Re-identification for Online Person Tracking by Modeling Space-Time Continuum

Neeti Narayan, Nishant Sankaran, Srirangaraj Setlur and Venu Govindaraju
University at Buffalo, SUNY
{neetinar, n6, setlur, govind}@buffalo.edu

Abstract

We present a novel approach to multi-person multi-camera tracking based on learning the space-time continuum of a camera network. Some challenges involved in tracking multiple people in real scenarios include a) ensuring reliable continuous association of all persons, and b) accounting for presence of blind-spots or entry/exit points. Most of the existing methods design sophisticated models that require heavy tuning of parameters and it is a non-trivial task for deep learning approaches as they cannot be applied directly to address the above challenges. Here, we deal with the above points in a coherent way by proposing a discriminative spatio-temporal learning approach for tracking based on person re-identification using LSTM networks. This approach is more robust when no a-priori information about the aspect of an individual or the number of individuals is known. The idea is to identify detections as belonging to the same individual by continuous association and recovering from past errors in associating different individuals to a particular trajectory. We exploit LSTM's ability to infuse temporal information to predict the likelihood that new detections belong to the same tracked entity by jointly incorporating visual appearance features and location information. The proposed approach gives a 50% *improvement in the error rate compared to the previous state-of-the-art method on the CamNeT dataset and* 18% *improvement as compared to the baseline approach on DukeMTMC dataset.*

1. Introduction

Tracking and monitoring of human activity and behavior characterization in a scene are increasingly useful but challenging tasks given the vast numbers of deployed surveillance cameras. Automated systems for analyzing and understanding massive streams of video data have become a necessity. Reliable automatic re-identification and tracking of people in dense crowds will enable continuous monitoring and analysis of events without the need for human supervision.

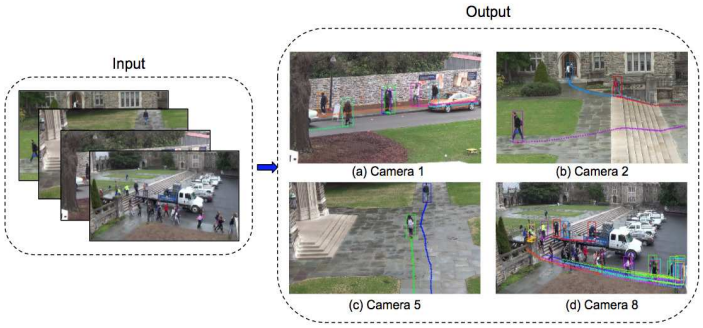


Figure 1: Camera sequences from the DukeMTMC dataset with person trajectories highlighted.

However, tracking multiple people across multiple cameras is not a trivial task, especially in complex and crowded scenarios with different scene illuminations, camera properties, frequent occlusions and interaction of individuals. Figure 1 illustrates this problem. Given a sequence of frames as input, with possibly several simultaneous observations across cameras at a given time instance, the output of a tracking system is the trajectory of each individual. Much of the work on multi-target multi-camera tracking involve two tasks: employing a motion based tracking within cameras and a separate process of re-identification or data association for targets exiting a camera boundary and re-entering the same/ different camera view [33, 10]. The main idea is to link two short tracklets into longer tracks by optimizing probabilities between tracklets globally. Existing tracking systems can follow and predict the location of known targets for long times [31, 16, 8]. However, when targets are not known, extracting the track of each individual is a difficult problem. The reason is that when two or more individuals overlap or cross or re-enter a camera view, it can be difficult to assign correct identities. An identity switch will propagate to the rest of the video and result in assigning random labels to all individuals after some time.

Owing to the recent rise of deep learning with the availability of large annotated data, we adapt the Convolutional Neural Networks (CNN) learning methodology to multi-camera tracking [28]. In addition, taking advantage of the

spatial and temporal information in videos, we study the robustness of high-level appearance features produced by convolutional networks and prediction efficacy of LSTM networks in the temporal domain for understanding and associating detections. Our proposed trajectory tracking method can deal with local temporal difficulties generated by multi-person interaction and occlusion. It is capable of recovering from past errors or misassociations and handling entry/exit scenarios more accurately. Moreover, the approach unifies the two disjoint acts of tracking within and across cameras and can thus handle time-based sparseness in video data.

Our main contributions include:

1. Learning the space-time continuum for the camera network to effectively capture the variation in the features as a function of time and space. Our approach extends the neural network learning methodology into the spatial and temporal domain for efficient multi-person multi-camera tracking.
2. Our tracking algorithm is completely automatic, giving reliably correct identities even for multi-camera scenarios with complex indoor and outdoor movements, and varying number of persons. It is also capable of handling temporal gaps in the input video.

2. Related Work

Tracking-by-detection approaches [14, 3, 17], owing to the progress in object detection [30, 20], have emerged as a useful and popular tracking strategy. All of them handle the data association problem, some taking advantage of social factors to improve tracking [2, 21]. The system is usually composed of several modules, with the feature extractor being the most important module of a tracker [29]. Thus, the performance of a tracking system significantly depends on the feature used.

