68.0%	0.50	2382	2959	51.7%	63.7%
IHTLS [45]	**75.5%**	68.0%	71.6%	0.41	2013	2880	56.2%	63.7%
Pre-trained	60.1%	88.8%	71.7%	**0.17**	931	2140	51.5%	79.5%
Ours-mTLD	69.1%	88.1%	77.4%	0.21	1914	2786	57.7%	**80.1%**
Ours-Siamese	71.5%	89.4%	79.5%	0.19	986	2512	62.3%	64.0%
Ours-Triplet	71.8%	88.8%	79.4%	0.20	902	2546	61.8%	64.2%
Ours-SymTriplet	71.8%	**89.7%**	**79.8%**	0.19	699	2563	**62.8%**	64.3%

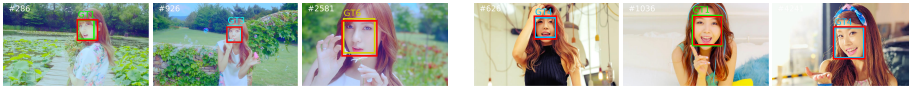


Fig. 7. Failure cases. Our method incorrectly identifies different persons as the same one across shots on the APINK and DARLING sequences. Numbers and colors of rectangles indicate the *ground truth* identities of persons. The red rectangles show the predicted locations, and are tracked as one person by our method. On the APINK sequence on the left, the three different persons are incorrectly assigned with the same identity. On the DARLING sequence on the right, the middle person is assigned with an incorrect identity.

proposed algorithm is able to track multiple faces well despite large appearance variations in unconstrained videos. For example, there are significant changes in scale and appearance (due to makeup and hairstyle) in the HELLO BUBBLE sequence (first row). In the fourth row, the six singers have similar looks and thus make multi-face tracking particularly challenging within and across shots. Nonetheless, the proposed algorithm is able to distinguish the faces and track them reliably with few id switches. The results in other rows illustrate that our method is able to generate correct identities and trajectories when the same person appears in different shots or different scenes. More results and large images are available in the supplementary material.

6.6 Discussions

While the proposed adaptive feature learning method performs favorably against the state-of-the-art face tracking and clustering methods in handling challenging video sequences, there are three main limitations.

First, as our algorithm takes face detections as inputs, the tracking performance depends on whether faces can be reliably detected. For examples, in the fourth row of

Figure 6, the left-most person was not detected in Frame 419 and adjacent frames due to occlusion, and thus is not tracked. In addition, falsely detected faces could be incorrectly linked as a trajectory, e.g., the Marilyn Monroe image on the T-shirt in Frame 5,704 in the eighth row of Figure 6.

Second, our method may not perform well on sequences where there are no sufficient training samples. We show in Figure 7 two failure cases in the DARLING and APINK sequences. In these two sequences, many shots contain only one single person. Our method thus cannot generate negative face pairs for training the Siamese/Triplet network for distinguishing similar faces. Our method incorrectly identifies different persons as the same one. A promising direction would exploit other weak supervision signals (e.g., scripts, voice, contextual information) to generate visual constraints.

Third, the CNN fine-tuning is time-consuming. It takes around 1 hour on a NVIDIA GT980Ti GPU to run 10,000 back-propagation iterations. There are two approaches that may alleviate this issue. First, we may use faster training algorithms [46,47]. Second, for TV Sitcom episodes we can use one or a few videos for feature adaptation and apply the learned features to all other episodes. Note that we only need to adapt features *once* as the main characters are the same. In Table 1, we trained Ours-SymTriplet features on BBT02 (referred to Ours-SymTriplet-BBT02), and tested on other episodes. Although the weight purity of Ours-SymTriplet-BBT02 is slightly inferior than that of Ours-SymTriplet, it still outperforms the pre-trained features and the VGG-Face.

Several other work exploit body descriptors for improving character tracking in TV Sitcom videos [48]. However, these body descriptors may not be helpful for music videos as a person may be in completely different outfits in different scenes, e.g., the first row in Figure 6. While in this work we investigate the multi-face tracking problem, we believe that our framework based on tracklet information is also applicable to other general unconstrained tracking problem, e.g., crowd analysis for pedestrians or cars in surveillance videos. We leave this for future work.

7 Conclusions

We propose an effective feature learning method for multi-face tracking. We first pre-train a CNN on a large-scale face recognition dataset to learn identity-preserving features. This CNN is adapted with samples automatically discovered from the visual constraints. In addition, we propose the SymTriplet loss function to learn more discriminative features for handling appearance variations of faces presented in a specific video. A hierarchical clustering algorithm is used to link face tracklets across multiple shots. Experimental results show that the proposed algorithm outperforms the state-of-the-art methods in terms of clustering accuracy and tracking performance.

Acknowledgement The work is partially supported by National Basic Research Program of China (973 Program, 2015CB351705), NSFC (61332018), Office of Naval Research (N0014-16-1-2314), R&D programs by NRF (2014R1A1A2058501) and MSIP/IITP (IITP-2016-H8601-16-1005) of Korea, NSF CAREER (1149783) and gifts from Adobe and NVIDIA.

