# Social Scene Understanding: End-to-End Multi-Person Action Localization and Collective Activity Recognition

Timur Bagautdinov[1], Alexandre Alahi[2], François Fleuret[1,3], Pascal Fua[1], Silvio Savarese[2]
[1]École Polytechnique Fédérale de Lausanne (EPFL)
[2]Stanford University
[3]IDIAP Research Institute
{timur.bagautdinov, francois.fleuret, pascal.fua}@epfl.ch, {alahi, ssilvio}@stanford.edu

## Abstract

*We present a unified framework for understanding human social behaviors in raw image sequences. Our model jointly detects multiple individuals, infers their social actions, and estimates the collective actions with a single feed-forward pass through a neural network. We propose a single architecture that does not rely on external detection algorithms but rather is trained end-to-end to generate dense proposal maps that are refined via a novel inference scheme. The temporal consistency is handled via a person-level matching Recurrent Neural Network. The complete model takes as input a sequence of frames and outputs detections along with the estimates of individual actions and collective activities. We demonstrate state-of-the-art performance of our algorithm on multiple publicly available benchmarks.*

## 1. Introduction

Human social behavior can be characterized by "social actions" – an individual act which nevertheless takes into account the behaviour of other individuals – and "collective actions" taken together by a group of people with a common objective. For a machine to perceive both of these actions, it needs to develop a notion of collective intelligence, *i.e.*, reason jointly about the behaviour of multiple individuals. In this work, we propose a method to tackle such intelligence. Given a sequence of image frames, our method jointly locates and describes the social actions of each individual in a scene as well as the collective actions (see Figure 1). This perceived social scene representation can be used for sports analytics, understanding social behaviour, surveillance, and social robot navigation.

Recent methods for multi-person scene understanding take a sequential approach [20, 10, 28]: i) each person is detected in every given frame; ii) these detections are associated over time by a tracking algorithm; iii) a feature representation is extracted for each individual detection; and finally iv) these representations are joined via a structured model. Whereas the aforementioned pipeline seems reasonable, it has several important drawbacks. First of all, the vast majority of state-of-the-art detection methods do not use any kind of joint optimization to handle multiple objects, but rather rely on heuristic post-processing, and thus are susceptible to greedy non-optimal decisions. Second, extracting features individually for each object discards a large amount of context and interactions, which can be useful when reasoning about collective behaviours. This point is particularly important because the locations and actions of humans can be highly correlated. For instance, in team sports, the location and action of each player depend on the behaviour of other players as well as on the collective strategy. Third, having independent detection and tracking pipelines means that the representation used for localization is discarded, whereas re-using it would be more efficient. Finally, the sequential approach does not scale well with

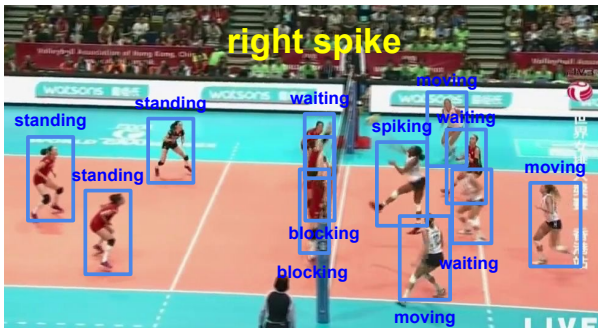



Figure 1. Jointly reasoning on social scenes. Our method takes as input raw image sequences and produces a comprehensive social scene interpretation: locations of individuals (as bounding boxes), their individual social actions (e.g., "blocking"), and the collective activity ("right spike" in the illustrated example).

the number of people in the scene, since it requires multiple runs for a single image.

Our method aims at tackling these issues. Inspired by recent work in multi-class object detection [30, 29] and image labelling [23], we propose a single architecture that jointly localizes multiple people, and classifies the actions of each individual as well as their collective activity. Our model produces all the estimates in a single forward pass and requires neither external region proposals nor pre-computed detections or tracking assignments.

Our contributions can be summarized as follows:

- We propose a unified framework for social scene understanding by simultaneously solving three tasks in a single feed forward pass through a Neural Network: multi-person detection, individual's action recognition, and collective activity recognition. Our method operates on raw image sequences and relies on joint multi-scale features that are shared among all the tasks. It allows us to fine-tune the feature extraction layers early enough to enable the model to capture the context and interactions.
- We introduce a novel multi-object detection scheme, inspired by the classical work on Hough transforms. Our scheme relies on probabilistic inference that jointly refines the detection hypotheses rather than greedily discarding them, which makes our predictions more robust.
- We present a person-level matching Recurrent Neural Network (RNN) model to propagate information in the temporal domain, while not having access to the the trajectories of individuals.

In Section 4, we show quantitatively that these components contribute to the better overall performance. Our model achieves state-of-the-art results on challenging multi-person sequences, and outperforms existing approaches that rely on the ground truth annotations at test time. We demonstrate that our novel detection scheme is on par with the state-of-the art methods on a large-scale dataset for localizing multiple individuals in crowded scenes. Our implementation will be made publicly available.

## 2. Related Work

The main focus of this work is creating a unified model that can simultaneously detect multiple individuals and recognize their individual social actions and collective behaviour. In what follows, we give a short overview of the existing work on these tasks.