In a discrete Hidden Markov Model (HMM) model, the trajectory tracking problem can be solved using the Viterbi algorithm [22], which is a dynamic programming algorithm that keeps all best sequences ending at all possible states in current frame. A well-known early work in trajectory tracking is the multiple hypothesis tracking (MHT) algorithm developed by Reid [23]. In MHT, the multi-target tracking problem is decomposed into the state estimation and data association components. Some methods are presented to model data association as random variables which are estimated jointly with state estimation by EM iterations [6, 27]. Most of these methods are in the small target tracking community where object representation is simple.

A. Alahi et al. in [1], predict trajectories of people based on their past positions using Long-Short Term Memory networks (LSTM). For every person trajectory, a separate LSTM network is used. Based on the observed positions and information shared between LSTMs through a novel Social pooling (S-pooling) layer, their model tries to predict future paths. In [26], an online method for tracking is proposed by using multiple cues such as appearance, motion, and interaction. LSTM networks are used to learn motion and interactions of humans. However, this solution is for single camera person tracking and there is no evidence of how well the system scales for multi-camera environment. A cross domain knowledge transfer scheme [32] is explored for deep learning based person re-identification by transferring knowledge of mid-level attribute features and high-level classification features. Also, the LSTM based model is extended by a special gate for use in the re-identification method. However, this work is similar to other re-identification approaches that use a pair of images over a set of non-overlapping camera views and do not model real-world multi-camera tracking scenarios.

Recently, Narayan et al. in [19] presented an association based approach for person tracking. Even though their framework overcomes the weakness of prior research in terms of tracking by re-identification, it does not propagate associations nor learn from past associations. A more coherent approach is required in order to have an effective multi-person multi-camera tracking system. We draw motivation from this work and extend it significantly to address its shortcomings.

3. Our Approach

This section presents a detailed description of the tracking approach. It is organized as follows: first, we describe the general overview of the approach. The following subsections present in detail each of the steps of the tracking algorithm - feature computation, training LSTM network and tracking using the trained model.

3.1. Overview

At each time-instant t, every active person in the scene is represented by the person bounding-box coordinates i.e. (x_t,y_t,w_t,h_t). We then extract appearance-based features for a person using a deep CNN, described in the next section. The pairwise matching probability/score s_{ij} of every previously tracked person i with every detected person j across consecutive time instances is computed based on the appearance features. These match scores s_{ij} are encoded along with location information pertaining to the corresponding observations being matched and forms a feature input z_{ij}^t at time-instant t for the LSTM. We observe and accumulate these features for upto a fixed time period for every person tracked and the entire sequence/track of features Z_i are provided to the LSTM network for predicting associations. Since at any time instance, there can be multiple detected persons to associate to a previously tracked identity, we accumulate the features arising out of each possible association to the tracked identity's feature vector and utilize the LSTM to predict its match probability. The association

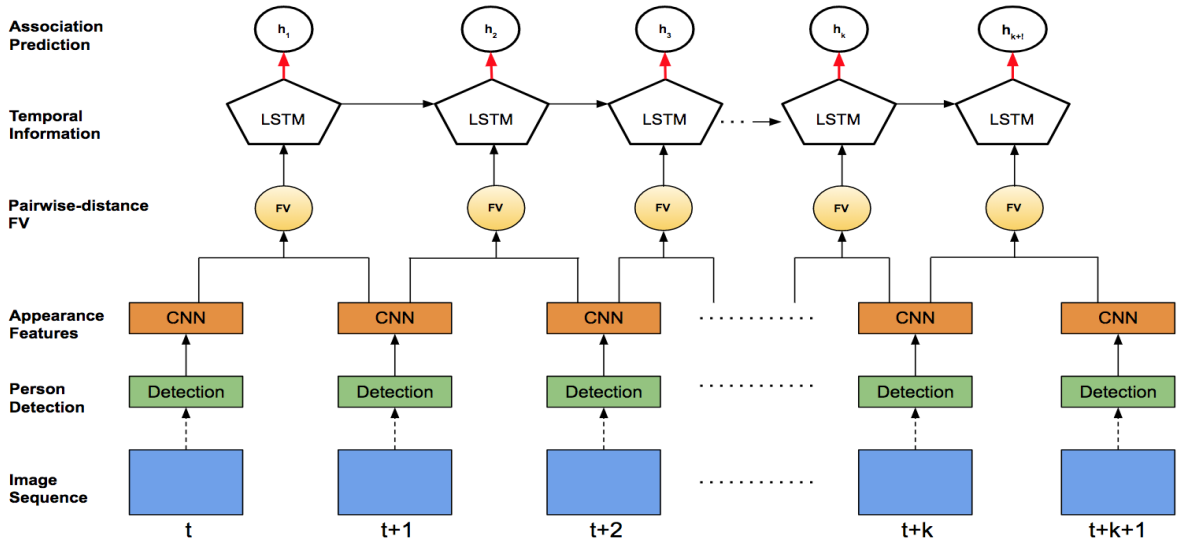


Figure 2: Proposed system overview

with the highest match probability is chosen as belonging to the tracked identity and we use this new feature vector as that identity's track for the next time instance's associations. Since no constraint is placed on the feature vector sequences being of consecutive timesteps, this formulation can effectively handle temporal gaps in multi-camera environments. When the feature vector for an identity is not long enough for the LSTM, our method defaults to using attribute-based inference for predicting the association.

