# Online Video Object Detection using Association LSTM

Yongyi Lu
HKUST
yluaw@cse.ust.hk

Cewu Lu
Shanghai Jiao Tong University
lucewu@sjtu.edu.cn

Chi-Keung Tang
HKUST
cktang@cse.ust.hk

## Abstract

*Video object detection is a fundamental tool for many applications. Since direct application of image-based object detection cannot leverage the rich temporal information inherent in video data, we advocate to the detection of long-range video object pattern. While the Long Short-Term Memory (LSTM) has been the* de facto *choice for such detection, currently LSTM cannot fundamentally model object association between consecutive frames. In this paper, we propose the association LSTM to address this fundamental association problem. Association LSTM not only regresses and classifiy directly on object locations and categories but also associates features to represent each output object. By minimizing the matching error between these features, we learn how to associate objects in two consecutive frames. Additionally, our method works in an online manner, which is important for most video tasks. Compared to the traditional video object detection methods, our approach outperforms them on standard video datasets.*

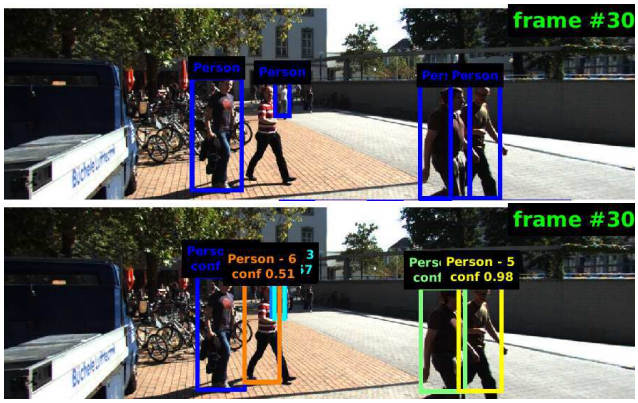


Figure 1. ConvNet-based object detection results (top) and our association LSTM results (bottom). Note that even though the detector produces one missing detection due to a high confidence threshold of 0.9, our association LSTM can recover the false negative successfully thanks to its ability of regressing the category scores and locations as well as associating features between frames.

## 1. Introduction

Object detection assigns a label and a bounding box to detected objects in a single image. With the rapid growth of video data, video object detection has attracted more attention, since it forms the basic tool for various useful video tasks such as action recognition and event understanding. A video provides richer visual information than a still image, and many computer vision applications (e.g., self-driving cars) can only use video but not still images to perform the recognition task.

Comparing to image-based object detection, temporal coherence information in video can be employed to significantly improve the accuracy of object detection. For example, if an object is detected in neighboring frames but not in the current frame, we can recover the missing object in the current frame by applying temporal coherence. Another example is that mistakenly-labeled objects can be corrected by checking the semantic labels across the frames. Therefore, the key to video object detection lies on how to capture the temporal coherence information inherent in the video data, and integrate such information with spatial object detection. Early work such as Kalman filter [9] and particle filter [2] attempted to achieve this goal and had produced reasonable results. However, they still fall short of capturing long-range frames information due to their limited parameter space. With the recent advent in deep learning, recurrent neural network (RNN) has since become a dominating tool for sequence tasks such as sequential learning [6], tracking [19], object recognition [15] and detection [26, 3] thanks to its power in long-range temporal representation.

However, video object detection is a much more challenging problem in comparison to the common sequence problem. First, in detection tasks, RNN should capture multiple objects at the same time, whereas the number of object varies from frame to frame. Second, how to associate object in the RNN structure across multiple frames is a challenging problem. Without a principled way for object association, the power of temporal information cannot be fully utilized.

In this paper, we propose a novel association LSTM framework to advance the state-of-the-arts in video ob-

ject detection. The association RNN receives frame-wise, image-based object detection results (bounding box, score and object feature). Different with traditional LSTM, however, we directly regress object locations and categories, and in the meanwhile produce association features to represent different detected objects. These association features are a representation of the detected objects that captures both spatial and temporal information, since it is a CNN feature filtered by LSTM. Moreover, these representations are partly optimized by minimizing an association error term to encourage that two detections associated with the same object should have a close representation. This good association will improve the information flow across the detected objects in the video, which will in turn encourage the LSTM structure to output high quality association features. In our objective function, the object regression error and association error are jointly optimized.

## 2. Related Work

We survey object detection task, and mainly focus on video object detection.

### 2.1. Image-based Object Detection

The vast development of convolutional neural networks has prompted research into designing different CNN models for object detection. There are two main streams of CNN-based object detection. Most of the methods utilize low-level proposals to first generate candidate boxes, followed by classifying each ROI with state-of-the-art classification models. An optional bounding box regression step is usually done. Typical method is Fast-RCNN [7]. Faster-RCNN [22] replace the proposal generation by a Region Proposal Network (RPN) and is one of the most popular framework for object detection in still images. Sliding window approaches such as Overfeat [23] are another mainstream. One criticism is that these pipelines involve many positions and scale to test.