**Multi-object detection** - There already exists large body of research in the area of object detection. Most of the current methods either rely on a sliding window approach [31, 41], or on the object proposal mechanism [17, 30], followed by a CNN-based classifier. The vast majority of those state-of-the-art methods do not reason jointly on the presence of multiple objects, and rely on very heuristic post-processing steps to get the final detections. A notable exception is the ReInspect [35] algorithm, which is specifically designed to handle multi-object scenarios by modeling detection process in a sequential manner, and employing a Hungarian loss to train the model end-to-end. We approach this problem in a very different way, by doing probabilistic inference on top of a dense set of detection hypotheses, while also demonstrating state-of-the-art results on challenging crowded scenes. Another line of work that specifically focuses on joint multi-person detection [15, 3] uses generative models, however, those methods require multiple views or depth maps and are not applicable in monocular settings.

**Action recognition** - A large variety of methods for action recognition traditionally rely on handcrafted features, such as HOG [9, 40], HOF [26] and MBH [38]. More recently, data-driven approaches based on deep learning have started to emerge, including methods based on 3D CNNs [22] and multi-stream networks [14, 33]. Some methods [39, 34], exploit the strengths of both handcrafted features and deep-learned ones. Most of these methods rely in one way or another on temporal cues: either through having a separate temporal stream [14, 34], or directly encoding them into compact representations [26, 38, 38]. Yet another way to handle temporal information in a data-driven way is Recurrent Neural Networks (RNNs). Recently, it has received a lot of interest in the context of action recognition [33, 12, 37, 11]. All these methods, however, are focusing on recognizing actions for single individuals, and thus are not directly applicable in multi-person settings.

**Collective activity recognition** - Historically, a large amount of work on collective activity recognition relies on graphical models defined on handcrafted features [6, 7, 2]. The important difference of this type of methods with the single-person action recognition approaches is that they explicitly enforce simultaneous reasoning on multiple people. The vast majority of the state-of-the-art methods for recognizing multi-person activities thus also rely on some kind of structured model, that allows sharing information between representations of individuals. However, unlike earlier handcrafted methods, the focus of the recent developments has shifted towards merging the discriminative power of neural networks with structured models. In [10], authors propose a way to refine individual estimates obtained from CNNs through inference: they define a trainable graphical model with nodes for all the people and the scene, and pass messages between them to get the final scene-level estimate. In [20], authors propose a hierarchical model that takes into account temporal information. The model consists of two LSTMs: the first operates on person-level representations, obtained from a CNN, which are then max pooled and

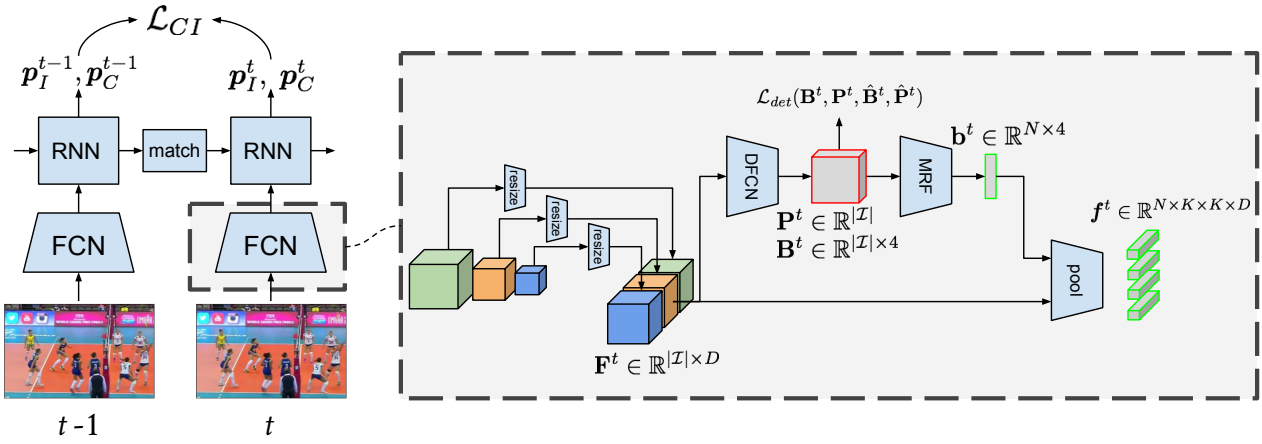



Figure 2. General overview of our architecture. Each frame of the given sequence is passed through a fully-convolutional network (FCN) to produce a multi-scale feature map $\mathbf{F}^t$, which is then shared between the detection and action recognition tasks. Our detection pipeline is another fully-convolutional network (DFCN) that produces a dense set of detections $\mathbf{B}^t$ along with the probabilities $\mathbf{P}^t$, followed by inference in a hybrid MRF. The output of the MRF are reliable detections $\mathbf{b}^t$ which are used to extract fixed-sized representations $\boldsymbol{f}^t$, which are then passed to a matching RNN that reasons in the temporal domain. The RNN outputs the probability of an individual's action, $\boldsymbol{p}_I$, and the collective activity, $\boldsymbol{p}_c$ across time. Note that $\mathcal{L}_{det}$ (3) is the loss function for the detections, and $\mathcal{L}_{CI}$ (14) is the loss function for the individual and collective actions.

passed as input to the second LSTM capturing scene-level representation. [28] explores a slightly different perspective: authors notice that in some settings, the activity is defined by the actions of a single individual and propose a soft attention mechanism to identify her. The complete model is very close to that of [20], except that the attention pooling is used instead of a max pool. All of those methods are effective, however, they start joint reasoning in late inference stages, thus possibly discarding useful context information. Moreover, they all rely on ground truth detections and/or tracks, and thus do not really solve the problem end-to-end.

Our model builds upon the existing work in that it also relies on the discriminative power of deep learning, and employs a version of person-level temporal model. It is also able to implicitly capture the context and perform social scene understanding, which includes reliable localization and action recognition, all in a single end-to-end framework.