Our model is spatially and temporally deep in that it exploits robust appearance features and location information of past frames. The intuition behind the method is to model the evolution of an identity's match scores (being a function of the features) across time (over a fixed time period) and space (across multi-camera and within camera location transitions), thereby uncovering the space-time continuum manifested in the camera network. We conjecture that by being able to capture how the match performance varies temporally and spatially, we can make more informed associations. The overview of our tracking system is unfolded and shown in Figure 2.

3.2. Appearance Feature Cues

We use AlexNet model [13] and DenseNet model [11] for extracting features from person detections. These appearance-based features represent traits and characteristics of an individual.

3.3. Long Short Term Memory Networks

We consider LSTM networks [9] as they are capable of learning temporal dependencies. The key to LSTM units is the embedded representation of the cell state c that acts as a memory. The presence of structures called gates controls how much of the previous state information should be kept or replaced by the new input. An LSTM has three gates, each consists of a sigmoid layer and an element-wise multiplication operation. More formally, the input i_t, output o_t and forget f_t gates are all functions of the hidden representation h_{t-1} and current input x_t. Using a sigmoid layer, a separate weight matrix W_g and bias b_g for each gate, the memory update that decides the flow of information is modeled as:

$$i_t, o_t, f_t = \sigma[W_g(h_{t-1}, x_t) + b_g] \quad (1)$$

The old cell state, c_{t-1}, is updated to the new cell state c_t as $c_t = f_t * c_{t-1} + i_t * \tilde{c_t}$ and the hidden representations are computed as $h_t = o_t * tanh(c_t)$, where $*$ represents element-wise multiplication and $\tilde{c_t}$ is a vector of new candidate values created by a tanh layer.

3.4. LSTM Input Generation

At every time instant, we model the current observation j with previously seen observation i together for the purpose of continuous entity association. A single association comprises one time-step input for the LSTM which is formed with a pairwise-distance feature vector (FV) by combining information of subject id_i with subject id_j as shown below:

$$< s_{ij}, cam_i, cam_j, x_i, y_i, w_i, h_i, x_j, y_j, w_j, h_j > \quad (2)$$

where id_i and id_j are the i^{th} and j^{th} person's ID respectively, s_{ij} is the cosine distance of appearance features computed between the two observations, cam_i and cam_j are the IDs of the cameras where subject id_i and subject id_j are

Feature Seq	Label
GGG GGG GGG G	1
GGG GGG GGG I	0
III III III G	1
III III III I	0

Table 1: Case 1

Feature Seq	Label
GGG GGG III G	1
GGG GGG III I	0
III III GGG G	1
III III GGG I	0

Table 2: Case 2

Feature Seq	Label
GGG III GII G	1
GGG III GIG I	0
III GIG IIII G	1
III GIG III I	0

Table 3: Case 3

Table 4: Association samples

seen respectively, $< x_i, y_i, w_i, h_i >$ is the person bounding box of subject id_i and $< x_j, y_j, w_j, h_j >$ is the person bounding box of subject id_j.

We generate 10-frame long sequences by extracting the pairwise-distance feature vector for 10 time instances, each feature vector of size 11. The entire 10-frame long sequence of feature vectors is fed to the LSTM network to obtain the network's final association prediction.

3.5. Learning the Association

The association of two entities is based on the pairwise-distance feature vector generated. The feature vector is attributed to being "genuine" (*G*) if $id_i = id_j$ and an "imposter" (*I*) otherwise. Our objective is to predict the association for the current sample, having observed the last 9 associations for a particular identity. This method of looking back in time by keeping few past observations to find the most likely association for a future observation helps recover from misassociations.

Below, we discuss the different cases that are likely to occur in real situations. Table 4 shows label prediction for the 10^{th} instance given 10-frame long feature sequences.

Case 1: Here, we assume that the last 9 associations are all *G* or all *I*. If a *G* feature vector is observed at the 10^{th} instance, our model should predict label 1 with high confidence; else predict label 0 meaning an association should not be made, as shown in Table 1.

Case 2: In this scenario, we introduce noise in 7^{th}, 8^{th} and 9^{th} instances. This is to model more realistic scenarios such as recovery from past incorrect associations. Table 2 depicts Case 2 scenario.