Some recent approaches posed object detection as a regression problem and directly predict the object classes and locations from predefined sliding window on the feature maps. YOLO [21] suggests dividing an image into grid cells and regressing the bounding box of each object falling into the cells and also their object class scores. SSD [16] extends the single shot detection by utilizing multiple feature map layers. It is the one of the best object detector so far considering the balance between speed and accuracy.

### 2.2. Video Object Detection

Kang *et al*. [10] propose a CNN based framework followed by simple object tracking for detecting object from video. Although this method produced good results and won the ImageNet VID task, it consists of separate stages such as tubelet proposal generation, classification and rescoring, making it less efficient for detection in video. A number of methods adopt recurrent neural networks (RNNs) with long short-term memory (LSTM) cells [8], but not simple tractor. It has been demonstrated especially successful at visual and sequence learning [6], tracking [19], object recognition [15] and detection [26, 3]. The feedback connections and gating mechanism of the LSTM cells enable a model to memorize the spatial dependencies and selectively propagate relevant information under the assumption of temporal smoothness through the network.

Recently, multi-object tracking has become a popular topic that also requires detecting objects and tracking them. Existing appearance-based tracking methods adopt either the generative or discriminative model to separate the foreground from background and distinct co-occurring objects. Bayesian filtering [2] is an online tracking approach that involves state estimation and data association. Though the above methods yield promising results on certain datasets, they usually do not perform well in other datasets, as these methods use low-level hand-crafted features and thus the generalization ability is limited. Xiang *et al*. [28] propose to perform data association using a learning-based process which can be embedded in a reinforcement learning framework. In a recent framework social LSTM [1], each LSTM is added a social pooling layer that pools the hidden states of the neighbors within a spatial radius. The embedded pooled hidden state is then concatenated with the current coordinates to serve as the current input. Though using deep learning approaches, this method only addresses the tracking but not the detection problem. Note that a number of previous work [27, 18] also use RNN with convolutional features to perform refinement task, however, they just concatenate the high-level features with inputs to RNN, rather than defining a particular loss for explicitly optimizing the high-level features when they are fed into the RNN. By contrast, we perform the association task by explicitly defining the association error.

In short, the above approaches do not inadequately address object association across frames. We believe that effective object association is instrumental in utilizing temporal information to its fullest, which is crucial for object detection.

## 3. Approach

We introduce our association LSTM in this section.

In the following we first introduce the architecture, mainly including the input front-end representation and the unconventional output format. Note that our output includes not only object bounding boxes and confident scores but also the novel association features output. Two energy terms (to achieve two goals) imposed on them will be discussed in the training phase. The first term is the object

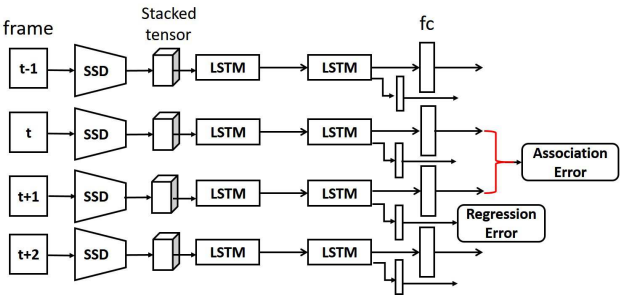


Figure 2. Our LSTM architecture in detail. We solve the two complementary tasks, namely regression and association in a unified framework by defining the regression error and association error.

regression term, which is used to predict object bounding boxes and confident scores. The second term is the association term, which is used to encourage object association. Finally, we describe how to combine these two terms into a unified objective function.

We let $c$ be the number of object categories, and $N$ be the maximum number of objects.

## 3.1. Architecture

Our association LSTM is an unconventional architecture. Our input and output structures are detailed as follows.

### 3.1.1 Input: Front-end Representation

Our association LSTM receives spatial information from the input frames. We apply the SSD [16] detector to extract objects in the frames, since SSD is a fast one-shot detector with good accuracy and speed performance. We keep the result of SSD as a $(c+4)$-dimension location-score vector. Specifically, the vector is formed by the $c$ object category scores and the 4 location parameters. For reliable tracking, we only keep the output boxes whose confidence score is above a threshold of 0.8.

Besides the location-score vector, we also extract a fixed-size descriptor for each detected object using RoI pooling [7]. Inspired by [3], each detected box is max-pooled from several layers to produce a fixed-length $s \times s$ feature descriptor, where the sub-window $s$ is set to be 7 in our experiment. We also add a normalization process to force features from different scales to lie on the same unit sphere.

Now for each detected object, we have a location-score vector of dimension $(c+4)$ and descriptor vector of dimension $(s \times s)$. We concatenate the location-score vector and and feature descriptor vector of different objects respectively into a frame-level tensor with two dimension $N \times D$, where $D = c + 4 + s \times s$ is the composite feature length for each detected object. When the number of detected objects is smaller than $N$, we pad the missing elements with zeros. Otherwise, we only take top $N$ objects to form the frame-level vector. To make use of the past frames so that LSTM can selectively remember what is useful for prediction, we stack the current frame-level tensor with the $\tau - 1$ frames backward, yielding a stacked tensor input with size $\tau \times N \times D$. The resulting frame-level vector will be fed into our LSTM architecture. Note that [27] also utilizes past frames as input to the network.