## 3. Method

Our main goal is to construct comprehensive interpretations of social scenes from raw image sequences. To this end, we propose a unified way to jointly detect multiple interacting individuals and recognize their collective and individual actions.

### 3.1. Overview

The general overview of our model is given in Figure 2. For every frame $\mathbf{I}^t \in \mathbb{R}^{H_0\times W_0\times 3}$ in a given sequence, we first obtain a dense feature representation $\mathbf{F}^t \in \mathbb{R}^{|\mathcal{I}|\times D}$, where $\mathcal{I} = \{1, \dots, H \times W\}$ denotes the set of all pixel locations in the feature map, $|\mathcal{I}| = H \times W$ is the number of pixels in that map, and $D$ is the number of features. The feature map $\mathbf{F}^t$ is then shared between the detection and action recognition tasks. To detect, we first obtain a preliminary set of detection hypotheses, encoded as two dense maps $\mathbf{B}^t \in \mathbb{R}^{|\mathcal{I}|\times 4}$ and $\mathbf{P}^t \in \mathbb{R}^{|\mathcal{I}|}$, where at each location $i \in \mathcal{I}$, $\mathbf{B}_i^t$ encodes the coordinates of the bounding box, and $\mathbf{P}_i^t$ is the probability that this bounding box represents a person. Those detections are refined jointly by inference in a hybrid Markov Random Field (MRF). The result of the inference is a smaller set of $N$ reliable detections, encoded as bounding boxes $\mathbf{b}^t \in \mathbb{R}^{N\times 4}$. These bounding boxes are then used to smoothly extract fixed-size representations $\boldsymbol{f}_n^t \in \mathbb{R}^{K\times K\times D}$ from the feature map $\mathbf{F}^t$, where $K$ is the size of the fixed representation in pixels. Representations $\boldsymbol{f}_n^t$ are then used as inputs to the matching RNN, which merges the information in the temporal domain. At each time step $t$, RNN produces probabilities $\boldsymbol{p}_{I,k}^t \in \mathbb{R}^{N_I}$ of individual actions for each detection $\mathbf{b}_n^t$, along with the probabilities of collective activity $\boldsymbol{p}_C^t \in \mathbb{R}^{N_C}$, where $N_I, N_c$ denote respectively the number of classes of individual and collective actions. In the following sections, we will describe each of these components in more detail.

### 3.2. Joint Feature Representation

We build upon the Inception architecture [36] for getting our dense feature representation, since it does not only demonstrate good performance but is also more computationally efficient than some of the more popular competitors [32, 25].

One of the challenges when simultaneously dealing with multiple tasks is that representations useful for one task may be quite inefficient for another. In our case, person detection requires reasoning on the type of the object, whereas discriminating between actions can require looking at lower-level details. To tackle this problem, we pro-

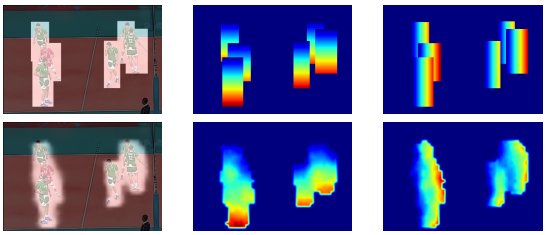


Figure 3. Example of ground truth (top) and predicted (bottom) maps. We show segmentation map $\mathbf{P}$ projected on the original image, followed by two out of four channels of the regression map $\mathbf{B}$, which encode respectively vertical and horizontal displacement from the location $i$ to one of the bounding box corners.

pose using multi-scale features: instead of simply using the final convolutional layer, we produce our dense feature map $\mathbf{F} \in \mathbb{R}^{|\mathcal{I}| \times D}$ (here and later $t$ is omitted for clarity) by concatenating multiple intermediate activation maps. Since they do not have fitting dimensions, we resize them to the fixed size $|\mathcal{I}| = H \times W$ via differentiable bilinear interpolation. Note that similar approaches have been very successful for semantic segmentation [27, 18], when one has to simultaneously reason about the object class and its boundaries.

### 3.3. Dense Detections

Given the output of the feature extraction stage, the goal of the detection stage is to generate a set of reliable detections, that is, a set of bounding box coordinates with their corresponding confidence scores. We do it in a dense manner, meaning that, given the feature map $\mathbf{F} \in \mathbb{R}^{|\mathcal{I}| \times D}$, we produce two dense maps $\mathbf{B} \in \mathbb{R}^{|\mathcal{I}| \times 4}$ and $\mathbf{P} \in \mathbb{R}^{|\mathcal{I}|}$, for bounding boxes coordinates and presence probability, respectively. Essentially, $\mathbf{P}$ represents a segmentation mask encoding which parts of the image contain people, and $\mathbf{B}$ represents the coordinates of the bounding boxes of the people present in the scene, encoded relative to the pixel locations. This is illustrated by Figure 3.

We can interpret this process of generating $\mathbf{P}, \mathbf{B}$ from $\mathbf{F}$ in several different ways. With respect to recent work on object detection [17, 29, 30], it can be seen as a fully-convolutional network that produces a dense set of object proposals, where each pixel of the feature map $\mathbf{F}$ generates a proposal. Alternatively, we can see this process as an advanced non-linear version of the Hough transform, similar to Hough Forests [16, 5]. In these methods, each patch of the image is passed through a set of decision trees, which produce a distribution over potential object locations. The crucial differences with the older methods are, first, leveraging deep neural network as a more powerful regressor and, second, the ability to use large contexts in the image, in particular to reason jointly about parts.