Case 3: Here, we introduce noise at any time instance. This is to model situations when our system recovers from misassociations but makes an association error again. Few sample situations are shown in Table 3.

Sequences of feature vectors are generated from a dataset according to the above mentioned scenarios and used to train the LSTM model so that it learns to capture the inherent geographic constraints in the multi-camera environment and how the genuine association scores should evolve with these constraints.

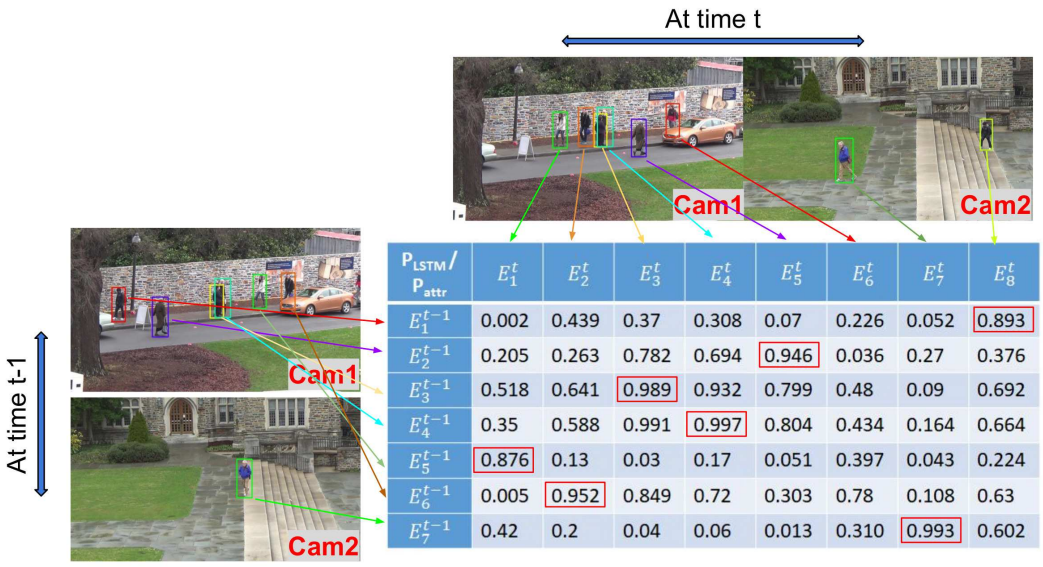
3.6. LSTM-based Space-Time Tracker

We describe the general outline of how a space-time tracker based on the LSTM network trained as described in the previous section can be utilized. All persons detected are represented by their corresponding appearance features and their location information. When a previously unseen person appears, the system would store the appearance features to match within the next timestep. Once the system has registered a match, it saves the history of matches/associations along with the location information corresponding to the match. The system accumulates the history of associations for 9 timesteps. In the 10^{th} timestep for the tracked individual, for n currently detected persons, we make n copies of the individual's association history and populate each with the association to one of the n detected persons. If m represents every individual tracked by the system for longer than 9 time steps, we obtain an $m \times n$ association matrix with each cell containing the accumulated history of the individual.

Using the LSTM network we obtain a rectified prediction for the probability of a cell's association given it's history. With the LSTM predictions, we can apply a greedy technique of selecting associations based on decreasing probability of association. Figure 3 depicts how the proposed tracker works on the association probability matrix P, where each row represents a previously tracked person and the columns hold currently detected persons.

Once the associations are decided, the system stores the corresponding history of associations for each individual and uses it for subsequent predictions and any detections not associated to a previous individual are initialized for tracking. Some key improvements of the inference-based association algorithm include the following:

Entropy-based association: With the LSTM predictions, we can apply a greedy technique of selecting associations based on decreasing probability of association. However, this method would suffer during situations where the LSTM predictions for a particular individual are extremely similar. To address this issue, we perform entropy based greedy associations where each of the m individuals are associated to one of the n detected persons according to increasing entropy of the individuals association probabilities computed as $H_i = -\sum_j^n p_{ij} \log p_{ij}$. Clearly, this policy would pri-



P_{LSTM} / P_{attr}	E_1^t	E_2^t	E_3^t	E_4^t	E_5^t	E_6^t	E_7^t	E_8^t
E_1^{t-1}	0.002	0.439	0.37	0.308	0.07	0.226	0.052	0.893
E_2^{t-1}	0.205	0.263	0.782	0.694	0.946	0.036	0.27	0.376
E_3^{t-1}	0.518	0.641	0.989	0.932	0.799	0.48	0.09	0.692
E_4^{t-1}	0.35	0.588	0.991	0.997	0.804	0.434	0.164	0.664
E_5^{t-1}	0.876	0.13	0.03	0.17	0.051	0.397	0.043	0.224
E_6^{t-1}	0.005	0.952	0.849	0.72	0.303	0.78	0.108	0.63
E_7^{t-1}	0.42	0.2	0.04	0.06	0.013	0.310	0.993	0.602

Figure 3: Illustration of LSTM-based tracking. The example above shows an association matrix in a 2 camera network. E_i^t represents the i^{th} entity at timestep t, P_{LSTM}/P_{Attr} is the association probability based on LSTM prediction or cosine similarity based on pairwise appearance attributes (best viewed in color).

oritize associations about which the LSTM network is most confident.