### 3.1.2 Output: Object Prediction and Association Feature

Given a frame-level tensor $\mathbf{x} \in \mathcal{R}^{\tau \times N \times D}$ in frame $t$, our association LSTM outputs the improved predictions $\mathbf{x}$ with the same size, in which we only consider the predictions for current frames $\hat{\mathbf{y}}_t$ with respect to the ground truth $\mathbf{y}_t$. Here $\hat{\mathbf{y}}_t$ consist of $N$ object predictions. We assume the ground truth objects in a frame do not exceed $N$ in number. In the case where the number of ground truth objects is smaller than $N$, we pad zeros into the vector to keep fixed all the output object predication numbers. Each object prediction include three items, namely, object location in 4D, category score vector with dimension $c$, and association feature with dimension $s \times s$. The network structure is shown in Figure 2.

Compared to the traditional LSTM, where a concise review is given in the appendix, our designed network can jointly solve the object regression and object association problem given the input tensor. The regressed category scores and object locations are obtained from the LSTM output hidden state in each timestep, while the association features are computed between the output hidden states in two consecutive timesteps (see Figure 3). Note that the hidden state in each frame solely depends on the network's current input and its hidden state in time $t-1$. We solve these three tasks jointly by carefully designing their loss functions, which we will address in Section 3.2. For the network structure, in order to accelerate the network training, we normalize each layer's output by adopting the recently proposed Batch Normalized LSTM [5]:

$$\begin{pmatrix} \tilde{f}_t \\ \tilde{i}_t \\ \tilde{o}_t \\ \tilde{g}_t \end{pmatrix} = \mathrm{BN}(\mathbf{W}_h \mathbf{h}_{t-1}; \gamma_h, \beta_h) + \mathrm{BN}(\mathbf{W}_x \mathbf{x}_t; \gamma_x, \beta_x) + \mathbf{b}$$
$$\mathbf{c}_t = \sigma(\tilde{\mathbf{f}}_t) \odot \mathbf{c}_{t-1} + \sigma(\tilde{\mathbf{i}}_t) \odot \tanh(\tilde{\mathbf{g}}_t)$$
$$\mathbf{h}_t = \sigma(\tilde{\mathbf{o}}_t) \odot \tanh(\mathrm{BN}(\mathbf{c}_t; \gamma_h, \beta_h)) \quad (1)$$

where $f, i, o$ are gates of a cell (see appendix). Batch normalization is applied to the hidden states of the previous time step $\mathbf{W}_h \mathbf{h}_{t-1}$, the input tensor $\mathbf{W}_x \mathbf{x}_t$ and the memory cell state $\mathbf{c}_t$. The batch normalization function $\mathrm{BN}(\mathbf{h}; \gamma, \beta)$ normalizes the values of $\mathbf{h}$ to zero mean and standard deviation 1, where the average and variance of $\mathbf{h}$ are computed in

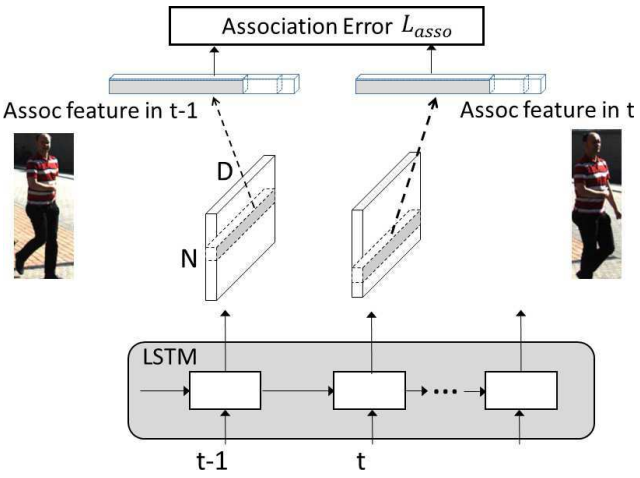

Figure 3. Illustration of the proposed feature association and association error.

the current minibatch for the current time step. As a result, we force the output features in each time step from different scales to lie on the same unit sphere. As a result, the association is more accurate and unbiased to different scales.

## 3.2. Two Goals

We design the two energy terms on the output of LSTM to achieve two goals, namely, object regression and object association. Then, we will discuss the joint minimization.

### 3.2.1 Object Regression Error

In designing the regression loss, we need to consider both the accuracy at the regressed box locations and of the class scores as well. The regression error in the proposed LSTM is different from what we use for the front-end detector in that it calculate both the object location error and object class classification error. We also add a smoothness constraint between consecutive frames within range $\tau$ to guarantee that the output box locations are consistent across the frames. Thus our regression loss function consists of three components:

$$\mathcal{L}_{reg}(l, g, c) = \sum(L_{conf}(c, c^*) + \lambda L_{loc}(l, g)) + \alpha \cdot \mathcal{L}_{smooth} \quad (2)$$

The first two terms are the object regression error, where the localization loss $L_{loc}$is a smooth L1 loss between the predicted box ($l$) and ground truth box ($g$). The confidence loss $L_{conf}$ is the softmax loss over multiple classes confidences $c$ toward ground truth score vector $c^*$. These two losses are adapted from [16]. Moreover, we regularize the LSTM model by applying the smoothness constraint across consecutive time-steps.

$$\mathcal{L}_{smooth} = \sum_{\tau}(\tilde{l}_t - \tilde{l}_{t+1}) \quad (3)$$

Note that our location loss and confidence loss are computed only on the last stacked frame, and the smoothness term is computed across consecutive frames within range $\tau$. With this training strategy, both fully labeled as well as weakly labeled datasets can be used to train our LSTM model. We show in our experiments that our new LSTM is capable of regressing the object location and score, even though they are highly non-linear.