References

1. Brendel, W., Amer, M., Todorovic, S.: Multiobject tracking as maximum weight independent set. In: CVPR. (2011) 2, 3
2. Collins, R.T.: Multitarget data association with higher-order motion models. In: CVPR. (2012) 2
3. Yang, B., Nevatia, R.: Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In: CVPR. (2012) 2
4. Zhang, L., Li, Y., Nevatia, R.: Global data association for multi-object tracking using network flows. In: CVPR. (2008) 2, 3
5. Zhao, X., Gong, D., Medioni, G.: Tracking using motion patterns for very crowded scenes. In: ECCV. (2012) 2
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. (2005) 2
7. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. In: ICML. (2014) 2
8. Sun, Y., Wang, X., Tang, X.: Deep learning face representation from predicting 10,000 classes. In: CVPR. (2014) 2, 4
9. Sun, Y., Chen, Y., Wang, X., Tang, X.: Deep learning face representation by joint identification-verification. In: NIPS. (2014) 2
10. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: A unified embedding for face recognition and clustering. CVPR (2015) 2, 4, 7
11. Hu, J., Lu, J., Tan, Y.P.: Discriminative deep metric learning for face verification in the wild. In: CVPR. (2014) 2
12. Andriyenko, A., Schindler, K., Roth, S.: Discrete-continuous optimization for multi-target tracking. In: CVPR. (2012) 3
13. Huang, C., Li, Y., Ai, H., et al.: Robust head tracking with particles based on multiple cues. In: ECCVW. (2006) 3, 5
14. Li, Y., Ai, H., Yamashita, T., Lao, S., Kawade, M.: Tracking in low frame rate video: A cascade particle filter with discriminative observers of different lifespans. In: CVPR. (2007) 3
15. Ben Shitrit, H., Berclaz, J., Fleuret, F., Fua, P.: Tracking multiple people under global appearance constraints. In: ICCV. (2011) 3
16. Huang, C., Wu, B., Nevatia, R.: Robust object tracking by hierarchical association of detection responses. In: ECCV. (2008) 3, 5, 8
17. Li, Y., Huang, C., Nevatia, R.: Learning to associate: Hybridboosted multi-target tracker for crowded scene. In: CVPR. (2009) 3
18. Wu, B., Lyu, S., Hu, B.G., Ji, Q.: Simultaneous clustering and tracklet linking for multi-face tracking in videos. In: ICCV. (2013) 3, 4, 9, 11
19. Andriyenko, A., Schindler, K.: Multi-target tracking by continuous energy minimization. In: CVPR. (2011) 3
20. Roth, M., Bauml, M., Nevatia, R., Stiefelhagen, R.: Robust multi-pose face tracking by multi-stage tracklet association. In: ICPR. (2012) 3
21. Wang, B., Wang, G., Chan, K.L., Wang, L.: Tracklet association with online target-specific metric learning. In: CVPR. (2014) 3
22. Kuo, C.H., Nevatia, R.: How does person identity recognition help multi-person tracking? In: CVPR. (2011) 3
23. Cinbis, R.G., Verbeek, J., Schmid, C.: Unsupervised metric learning for face identification in tv video. In: ICCV. (2011) 3, 4

24. Wu, B., Zhang, Y., Hu, B.G., Ji, Q.: Constrained clustering and its application to face clustering in videos. In: CVPR. (2013) 3, 4, 9, 11
25. Tapaswi, M., Parkhi, O.M., Rahtu, E., Sommerlade, E., Stiefelhagen, R., Zisserman, A.: Total cluster: A person agnostic clustering method for broadcast videos. In: ICVGIP. (2014) 3, 4
26. Xiao, S., Tan, M., Xu, D.: Weighted block-sparse low rank representation for face clustering in videos. In: ECCV. (2014) 3, 4
27. Bauml, M., Tapaswi, M., Stiefelhagen, R.: Semi-supervised learning with constraints for person identification in multimedia data. In: CVPR. (2013) 3
28. Sivic, J., Everingham, M., Zisserman, A.: "Who are you?" – Learning person specific classifiers from video. In: CVPR. (2009) 3
29. Everingham, M., Sivic, J., Zisserman, A.: "Hello! My name is... Buffy" – Automatic naming of characters in tv video. In: BMVC. (2006) 3
30. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: DeepFace: Closing the gap to human-level performance in face verification. In: CVPR. (2014) 4
31. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: BMVC. (2015) 4, 10
32. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. TPAMI **34**(7) (2012) 1409–1422 4, 11, 13
33. Pernici, F.: Facehugger: The alien tracker applied to faces. In: ECCV. (2012) 4
34. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: NIPS. (2012) 4, 8
35. Mathias, M., Benenson, R., Pedersoli, M., Van Gool, L.: Face detection without bells and whistles. In: ECCV. (2014) 5
36. Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E., Van Gool, L.: Robust tracking-by-detection using a detector confidence particle filter. In: ICCV. (2009) 5
37. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: CVPR. (2005) 6
38. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: CVPR. (2006) 6
39. Xing, J., Ai, H., Lao, S.: Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In: CVPR. (2009) 8
40. Yi, D., Lei, Z., Liao, S., Li, S.Z.: Learning face representation from scratch. arXiv (2014) 8
41. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: ACM MM. (2014) 8
42. Zhang, S., Wang, J., Wang, Z., Gong, Y., Liu, Y.: Multi-target tracking by learning local-to-global trajectory models. PR **48**(2) (2015) 580–590 9
43. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. JMLR **9**(2579-2605) (2008) 85 10
44. Ayazoglu, M., Sznaier, M., Camps, O.I.: Fast algorithms for structured robust principal component analysis. In: CVPR. (2012) 11, 13
45. Dicle, C., Camps, O.I., Sznaier, M.: The way they move: Tracking multiple targets with similar appearance. In: ICCV. (2013) 11, 13
46. Lin, Z., Courbariaux, M., Memisevic, R., Bengio, Y.: Neural networks with few multiplications. arXiv (2015) 14
47. Jaderberg, M., Vedaldi, A., Zisserman, A.: Speeding up convolutional neural networks with low rank expansions. BMVC (2014) 14
48. Tapaswi, M., Bauml, M., Stiefelhagen, R.: Knock! Knock! Who is it? probabilistic person identification in tv-series. In: CVPR. (2012) 14