Let us now introduce $\mathbf{B}$ and $\mathbf{P}$ more formally, by defining how we convert the given ground truth object locations into dense ground truth maps $\hat{\mathbf{B}}, \hat{\mathbf{P}}$. For each image $\mathbf{I}$, the detection ground truth is given as a set of bounding boxes $\{(y_0, x_0, y_1, x_1)_1, \dots, \}$. To obtain the value for the specific location $i = (i_y, i_x) \in \mathcal{I}$ of the ground truth probability map $\hat{\mathbf{P}}$, we set $\hat{\mathbf{P}}_i = 1$ if $y_0 \leq i_y \leq y_1, x_0 \leq i_x \leq x_1$ for any of the ground truth boxes, and $\hat{\mathbf{P}}_i = 0$ otherwise. For the regression map, each location $i$ represents a vector $\hat{\mathbf{B}}_i = (t_{y0}, t_{x0}, t_{y1}, t_{x1})$, where:

$$t_{y0} = (i_y - y_0)/s_y, \; t_{x0} = (i_x - x_0)/s_x \; , \tag{1}$$

$$t_{y1} = (y_1 - i_y)/s_x, \; t_{x1} = (x_1 - i_x)/s_y \; , \tag{2}$$

where $s_y, s_x$ are scaling coefficients that are fixed, and can be taken either as the maximum size of the bounding box over the training set, or the size of the image. Ultimately, our formulation makes it possible to use ground truth instance-level segmentation masks to assign each $i$ to one of the ground truth instances. However, since these masks are not available, and there can be multiple ground truth bounding boxes that contain $i$, we assign each $i$ to the bounding box with the highest $y_0$ coordinate, as shown in Figure 3. Note that, $\hat{\mathbf{B}}_i$ are only defined only for $i : \hat{\mathbf{P}}_i = 1$, and the regression loss is constructed accordingly.

The mapping from $\mathbf{F}$ to $\mathbf{B}$, $\mathbf{P}$ is a fully-convolutional network, consisting of a stack of two $3 \times 3$ convolutional layers with 512 filters and a shortcut connection [19]. We use softmax activation function for $\mathbf{P}$ and ReLU for $\mathbf{B}$. The loss is defined as follows:

$$\begin{aligned} \mathcal{L}_{det} = &- \frac{1}{|\mathcal{I}|} \sum_i \hat{\mathbf{P}}_i \log \mathbf{P}_i + \\ & w_{reg} \frac{1}{\sum_i \hat{\mathbf{P}}_i} \cdot \sum_i \hat{\mathbf{P}}_i ||\hat{\mathbf{B}}_i - \mathbf{B}_i||_2^2 \; , \end{aligned} \tag{3}$$

where $w_{reg}$ is a weight that makes training focused more on classification or regression. For datasets where classification is easy, such as `volleyball` [20], we set it to $w_{reg} = 10$, whereas for cluttered scenes with large variations in appearance lower values could be beneficial.

### 3.4. Inference for Dense Detection Refinement

The typical approach to get the final detections given a set of proposals is to re-score them using an additional recognition network and then run non-maxima suppression (NMS) [23, 30]. This has several drawbacks. First, if the amount of the proposals is large, the re-scoring stage can be prohibitively expensive. Second, the NMS step itself is by no means optimal, and is susceptible to greedy decisions. Instead of this commonly used technique, we propose using a simple inference procedure that does not require re-scoring, and makes NMS in the traditional sense unnecessary. Our key observation is that instead of making similar hypotheses suppressing each other, one can rather make

them refine each other, thus increasing the robustness of the final estimates.

To this end, we define a hybrid MRF on top of the dense proposal maps $\mathbf{B}^*$, which we obtain by converting $\mathbf{B}$ to the global image coordinates. For each hypothesis location $i \in \mathcal{I}$ we introduce two hidden variables, one multinomial Gaussian $\mathbf{X}_i \in \mathbb{R}^4$, and one categorical $A_i \in \mathcal{I}$. $\mathbf{X}_i$ encodes the "true" coordinates of the detection, and $A_i$ encodes the assignment of the detection to one of the hypothesis locations in $\mathcal{I}$. Note that, although this assignment variable is discrete, we formulate our problem in a probabilistic way, through distributions, thus allowing a detection to be "explained" by multiple locations. The joint distribution over $\mathbf{X}_{1:|\mathcal{I}|}, A_{1:|\mathcal{I}|}$ is defined as follows:

$$P(\mathbf{X}_{1:|\mathcal{I}|}, A_{1:|\mathcal{I}|}) \propto \prod_{i,j} \exp\left(-\frac{\mathbb{1}[A_i = j] \cdot ||\mathbf{X}_i - \mathbf{X}_j||_2^2}{2\sigma^2}\right) , \tag{4}$$

where $\sigma$ is the standard deviation parameter, which is fixed.

Intuitively, (4) jointly models the relationship between the bounding box predictions produced by the fully-convolutional network. The basic assumption is that each location $i \in \mathcal{I}$ on the feature map belongs to a single "true" detection location $j$, which can be equal to $i$, and the observation $\mathbf{X}_i$ should not be far from the observation $\mathbf{X}_j$ at this "true" location. The goal of inference is to extract those "true" locations and their corresponding predictions by finding the optimal assignments for $A_i$ and values of $\mathbf{X}_i$. In other words, we want to compute marginal distributions $P(\mathbf{X}_i), P(A_i), \forall i \in \mathcal{I}$. Unfortunately, the exact integration is not feasible, and we have to resort to an approximation. We use the mean-field approximation, that is, we introduce the following factorized variational distribution:

$$Q(\mathbf{X}_{1:|\mathcal{I}|}, A_{1:|\mathcal{I}|}) = \prod_i \mathcal{N}(\mathbf{X}_i\,; \boldsymbol{\mu}_i, \sigma^2) \cdot \mathrm{Cat}(A_i\,; \boldsymbol{\eta}_i) , \tag{5}$$

where $\boldsymbol{\mu}_i \in \mathbb{R}^4$ and $\boldsymbol{\eta}_i \in \mathbb{R}^{|\mathcal{I}|}$ are the variational parameters of the Gaussian and categorical distributions respectively. Then, we minimize the KL-divergence between the variational distribution (5) and the joint (4), which leads to the following fixed-point updates for the parameters of $Q(\cdot)$:

$$\eta_{ij}^\tau \propto -\frac{||\boldsymbol{\mu}_i^{\tau-1} - \boldsymbol{\mu}_j^{\tau-1}||_2^2}{2\sigma^2} , \boldsymbol{\alpha}_i^\tau = \mathrm{softmax}(\boldsymbol{\eta}_i^\tau) , \tag{6}$$

$$\hat{\boldsymbol{\mu}}_i^\tau = \sum_j \alpha_{ij} \boldsymbol{\mu}_j^{\tau-1} , \tag{7}$$

where $\tau \in \{1, \ldots, \mathcal{T}\}$ is the iteration number, $\boldsymbol{\alpha}_i^\tau \in \mathbb{R}^{|\mathcal{I}|}, \sum_j \alpha_{ij}^\tau = 1$ is the reparameterization of $\boldsymbol{\eta}_i^\tau$. The complete derivation of those updates is provided in the supplementary material.

Starting from some initial $\boldsymbol{\mu}^0$, one can now use (6), (7) until convergence. In practice, we start with $\boldsymbol{\mu}^0$ initialized from the estimates $\mathbf{B}^*$, thus conditioning our model on the observations, and only consider those $i \in \mathcal{I}$, for which the segmentation probability $\mathbf{P}_i > \rho$, where $\rho$ is a fixed threshold. Furthermore, to get $\boldsymbol{\mu}^\tau$ we use the following smoothed update for a fixed number of iterations $\mathcal{T}$:

$$\boldsymbol{\mu}_i^\tau = (1 - \lambda) \cdot \boldsymbol{\mu}^{\tau-1} + \lambda \cdot \hat{\boldsymbol{\mu}}^\tau , \tag{8}$$

where $\lambda$ is a damping parameter that can be interpreted as a step-size [4].

To get the final set of detections, we still need to identify the most likely hypothesis out of our final refined set $\boldsymbol{\mu}^\mathcal{T}$. Luckily, since we also have the estimates $\boldsymbol{\alpha}_i^\mathcal{T}$ for the assignment variables $A_i$, we can identify them using a simple iterative scheme similar to that used in Hough Forests [5]. That is, we identify the hypothesis with the largest number of locations assigned to it, then remove those locations from consideration, and iterate until there are no unassigned locations left. The number of assigned locations is then used as a detection score with a very nice interpretation: a number of pixels that "voted" for this detection.

### 3.5. Matching RNN for Temporal Modeling

Previous sections described a way to obtain a set of reliable detections from raw images. However, temporal information is known to be a very important feature when it comes to action recognition [26, 38]. To this end, we propose using a matching Recurrent Neural Network, that allows us to merge and propagate information in the temporal domain.

For each frame $t$, given a set of $N$ detections $\mathbf{b}_n^t, n \in \{1, \ldots, N\}$, we first smoothly extract fixed-sized representations $\boldsymbol{f}_n^t \in \mathbb{R}^{K \times K \times D}$ from the the dense feature map $\mathbf{F}^t$, using bilinear interpolation. This is in line with the ROI-pooling [30], widely used in object detection, and can be considered as a less generic version of spatial transformer networks [21], which were also successfully used for image captioning [23]. Those representations $\boldsymbol{f}_n^t$ are then passed through a fully-connected layer, which produces more compact embeddings $\boldsymbol{e}_n^t \in \mathbb{R}^{D_e}$, where $D_e$ is the number of features in the embedded representation. These embeddings are then used as inputs to the RNN units.

We use standard Gated Recurrent Units (GRU) [8] for each person in the sequence, with a minor modification. Namely, we do not have access to the track assignments neither during training nor testing, which means that the hidden states $\boldsymbol{h}_n^t \in \mathbb{R}^{D_h}$ and $\boldsymbol{h}_n^{t+1} \in \mathbb{R}^{D_h}$, where $D_h$ is the number of features in the hidden state, are not necessarily corresponding to the same person. Our solution to this is very simple: we compute the Euclidean distances between each pair of representations at step $t$ and $t-1$, and then update the hidden state based on those distances. A naive

version that works well when the ground truth locations are given, is to use bounding box coordinates $\mathbf{b}^t, \mathbf{b}^{t-1}$ as the matching representations, and then update $\boldsymbol{h}_n^t$ by the closest match $\boldsymbol{h}_{n^*}^{t-1}$:

$$n^* = \arg\min_m ||\mathbf{b}_n^t - \mathbf{b}_m^{t-1}||_2^2 \,, \tag{9}$$

$$\boldsymbol{h}_n^t = \mathrm{GRU}(\boldsymbol{e}_n^t, \boldsymbol{h}_{n^*}^{t-1}) \,. \tag{10}$$

Alternatively, instead of bounding box coordinates $\mathbf{b}^t$, one can use the embeddings $\boldsymbol{e}^t$. This allows the model to learn a suitable representation, which can be potentially more robust to missing/misaligned detections. Finally, instead of finding a *single* nearest-neighbor to make the hidden state update, we can use *all* the previous representations, weighted by the distance in the embedding space as follows:

$$w_{nm}^t \propto \exp(-||\boldsymbol{e}_n^t - \boldsymbol{e}_m^{t-1}||_2^2) \,, \sum_m w_{nm}^t = 1, \tag{11}$$

$$\hat{\boldsymbol{h}}^{t-1} = \sum_m w_{nm}^t \boldsymbol{h}_m^{t-1} \,, \tag{12}$$

$$\boldsymbol{h}_n^t = \mathrm{GRU}(\boldsymbol{e}_n^t, \hat{\boldsymbol{h}}^{t-1}) \,. \tag{13}$$

We experimentally evaluated all of these matching techniques, which we call respectively `boxes`, `embed` and `embed-soft`. We provide results in Section 4.