### 3.2.2 Association Error

In video object detection, we need to detect each isolated target and maintain their identities. Recent work has pointed out that data association is a matching problem [13]. Inspired by the siamese network [24] for matching pairwise object, we describe how to pose this association problem as pairwise feature matching using the filtered association features. We perform the association task by defining an association error [24]. The difference with [24] is that we associate multiple objects in a video, while [24] deal with pairs only.

The association error has to achieve two objectives at the same time: we want the network to output features representations which are close enough by the positive pairs, while they should stay as far as possible from the negative pairs. As before, we define $\tilde{f}_t = \{\phi_t^1, \ldots, \phi_t^N\}$, where $\phi_t^i$ is the $i^{th}$ detected objected association feature in frame $t$.

We migrate the margin contrastive loss to solve multiple object association problem:

$$\mathcal{L}_{asso} = \sum_t \sum_{i,j} \theta_{ji} |\phi_{t-1}^i \cdot \phi_t^j| \quad (4)$$

where $\theta_{jk} \in 0, 1$ is an indicator, $\theta_{ji} = 1$ if and only if object $i$ in frame $t-1$ is associated with $j^{th}$ object in frame $t$, *i.e.*, they have the smallest distance among all pairs. $\cdot$ is a dot product operation. Note that the association features are normalized (in Section 3.1.2) for computing the association loss.

We impose the conventional constraint that each target (object i in frame t-1) can only be assigned to one observation (object j in frame t), i.e. $\forall i : \sum_j \theta_{ji} = 1$. The indicator $\theta_{ji}$ is computed using the input association features and the spatial distance of the corresponding objects. Then by defining association error in Equation 4 based on $\theta_{ji}$, the association features are updated in return. We adopted the traditional Hungarian algorithm to solve the assignment problem in each time step of our LSTM.

## 3.3. Objective Function and Training

We solve the two complementary tasks, namely regression and association in a unified framework. Specifically, we combine these two loss functions together as our final

objective function, which is weighted by the hyperparameter $\xi$, defined using grid search strategy according to the performance on the validation set. We set it to 1 in our experiments.

$$\mathcal{L} = \mathcal{L}_{reg} + \xi \mathcal{L}_{asso} \tag{5}$$

### 3.4. Testing

Testing is straightforward: for each input sequence, and for each forward computation at time step $t$ and $t+1$, we match their association features using Equation 4 and the matched pairs are assigned the same target identities. The LSTM network also generates the regressed detection output including bounding boxes and their label scores. Note that our framework is an online approach which does not utilize future information.

### 3.5. Implementation Details

**Choosing Pooling Layers** Rich visual information is inherent in high-level visual features as the colors and the rough contour of an image object can be reconstructed from the activations in higher network layers. This factors into our choice of ROI pooling layers for feature extraction described in Section 3.1.1. The insight is that including lower level features will not help much as they are quite large and not sufficiently discriminative ($38 \times 38$ for $conv4\_3$), while the last global average pooling ($pool6$) is too coarse in resolution. To produce clearer evidence and facilitate our selection of feature maps to add into the input tensor, we first find layers which activates to each high confidence object in VOC test sets with the pre-trained SSD model. Noting that most of the objects are detected using "$fc7$", "$conv6\_2$", "$conv7\_2$" and "$conv8\_2$" feature maps. Thus, we utilize these four feature maps as ROI pooling layers and pool the output boxes onto these feature maps to compute fixed-size feature descriptor.

**Data Augmentation** We apply the following data augmentation: 1) mirroring and random crop with IoU $> 0.8$, 2) reversed sequence for each trajectory with probability 0.5, resulting in the same effect of backward tracking. These augmentations further boost our performance on the two public datasets ($c.f.$ Section 4).

**Training** For training, we first perform sampling on video snippets; otherwise training would be biased toward longer clips. We train a two-layer LSTM model by minimizing the above defined loss using back-propagation through time and RMSProp with a initial learning rate of $10^{-3}$ and a decay rate of 0.9 for 200 epochs. Both LSTM models adopt two-layer stateful LSTM. For state estimation, the hidden units are set to be 150 while the LSTM for data association is set to be 300 hidden units. Also, we perform early-stopping on the validation set. The hyperparameters $\tau = 30$, $\lambda = 1$, $\alpha = 0.1$, $\gamma = 1$ are chosen based on the performance on the validation set.

## 4. Experiments

In this section we evaluate our algorithm on three public datasets, Youtube-Object dataset, 2DMOT dataset, providing both quantitative and qualitative results of the model's performance.