Re-association of tracklets: Tracks are created and updated based on the association at every timestamp. When an association error occurs, a new track is created even though the ground-truth is otherwise. To avoid propagating association errors, we re-associate tracklets once every 10 timestamps. The overall number of tracks is hence reduced and we have a more coherent tracking system now.

Subsampling: We subsample real-time footage to achieve real-time performance of the system. We take 1 in every 60 frames of footage. The LSTM network is trained on the subsampled data and the tracking model is evaluated on the subsampled set showcasing the ability of the tracking algorithm to effectively handle temporal discontinuities in multi-camera environments. The scalability of the model is also demonstrated by showing that it can learn to track people with fewer data from a dataset.

4. Experiments and Results

We now describe the dataset, training specifics, evaluation protocols, and specifics of the parameters used for preliminary experimental evaluation. Figure 4 shows the architecture of our proposed tracking by re-identification method. Specifically, each sample is of size $(k, 11)$, where $k = 10$ for our experiments.

4.1. Dataset

The availability of video data for the purpose of tracking in multi-camera environment is limited. Many commonly evaluated public datasets such as VIPeR [7] and UCY [15] are not suitable as they do not exhibit the characteristics that we are trying to demonstrate, and lack time and motion information. Hence, we use the CamNeT [34] and DukeMTMC [24] datasets for our experiments with the proposed approach.

CamNeT: CamNeT is a non-overlapping camera network tracking dataset in a university campus, covering both indoor and outdoor scenes. It has over 1600 frames, each of resolution 640 by 480 pixels, 20-30fps video, observing more than 25 identities and includes surveillance footage from 5 to 8 cameras. The dataset has six scenarios, each video sequence lasting at least 5 minutes. We use Scenario 1 for our experiments.

DukeMTMC: DukeMTMC is a large multi-camera tracking dataset recorded outdoors on the Duke University campus with 8 synchronized cameras. It consists of more than 2,000 identities, with over 2 million frames of 1080p, 60fps video, approximately 85 minutes of video for each camera. We report results across all 8 cameras to demonstrate the efficiency of our approach for the multi-person multi-camera tracking problem. We use the ground-truth available for the training set (called trainval) of DukeMTMC for evaluation. Only 25% of this set (we call this the 'net-set') is used for CNN and LSTM training.

Training Data: Deep networks need large amounts of training data to avoid overfitting the network. For the purpose of multi-person multi-camera tracking, we synthetically generate samples from real data. We use appearance-based feature scores and location information from CamNeT data

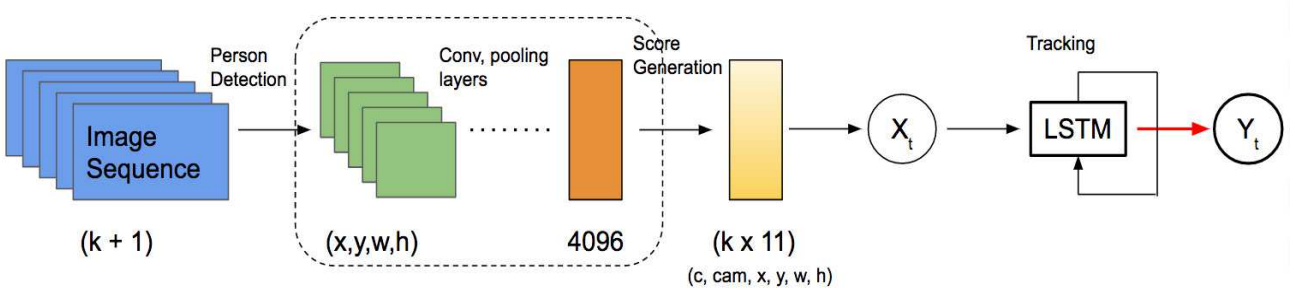


Figure 4: Continuous tracking by re-identification model

to generate sequences of feature vectors. This simple approach of sampling realistic data generates $84,858$ samples for Case 1; $3,900,565$ samples for Case 2 and $18,540,639$ samples for Case 3. We split the data into training set (67%: net-train) and test set (33%: net-test) for our LSTM experiments. Similar approach is followed for DukeMTMC dataset, where $318,769$ samples are generated for Case 1; $4,264,625$ samples for Case 2.