To get the final predictions $\boldsymbol{p}_C^t$ for collective activities, we max pool over the hidden representations $\boldsymbol{h}^t$ followed by a softmax classifier. The individual actions predictions $\boldsymbol{p}_{I,n}^t$ are computed by a separate softmax classifier on top of $\boldsymbol{h}_n^t$ for each detection $n$. The loss is defined as follows:

$$\begin{aligned} \mathcal{L}_{CI} = &- \frac{1}{T \cdot N_C} \sum_{t,c} \hat{\boldsymbol{p}}_{C,c}^t \log \boldsymbol{p}_{C,c}^t \\ &- w_I \frac{1}{T \cdot N \cdot N_I} \sum_{t,n,a} \hat{\boldsymbol{p}}_{I,n,a}^t \log \boldsymbol{p}_{I,n,a}^t \,, \end{aligned} \tag{14}$$

where $T$ is the number of frames, $N_C, N_I$ are the numbers of labels for collective and individual actions, $N$ is the number of detections, and $\hat{\boldsymbol{p}}_*$ is the one-hot-encoded ground truth. The weight $w_I$ allows us to balance the two tasks differently, but we found that the model is somewhat robust to the choice of this parameter. In our experiments, we set $w_I = 2$.

# 4. Evaluation

In this section, we report our results on the task of multi-person scene understanding and compare them to the baselines introduced in Section 2. We also compare our detection pipeline to multiple state-of-the-art detection algorithms on a challenging dataset for multi-person detection.

## 4.1. Datasets

We evaluate our framework on the recently introduced `volleyball` dataset [20], since it is the only publicly available dataset for multi-person activity recognition that is relatively large-scale and contains labels for people locations, as well as their collective and individual actions.

This dataset consists of 55 volleyball games with 4830 labelled frames, where each player is annotated with the bounding box and one of the 9 individual actions, and the whole scene is assigned with one of the 8 collective activity labels, which define which part of the game is happening. For each annotated frame, there are multiple surrounding unannotated frames available. To get the ground truth locations of people for those, we resort to the same appearance-based tracker as proposed by the authors of the dataset [20].

## 4.2. Baselines

We use the following baselines and versions of our approach in the evaluation:

- `Inception-scene` - Inception-v3 network [36], pre-trained on ImageNet and fine-tuned to predict collective actions on whole images, without taking into account locations of individuals.
- `Inception-person` - similar to previous baseline, but trained to predict individual actions based on high-resolution fixed-sized images of individual people, obtained from the ground truth detections.
- `HDTM` - A 2-stage deep temporal model model [20], consisiting of one LSTM to aggregate person-level dynamics, and one LSTM to aggregate scene-level temporal information. We report multiple versions of this baseline: the complete version which includes both scene-level and person-level temporal models, `scene`, which only uses scene-level LSTM, and `person`, which only uses person-level LSTM.
- `OURS-single` - A version of our model that does not use an RNN. We report results for ground truth locations, as well as detections produced by our detection pipeline.
- `OURS-temporal` - A complete version of our model with GRU units for temporal modeling. We report results both for ground truth locations and our detections, as well as results for different matching functions.

## 4.3. Implementation Details

All our models are trained using backpropagation using the same optimization scheme: for all the experiments and all datasets, we use stochastic gradient descent with ADAM [24], with the initial learning rate set to $10^{-5}$, and fixed hypereparameters to $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$.

We train our model in two stages: first, we train a network on single frames, to jointly predict detections, individual, and collective actions. We then fix the weights of the feature extraction part of our model, and train our temporal RNN to jointly predict individual actions together with collective activities. Note that in fact our model is fully-differentiable, and the reason for this two-stage training is purely technical: backpropagation requires keeping all the activations in memory, which is not possible for a batch of image sequences. The total loss is simply a sum of the detection loss (3) and the action loss (14) for the first stage, and the action loss for the second stage. We use a temporal window of length $T = 10$, which corresponds to 4 frames before the annotated frame, and 5 frames after.

The parameters of the MRF are the same for all the experiments. We run inference on the bounding boxes with the probability $\mathbf{P}_i$ above the threshold $\rho = 0.2$, and set the standard deviation $\sigma = 0.005$, step size $\lambda = 0.2$, and the number of iterations $\mathcal{T} = 20$.

Our implementation is based on TensorFlow [1] and its running time for a single sequence of $T = 10$ high-resolution (720x1080) images is approximately 1.2s on a single Tesla-P100 NVIDIA GPU.