### 4.1. Youtube-Object Evaluation

The Youtube-Object dataset [27, 20] contains 155 videos from 10 subclasses of the Pascal VOC Challenge. However, it only contains 6087 annotated frames; among them 4306 are for training and 1781 for testing. As stated above, our designed regression loss in Eq. (2) makes it feasible in training with such weakly supervised dataset.

We adopt cross validation and split the data into training and validation sets, and we report the final results on the test set. We utilize the Pascal VOC 20-class pre-trained SSD model and perform domain adaptation by fine-tuning on the training set. For evaluation criterion, since Youtube-Object aims at detecting objects in video, the dataset contains no ground truth for target IDs.

We compare the proposed method on Youtube-Object datasets with four competitors [21, 26, 12, 27]. Both [27] and our methods are based on neural network and both use the pretrained VOC 20-class model. We follow the popular mean AP metric for detecting objects in video. Quantitative results are shown in Table 1. Here $a\_LSTM$ denotes our proposed association LSTM. We also compare with two baselines. $Baseline\_1$ uses only SSD to detect object in each frame without LSTM regression. $Baseline\_2$ adds a post-processing using a keypoint-based tracker [17], which is robust in long-term tracking. From the table, we can see the following characteristics of our proposed framework. 1) Thanks to the powerful one shot object detector, the finetuned SSD model already performs much better than YOLO, but still inferior to [27], which use GRU for boxes and scores refinements. 2) Our full version of association LSTM outperforms [27] by $\sim 3.5\%$, which is the state-of-the-art using RNN architecture for video object detection. This illustrates the efficacy of our designed joint object regression and feature association. Also our method is slightly better than the widely used detection plus post-processing approach ($Baseline\_2$) because we consider multiple objects in association, while $Baseline\_2$ only uses single object tracking for refinement with the assumption that all objects are moving independently, which is not always correct, e.g., in crowd scene.

| Methods | mAP | airplane | bird | boat | car | cat | cow | dog | horse | mbike | train |
|---|---|---|---|---|---|---|---|---|---|---|---|
| VOP [25] | 37.41 | 29.77 | 28.82 | 35.34 | 41.00 | 33.7 | 57.56 | 34.42 | 54.52 | 29.77 | 29.23 |
| Unsupervised [12] | 55.7 | 56.5 | 66.4 | 58.0 | 76.8 | 39.9 | 69.3 | 50.4 | 56.3 | 53.0 | 31.0 |
| YOLO [21] | 56.53 | 76.67 | 89.51 | 57.66 | 65.52 | 43.03 | 53.48 | 55.81 | 36.96 | 24.62 | 62.03 |
| Context [27] | 68.73 | 76.11 | 87.65 | 62.16 | 80.69 | 62.42 | 78.02 | 58.72 | 81.77 | 41.54 | 58.23 |
| $Baseline\_1$ | 66.21 | 74.89 | 85.03 | 60.11 | 77.63 | 61.22 | 77.56 | 56.91 | 80.18 | 40.67 | 54.83 |
| $Baseline\_2$ | 70.43 | 77.14 | 91.02 | 63.34 | 81.70 | 63.47 | 79.38 | 59.18 | 83.56 | 42.33 | 59.80 |
| $a\_LSTM$ | **72.14** | 78.92 | 90.94 | 65.87 | 84.76 | 65.22 | 81.39 | 61.86 | 83.27 | 43.92 | 61.25 |

Table 1. Per-category object detection results on the Youtube-Objects datasets. $a\_LSTM$ denotes our proposed association LSTM.