4.2. Training

For extracting appearance features from CamNeT, we use AlexNet model that is pre-trained on ImageNet [25]. Features are extracted from the last fully connected layer (each feature is of length 4096). For DukeMTMC dataset, we use DenseNet trained on net-set. Features of length 1024 are extracted from the last dense layer. The DenseNet used has 4 blocks with a depth of 121, compression/reduction of 0.5 and growth rate of 32. We run for 15 epochs with a learning rate starting at 0.1 and we train using stochastic gradient descent with batch size 96 and momentum 0.9. As the number of remaining epochs halves, we drop learning rate by a factor of 10 and drop by a factor of 2 at epoch 14.

The LSTM architecture employed to learn the associations from history of location and visual features consists of three layers with 512, 256 and 32 units respectively. We use Adam [12] to minimize the loss and choose a learning rate of 0.001. The network is trained for 10 epochs with binary cross-entropy as the objective function. The data is divided into batches of 32 samples and normalized to the range $[-1, 1]$.

4.3. Test Results

We evaluate the performance of our LSTM model in computing the probability of two detections belonging to the same track by means of the ROC curve.

We observe the prediction on the net-test set for three result groups: first, network trained on Case 1 sequences, second, network trained on Case 2 sequences, and third, network trained on Case 3 sequences. Table 5 shows the model prediction for the three different result groups on CamNeT

Scenario	Test Accuracy(%)	TAR@10^{-5}FAR(%)
Case 1	99.31	97.11
Case 2	99.99	99.99
Case 3	99.99	99.99

Table 5: Training scenarios and model prediction on the net-test set for CamNeT

Scenario	Test Accuracy(%)	TAR@10^{-5}FAR(%)
Case 1	99.90	96.26
Case 2	99.98	98.44

Table 6: Training scenarios and model prediction on the net-test set for DukeMTMC

and table 6 shows Case 1 and Case 2 results on DukeMTMC dataset; where TAR is the True Acceptance Rate and FAR is the False Acceptance Rate.

4.4. Evaluation Metric

Since the problem under consideration is similar to [19], existing tracking evaluation metrics such as Multiple Object Tracking Accuracy is not suitable. Here, we use the below metric for continuous re-identification evaluation:

$$E = \frac{1}{T}\sum_{t=1}^{T}\frac{\text{number of misclassified detections at time } t}{\text{total number detections at time } t} \tag{3}$$

Existing Measures: Traditional biometric measures [18] such as FMR (False Match Rate), FNMR (False Non-match Rate) assume that the occurrence of an error is a static event which cannot impact future associations. However, in a re-identification system, the reference gallery is dynamically evolving, as new tracks are created (following "no association" outcomes) or existing tracks are updated (following "association" outcomes). MOTA (Multiple Object Tracking Accuracy) is typically used to measure single-camera, multi-target tracking performance. It is calculated as:

$$MOTA = 1 - (FN + FP + \phi)/T \tag{4}$$

However, $MOTA$ penalizes detection errors and has limitations if extended to multi-camera use [4]. In this paper, we

Approach	Error(%)
Attribute-based [19]	2.9
Ours	**1.37**

Table 7: Inference error rate on CamNeT

Approach	Error(%)
Baseline (BIPCC [24])	4.4
Ours	**3.6**

Table 8: Inference error rate on DukeMTMC

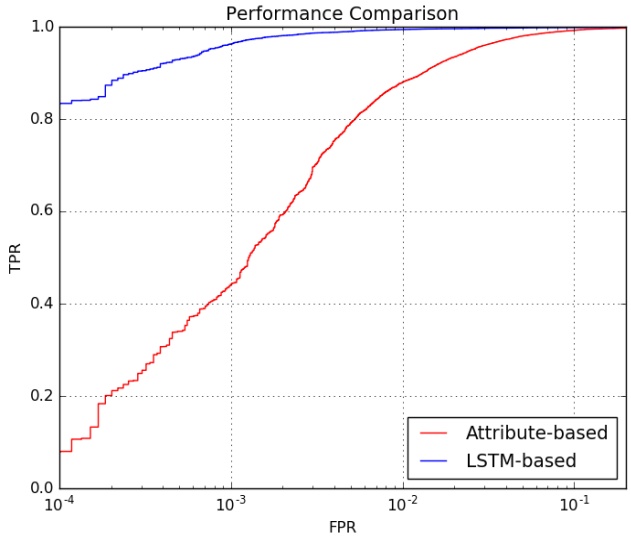


Figure 5: ROC for prediction performance on CamNeT

do not use a private person detector and we would like our tracker's performance evaluated based on association error.

For multi-camera multi-object tracking, the recent measures proposed [24] include Identification F-measure (IDF1), Identificatin Precision (IDP) and Identification Recall (IDR). This evaluation scheme is accountable for how often a target is correctly identified. However, the inference error is a real-time evaluation paradigm which measures how often a target is incorrectly associated. In [5], similar metrics such as false dynamic match (FDM) and false dynamic non-match (FDNM) are proposed for biometric re-identification systems. We do not claim that one measure is better than the other, but only suggest that different error metrics are suited for different applications.