### 4.4. Multi-Person Scene Understanding

The quantitative results on the `volleyball` dataset are given in Table 1. Whenever available, we report accuracies both for collective action recognition and individual action recognition. For variants of our methods, we report two numbers: when the output of our detection pipeline was used (MRF), and the ground truth bounding boxes (GT). Our method is able to achieve state-of-the-art performance for collective activity recognition even without ground truth locations of the individuals and temporal reasoning. With our matching RNN, performance improvements are even more noticeable. The comparison to `Inception-person`, which was fine-tuned specifically for the single task of individual action recognition, indicates that having a joint representation which is shared across multiple tasks leads to an improvement in average accuracy on individual actions. When we use the output of our detections, the drop in performance is expected, especially since we did not use any data augmentation to make the action recognition robust to imperfect localization. For collective actions, having perfect localization is somewhat less important, since the prediction is based on multiple individuals. In Figure 4 we provide some visual results, bounding boxes and actions labels are produced by `OURS-temporal` model with `embed-soft` matching from raw image sequences.

In Table 2 we compare different matching strategies. For the ground truth detections, as expected, simply finding the best match in the bounding box coordinates, `boxes`, works very well. Interestingly, using the `embed` and `embed-soft` matching are beneficial for the performance when detections are used instead of the ground truth. It is also understandable: appearance is more robust than coordinates, but it also means that our model is actually able to capture that robust appearance representation, which might not be absolutely necessary for the prediction in a single frame scenario. Note that, whereas for the collective actions the temporal data seems to help significantly, the improvement for the individual action estimation is very modest, especially for the detections. We hypothesize that in order to discriminate better between individual actions, it is necessary to look at how the low-level details change, which could be potentially smoothed out during the spatial pooling, and thus they are hard to capture for our RNN.

| Method | collective | individual |
|---|---|---|
| `Inception-scene` (GT) | 75.5 | - |
| `Inception-person` (GT) | - | 78.1 |
| `HDTM-scene` [20](GT) | 74.7 | - |
| `HDTM-person` [20](GT) | 80.2 | - |
| `HDTM` [20](GT) | 81.9 | - |
| `OURS-single` (MRF/GT) | 83.3 / 83.8 | 77.8 / 81.1 |
| `OURS-temporal` (MRF/GT) | 87.1 / **89.9** | 77.9 / **82.4** |

Table 1. Results on the `volleyball` dataset. We report average accuracy for collective activity and individual actions. For `OURS-temporal` for the ground truth bounding boxes (GT) we report results with the `bbox` matching, and for the detections (MRF) we report results with the `embed` matching.

| Method | collective | individual |
|---|---|---|
| `boxes` (MRF/GT) | 82.0 / 89.9 | 68.6 / **82.4** |
| `embed` (MRF/GT) | 87.1 / 90.0 | 77.9 / 81.9 |
| `embed-soft` (MRF/GT) | 86.2 / **90.6** | 77.4 / 81.8 |

Table 2. Comparison of different matching strategies for the `volleyball` dataset. `boxes` corresponds to the nearest neighbour (NN) match in the space of bounding box coordinates, `embed` corresponds to the NN in the embedding space $\boldsymbol{e}$, and `embed-soft` is a soft matching in $\boldsymbol{e}$.

| Method | collective | individual |
|---|---|---|
| `boxes` MRF | 82.0 | 68.6 |
| `boxes` NMS | 77.0 | 68.1 |
| `embed` MRF | **87.1** | **77.9** |
| `embed` NMS | 85.2 | 76.2 |
| `embed-soft` MRF | 86.2 | 77.4 |
| `embed-soft` NMS | 85.1 | 75.7 |

Table 3. Comparative results of detection schemes on the `volleyball` dataset. We report the average accuracy for the collective and individual action recognition.

We also conducted experiments to see if our joint detection using MRF is beneficial, and compare it to the tradi-

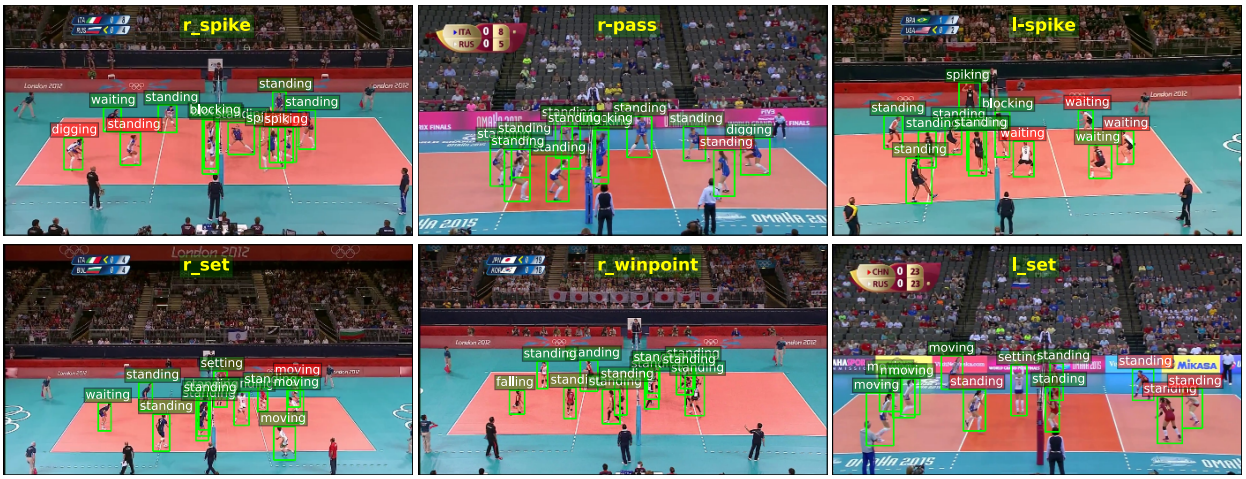



Figure 4. Examples of visual results (better viewed in color). Green boxes around the labels correspond to correct predictions, red correspond to mistakes. The bounding boxes in the images are produced by our detection scheme, and obtained in a single pass together with the action labels.