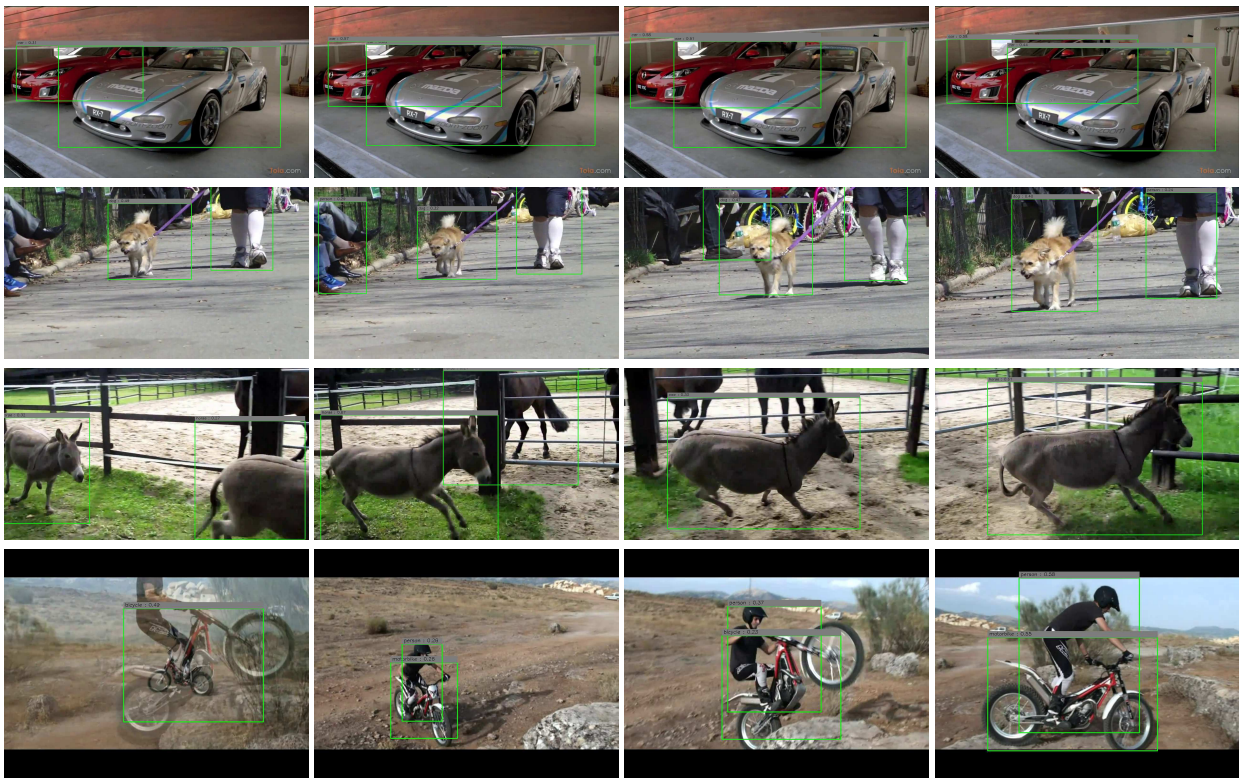

Figure 4. Qualitative video object detection results on Youtube-Object dataset from the sparsely sampled four different sequences. Each row shows the results of our proposed LSTM. In the first two examples, the RNN consistently predicts correct categories car and dog. In the third and fourth rows, drastic motion and body deformation are the main difficulties in video. With the help of the proposed LSTM, all the objects are successfully detected. The figure is best viewed in color.

### 4.2. MOTChallenge Evaluation

We evaluate the association performance of our model on the MOTChallenge dataset. This is a popular dataset for multi-object tracking consisting of 22 video sequences (11 for training and 11 for testing, respectively) with different view points, target motion, camera motion and person density. Since the annotations are available for the training sequences only, we use the training/validation split of [28] for training and testing to analyze the proposed association LSTM. Specifically, we separate 6 sequences from the 11 training sequences for validation; detailed splitting of the dataset is shown in Table 2.

For quantitative comparison on the MOTChallenge dataset, we adopt the widely-used CLEAR MOT [4] and five other metrics: Mostly-Tracked (MT) / Mostly-Lost (ML) / False Positive (FP) / False Negative (FN) / ID Switches (IDS) [14] metrics. The definition of the metrics are shown in Table 3. Note that these metrics address tracking performance rather than detection performance, in contrast to the mAP we use in Youtube-Objects dataset.

We compare with two state-of-the-arts [28, 11]. Instead of using ConvNet, [28] regards the data association problem as policy learning which is approached in a reinforcement learning fashion. In [11], model-free tracking is conducted in both forward and backward direction to restore undetected objects. Both methods use precomputed object

| Training | Testing |
|---|---|
| TUD-Stadtmitte | TUD-Campus |
| ETH-Bahnhof, PETS09-S2L1 | ETH-Sunnyday, ETH-Pedcross2 |
| ADL-Rundle-6 | ADL-Rundle-8, Venice-2 |
| KITTI-13 | KITTI-17 |

Table 2. Training and Testing sequences for validation in the MOTChallenge.

| Metric | Definition |
|---|---|
| MOTA ↑ | It considers false positives, false negatives and ID-Switches. |
| MOTP ↑ | It measures the tightness of the tracking results and ground truth. |
| MT ↑ | More than 80% of tracks are successfully tracked. |
| ML ↓ | Less than 20% of tracks are successfully tracked. |
| FP ↓ | The total number of false positives. |
| FN ↓ | The total number of false negatives (missed targets). |
| IDS ↓ | Total number of times that an output track changes its identity. |

Table 3. Evaluation metrics for comparisons in multi-object tracking frameworks.

detection outputs for data association. Noting that [11] also first collects detection results with high confidence levels to reduce spurious objects. Table 4 shows the quantitative results. From the results, our method produces the best performance in ML (46.8 compared to the second 47.6) which means that the proposed method that manages to track objects in most videos successfully due to accurate association between targets. Also the association features act as online-update appearance models in tracking, which prevents target lost and demonstrates the efficacy of the proposed feature association scheme. Our method also performs well in FP thanks to the high confident front-end SSD detector. The main issue is ID switches: our proposed framework is online which lacks explicit occlusion reasoning and global optimization.

Noting that [28] adopts public detector while [11] and our proposed method use private detectors. We solved two complementary tasks, i.e. regression and association in a unified framework, in stark contrast with [28, 11]. Our front-end detector generates stacked tensor in a single shot, combining locations, categories and feature descriptor. [11], on the other hand, use Fast RCNN with VGG16, which performs similarly with SSD trained on the Pascal dataset.