4.5. Inference Results

Table 7 and table 8 show the inference error rate for multi-person multi-camera tracking. Our results are for real CamNeT sequences using the LSTM network trained on Case 3 sequences, and for real DukeMTMC sequences (remaining 75% of trainval) using LSTM trained on Case 2 sequences. The results show that our learning approach is better than the attribute-based approach [19] and baseline method [24] for CamNeT and DukeMTMC respectively.

Figure 5 depicts the prediction performance evaluated using the ROC curve for CamNeT. We observe that, $TAR@0.01FAR = 87.99\%$ using attribute-based inference algorithm, and $TAR@0.01FAR = 99.39\%$ using the proposed algorithm. We also observe that, out of $4,111$ entry/exit scenarios, only 261 instances have been misassociated using the proposed LSTM tracker compared to 609 misassociations using method [19]. This reflects the efficiency of our LSTM-based association approach for person tracking.

4.6. Validation

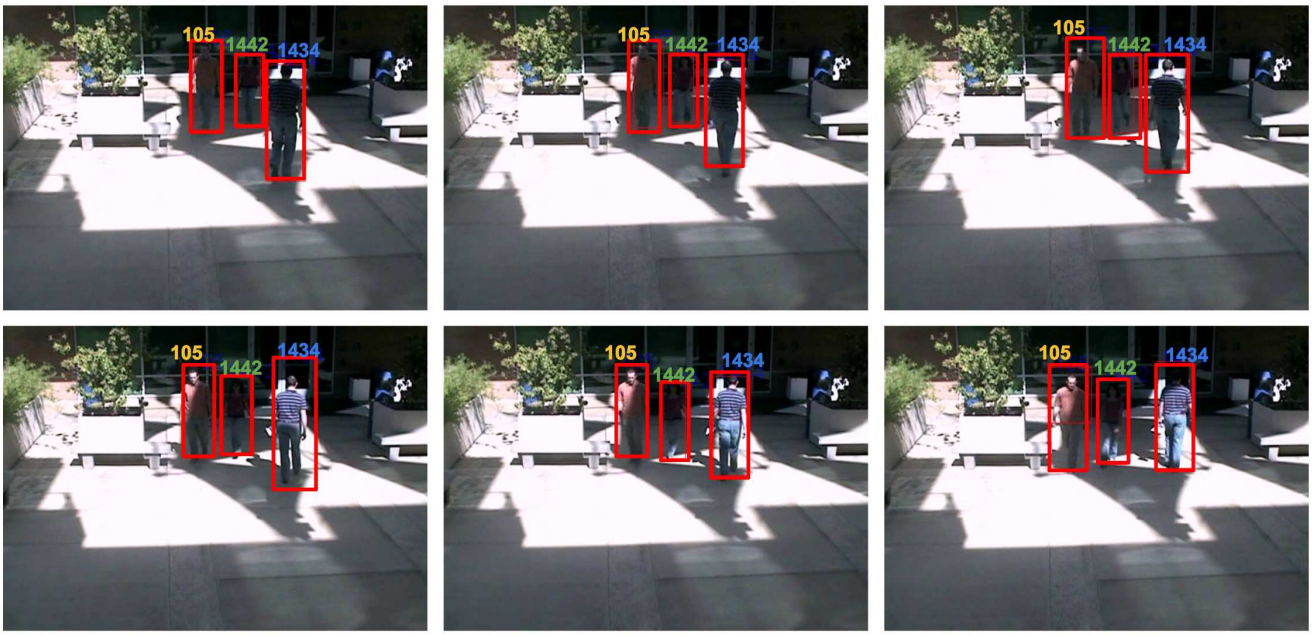
We further tested that the system maintains its performance in cases where people disappeared from one view because they were occluded by objects or because they left the camera's field of view. Our tracker kept the correct identities and we validated that identities obtained in one camera could be used for re-identifying individuals across cameras. For these capabilities, our system outperformed state-of-the art method. The visual tracking results for real-time sequences from CamNeT dataset are shown in Figure 6. The results show that our approach is invariant to illumination changes and can track people reliably for extended duration. Every surveillance dataset has different spatial and temporal dynamics. Our model inherently learns this by using the person's appearance features (spatial) and the transformation of these features with time (temporal). Tracklets (or continuous single detections initially) are merged to form trajectories. Thus the term "continuum", because we know that adjacent detections have important commonalities, although the extremes may differ.