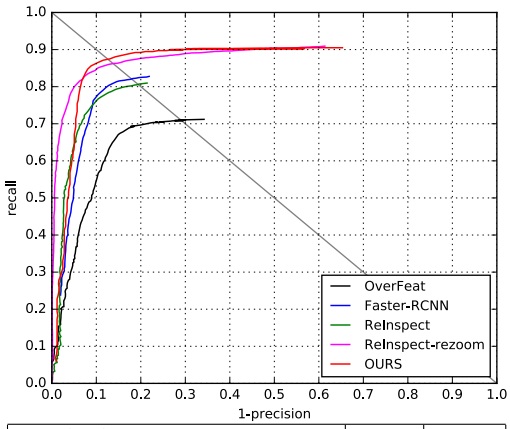
tional non-maxima suppression, both operating on the same dense detection maps. The results for various matching strategies are given in Table 3. For all of them, our joint probabilistic inference leads to better accuracy than non-maxima suppression.

### 4.5. Multi-Person Detection

For completeness, we also conducted experiments for multi-person detection using our dense proposal network followed by a hybrid MRF. Our main competitor is the `ReInspect` algorithm [35], which was specifically designed for joint multi-person detection. We trained and tested our model on the `brainwash` dataset [35], which contains more than 11000 training and 500 testing images, where people are labeled by bounding boxes around their heads. The dataset includes some highly crowded scenes in which there are a large number of occlusions.

Many of the bounding boxes are extremely small and thus have very little image evidence, however, our approach allows us to simultaneously look at different feature scales to tackle this issue. We use 5 convolutional maps of the original Inception-v3 architecture to construct our dense representation $\mathbf{F}$. We do not tune any parameters on the validation set, keeping them the same as for `volleyball` dataset.

In Figure 5 we report average precision (AP) and equal error rate (EER) [13], along with the precision-recall curves. We outperform most of the existing detection algorithms, including widely adopted `Faster-RCNN` [30], by a large margin, and perform very similarly to `ReInspect-rezoom`. One of the benefits of our detection method with respect to the `ReInspect`, is that our approach is not restricted only to detection, and can be also used for instance-level segmentation.




| Method | AP | EER |
|---|---|---|
| `Overfeat` [31] | 0.67 | 0.71 |
| `Faster-RCNN` [30] | 0.79 | 0.80 |
| `ReInspect` [35] | 0.78 | 0.81 |
| `ReInspect-rezoom` [35] | **0.89** | 0.85 |
| `OURS` | 0.88 | **0.87** |

Figure 5. Results for multi-person detection on the `brainwash` [35] dataset (better viewed in color). Our model outperforms most of the widely used baselines, and performs on par with the state-of-the-art `ReInspect-rezoom` [35].

## 5. Conclusions

We have proposed a unified model for joint detection and activity recognition of multiple people. Our approach does not require any external ground truth detections nor tracks, and demonstrates state-of-the-art performance both on multi-person scene understanding and detection datasets. Future work will apply the proposed framework to explicitly capture and understand human interactions.

## References


[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous systems, 2015. *Software available from tensorflow. org*, 1, 2015.

[2] M. R. Amer, P. Lei, and S. Todorovic. Hirf: Hierarchical random field for collective activity recognition in videos. In *European Conference on Computer Vision*, pages 572–585. Springer, 2014.

[3] T. Bagautdinov, F. Fleuret, and P. Fua. Probability occupancy maps for occluded depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2829–2837, 2015.

[4] P. Baque, T. Bagautdinov, F. Fleuret, and P. Fua. Principled parallel mean-field inference for discrete random fields. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[5] O. Barinova, V. Lempitsky, and P. Kholi. On detection of multiple object instances using hough transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1773–1784, 2012.

[6] W. Choi and S. Savarese. Understanding collective activitiesof people from videos. *IEEE transactions on pattern analysis and machine intelligence*, 36(6):1242–1257, 2014.

[7] W. Choi, K. Shahid, and S. Savarese. Learning context for collective activity recognition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3273–3280. IEEE, 2011.

[8] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005.

[10] Z. Deng, A. Vahdat, H. Hu, and G. Mori. Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[11] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634, 2015.

[12] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1110–1118, 2015.

[13] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015.

[14] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[15] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):267–282, 2008.

[16] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 33(11):2188–2202, 2011.

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

[18] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 447–456, 2015.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[20] M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori. A hierarchical deep temporal model for group activity recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[21] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[22] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.

[23] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

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

[25] 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.

[26] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[27] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[28] V. Ramanathan, J. Huang, S. Abu-El-Haija, A. Gorban, K. Murphy, and L. Fei-Fei. Detecting events and key actors in multi-person videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[29] 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.

[30] 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.

[31] 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.

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[33] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[34] S. Singh, C. Arora, and C. V. Jawahar. First person action recognition using deep learned descriptors. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[35] R. Stewart, M. Andriluka, and A. Y. Ng. End-to-end people detection in crowded scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[36] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *arXiv preprint arXiv:1512.00567*, 2015.

[37] V. Veeriah, N. Zhuang, and G.-J. Qi. Differential recurrent neural networks for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4041–4049, 2015.

[38] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision*, 103(1):60–79, 2013.

[39] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4305–4314, 2015.

[40] D. Weinland, M. Ozuysal, and P. Fua. Making Action Recognition Robust to Occlusions and Viewpoint Changes. 2010.

[41] S. Zhang, R. Benenson, and B. Schiele. Filtered feature channels for pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1751–1760, 2015.