We also present qualitative results in Figure 5. The color of the boxes represents the identity of the targets. Row 1 to 3 are respectively ground truth annotations, SSD detection outputs and our proposed association LSTM outputs. We can see that our association network is able to recover missing detections caused by CNN detector (Person-5 in frame 35 and Person-6 in frame 30), as well as reducing spurious-detections (one false positive has been deleted in frame 35). Note that our proposed method can also refine bounding box locations resulting in more accurate output boxes.

### 4.3. Component Evaluation

We present some empirical analysis to show the benefits of each component in our framework, *i.e.*, the object regression and feature association. Additional results are generated by disabling some components of our framework, as tabulated in Table 5 reports. We use the Youtube-Objects for evaluation here since by disabling feature association, we can only investigate the performance using mAP. In the table, the last row denotes our full version of association LSTM (with mAP 72.14), while row 1 to row 3 are simplified versions by disabling the chosen components. Specifically, row 3 denotes cutting the feature association stage, *i.e.*, only update the network using the regression error (with mAP 70.2). Row 2 denotes without using data augmentation, *i.e.*, reversing the sequence for the training data which acts like backward tracking (with mAP 69.08). Row 1 denotes further cutting off the regression network, which is exactly the $Baseline\_1$ in 4.1 (with mAP 66.21). From the results, we conclude that both the object regression and the feature association components play important roles in the proposed framework. Combining both gives a notable improvement ($\sim 5\%$) against baseline methods.

### 4.4. Runtime Performance

Our implementation is coded on an Intel I7 4.0 GHz PC with one Titan X 12 GB GPU and 32GB memory. The front-end one shot SSD runs 33 FPS, while the proposed association LSTM runs 12 FPS on both datasets in the testing stage, which is nearly real time.

## 5. Conclusions and Future Work

We propose a novel association LSTM framework to advance the state-of-the-arts in video object detection. The association RNN receives frame-wise image-based object detection results (bounding box, score and object feature). Different from the traditional LSTM, however, we directly regress object locations and categories, and in the meanwhile produce association features to represent detected objects. These representations are partly optimized by minimizing an association error. In our objective function, object regression error and association error are jointly optimized. Experimental results show that this good association improves the information flow across the detected objects in the video, which in turn encourage the LSTM structure to output high quality association features.

The weakness of our approaches is that our LSTM modules are post-hoc: CNN features are not updated in re-

| Methods | MOTA ↑ | MOTP ↑ | MT ↑ | ML ↓ | FP ↓ | FN ↓ | IDS ↓ |
|---|---|---|---|---|---|---|---|
| MDP (K=5) [28] | 26.7 | 73.7 | 12.0 | 53.0 | 3,386 | 13,415 | 111 |
| MDP (K=9) [28] | 26.7 | 73.6 | 12.0 | 51.7 | 3,290 | 13,491 | 133 |
| CDT [11] | 39.9 | 74.8 | 20.9 | 47.4 | 914 | 12,856 | 95 |
| $a\_LSTM$ | 38.6 | 74.2 | 14.9 | **46.8** | **788** | 13,253 | 154 |

Table 4. Tracking performance in the MOTChallenge validation set.

.

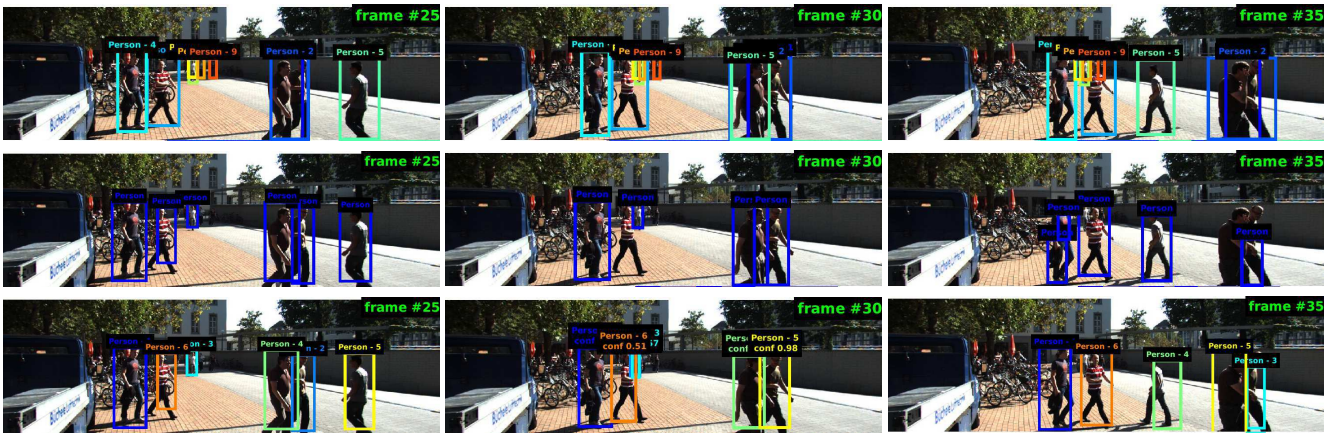


Figure 5. Qualitative results on a sequence from the MOTChallenge dataset. The top shows the ground truth. The middle shows object detection responses of [16], and the bottom shows that we can handle occlusion, moving camera and change of scale. The figure is best viewed in color.