5. Conclusion

We needed three technical elements to obtain an effective tracking system. One is the transformation of the detection of each person into a space in which individuals can be easily identified, even for individuals who change position and posture. Our appearance attributes are one way to do this. A second necessity is an automated procedure to extract features and form trajectories without having to obtain a reference set of frames of the same individual that prevents confusion with other individuals. This makes our tracker more robust and realistic compared to other systems which require a separate video or image of each individual to learn the representation. The third technical element is a LSTM-based system to incorporate the evolution of the frame-by-frame spatial representation with the temporal dependencies. Such a trajectory tracking method can deal with temporally local difficulties generated by cluttered background, multi-object interaction and occlusion. It is capable of recovering from errors (misassociations) and handling entry/exit scenarios. This proves that the estimated

(a) Track of ID #600 and #448

(b) Track of ID #1619



(c) Track of ID #105, #1442 and #1434

Figure 6: Tracking results for real-time CamNeT sequences. Red rectangles are the person detection bounding boxes. The ID on top of each person is the label generated for that individual, reliably maintained across all cameras.

frame-by-frame identity is a good indicator of the correct identity.

Thus, our analysis confirms our intuition regarding the need to propagate past associations. An interesting future research direction is to train our pipeline end-to-end. This has the potential of improving the performance further by producing more discriminative features and consequently, better associations.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant IIP #1266183.

References

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–971, 2016. 2

[2] S. Ali and M. Shah. Floor fields for tracking in high density crowd scenes. *Computer Vision–ECCV 2008*, pages 1–14, 2008. 2

[3] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1515–1522. IEEE, 2009. 2

[4] W. Chen, L. Cao, X. Chen, and K. Huang. An equalized global graph model-based approach for multicamera object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(11):2367–2381, 2017. 6

[5] B. DeCann and A. Ross. Modelling errors in a biometric re-identification system. *IET Biometrics*, 4(4):209–219, 2015. 7

[6] H. Gauvrit, J.-P. Le Cadre, and C. Jauffret. A formulation of multitarget tracking as an incomplete data problem. *IEEE Transactions on Aerospace and Electronic Systems*, 33(4):1242–1257, 1997. 2

[7] D. Gray and H. Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *European conference on computer vision*, pages 262–275. Springer, 2008. 5

[8] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 fps with deep regression networks. In *European Conference on Computer Vision*, pages 749–765. Springer, 2016. 1

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 3

[10] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *European Conference on Computer Vision*, pages 788–801. Springer, 2008. 1

[11] G. Huang and Z. Liu. Densely connected convolutional networks. 3

[12] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 3

[14] B. Leibe, K. Schindler, and L. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007. 2

[15] A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. In *Computer Graphics Forum*, volume 26, pages 655–664. Wiley Online Library, 2007. 5

[16] H. Li, Y. Li, and F. Porikli. Deeptrack: Learning discriminative feature representations online for robust visual tracking. *IEEE Transactions on Image Processing*, 25(4):1834–1848, 2016. 1

[17] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2953–2960. IEEE, 2009. 2

[18] A. J. Mansfield and J. L. Wayman. Best practices in testing and reporting performance of biometric devices. 2002. 6

[19] N. Narayan, N. Sankaran, D. Arpit, K. Dantu, S. Setlur, and V. Govindaraju. Person re-identification for improved multi-person multi-camera tracking by continuous entity association. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*. IEEE, 2017. 2, 6, 7

[20] W. Ouyang and X. Wang. Joint deep learning for pedestrian detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2056–2063, 2013. 2

[21] S. Pellegrini, A. Ess, and L. Van Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *European Conference on Computer Vision*, pages 452–465. Springer, 2010. 2

[22] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. 2

[23] D. Reid. An algorithm for tracking multiple targets. *IEEE transactions on Automatic Control*, 24(6):843–854, 1979. 2

[24] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, pages 17–35. Springer, 2016. 5, 7

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 6

[26] A. Sadeghian, A. Alahi, and S. Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. 2

[27] R. L. Streit and T. E. Luginbuhl. Maximum likelihood method for probabilistic multihypothesis tracking. In *SPIE's International Symposium on Optical Engineering and Photonics in Aerospace Sensing*, pages 394–405. International Society for Optics and Photonics, 1994. 2

[28] L. Wang, W. Ouyang, X. Wang, and H. Lu. Visual tracking with fully convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3119–3127, 2015. 1

[29] N. Wang, J. Shi, D.-Y. Yeung, and J. Jia. Understanding and diagnosing visual tracking systems. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3101–3109, 2015. 2

[30] X. Wang, T. X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 32–39. IEEE, 2009. 2

[31] Y. Xiang, A. Alahi, and S. Savarese. Learning to track: Online multi-object tracking by decision making. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4705–4713, 2015. 1

[32] Q. Xiao, K. Cao, H. Chen, F. Peng, and C. Zhang. Cross domain knowledge transfer for person re-identification. *arXiv preprint arXiv:1611.06026*, 2016. 2

[33] J. Xing, H. Ai, and S. Lao. Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1200–1207. IEEE, 2009. 1

[34] S. Zhang, E. Staudt, T. Faltemier, and A. K. Roy-Chowdhury. A camera network tracking (camnet) dataset and performance baseline. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 365–372. IEEE, 2015. 5