| Settings | Regression | Reverse Seq | Association | mAP |
|---|---|---|---|---|
| $Baseline\_1$ | | | | 66.21 |
| $Baseline\_1$ | √ | | | 69.08 |
| $Baseline\_1$ | √ | √ | | 70.2 |
| $Baseline\_1$ | √ | √ | √ | 72.14 |

Table 5. Component analysis. We present three additional results on Youtube-Objects by disabling the corresponding components of our approach.

sponse to the outputs of the LSTM. Our future work includes adding a feedback loop to the pre-trained CNN so that the weights of the feature extractor can also be updated online guided by examining forward/backward temporal context provided by the RNN modules.

## A. Long Short Term Memory (LSTM)

LSTM unit [8] plays an important role in modern deep RNN architecture for its ability to access long-range context in sequence. Conventional RNNs suffer from the vanishing gradient problem, which means that the gradient is either exploding or vanishing when back propagated through time. By contrast, LSTM overcomes this problem by introducing a memory cell structure and gating mechanism. A cell consists of three gates (input, forget and output). Gates can decide whether to let information through. Given an input $x_t$, its information will be forwarded to the cell if the input gate $i_t$ is activated. Meanwhile old cell state $c_{t-1}$ can be selectively forgotten by turning off the forget gate $f_t$, so that no information will contribute to the update of new cell state $c_t$. Whether the updated cell state $c_t$ will be forwarded to the output hidden state $h_t$ is further controlled by the output gate $o_t$. In our proposed framework, we use LSTM as the building block in both state estimation module and data association module, in order to discover long-range temporal relation when new states are predicted. We also use memory cells to store and output label assignment relations. We first define their update equations in the following commonly-used rules given inputs $z_t$, $h_{t-1}$ and $c_{t-1}$:

$$
\begin{aligned}
i_t &= \sigma(W_{zi}z_t + W_{hi}h_{t-1} + w_{ci} \odot c_{t-1} + b_i) \\
f_t &= \sigma(W_{zf}z_t + W_{hf}h_{t-1} + w_{cf} \odot c_{t-1} + b_f) \\
o_t &= \sigma(W_{zo}z_t + W_{ho}h_{t-1} + w_{co} \odot c_{t-1} + b_o) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(W_{zc}z_t + W_{hc}h_{t-1} + b_c) \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
\tag{6}
$$

**Acknowledgment**: The authors would like to thank NVIDIA for the GPU support (Titan X Pascal). The research is partly supported by the Google Faculty Research Award.

## 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[2] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188, 2002.

[3] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[4] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008(1):1–10, 2008.

[5] T. Cooijmans, N. Ballas, C. Laurent, and A. Courville. Recurrent batch normalization. *arXiv preprint arXiv:1603.09025*, 2016.

[6] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[7] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

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

[9] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[10] K. Kang, W. Ouyang, H. Li, and X. Wang. Object detection from video tubelets with convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[11] H.-U. Kim and C.-S. Kim. Cdt: Cooperative detection and tracking for tracing multiple objects in video sequences. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[12] S. Kwak, M. Cho, I. Laptev, J. Ponce, and C. Schmid. Unsupervised object discovery and tracking in video collections. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3173–3181, 2015.

[13] L. Leal-Taixé, C. Canton-Ferrer, and K. Schindler. Learning by tracking: Siamese cnn for robust target association. *arXiv preprint arXiv:1604.07866*, 2016.

[14] 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.

[15] M. Liang and X. Hu. Recurrent convolutional neural network for object recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed. Ssd: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2015.

[17] G. Nebehay and R. Pflugfelder. Consensus-based matching and tracking of keypoints for object tracking. In *IEEE Winter Conference on Applications of Computer Vision*, pages 862–869. IEEE, 2014.

[18] G. Ning, Z. Zhang, C. Huang, Z. He, X. Ren, and H. Wang. Spatially supervised recurrent convolutional neural networks for visual object tracking. *arXiv preprint arXiv:1607.05781*, 2016.

[19] P. Ondruska and I. Posner. Deep tracking: Seeing beyond seeing using recurrent neural networks. *arXiv preprint arXiv:1602.00991*, 2016.

[20] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3282–3289. IEEE, 2012.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[22] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[23] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[24] R. Tao, E. Gavves, and A. W. Smeulders. Siamese instance search for tracking. *arXiv preprint arXiv:1605.05863*, 2016.

[25] S. Tripathi, S. Belongie, Y. Hwang, and T. Nguyen. Detecting temporally consistent objects in videos through object class label propagation. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016.

[26] S. Tripathi, Z. C. Lipton, S. Belongie, and T. Nguyen. Context matters: Refining object detection in video with recurrent neural networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2016.

[27] S. Tripathi, Z. C. Lipton, S. Belongie, and T. Nguyen. Context matters: Refining object detection in video with recurrent neural networks. *BMVC*, 2016.

[28] Y. Xiang, A. Alahi, and S. Savarese. Learning to track: Online multi-object tracking by decision making. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.