End-to-End Learning of Visual Representations from Uncurated Instructional Videos

Antoine Miech, Jean-Baptiste Alayrac, Piotr Bojanowski, Ivan Laptev, Josef Sivic

Learning from Video and Text via Large-Scale Discriminative Clustering

Antoine Miech[1,2] Jean-Baptiste Alayrac[1,2] Piotr Bojanowski[2] Ivan Laptev [1,2] Josef Sivic[1,2,3]
[1]École Normale Supérieure [2]Inria [3]CIIRC

Abstract

Discriminative clustering has been successfully applied to a number of weakly-supervised learning tasks. Such applications include person and action recognition, text-to-video alignment, object co-segmentation and co-localization in videos and images. One drawback of discriminative clustering, however, is its limited scalability. We address this issue and propose an online optimization algorithm based on the Block-Coordinate Frank-Wolfe algorithm. We apply the proposed method to the problem of weakly-supervised learning of actions and actors from movies together with corresponding movie scripts. The scaling up of the learning problem to 66 feature-length movies enables us to significantly improve weakly-supervised action recognition.

1. Introduction

Action recognition has been significantly improved in recent years. Most existing methods [23, 34, 38, 39] rely on supervised learning and, therefore, require large-scale, diverse and representative action datasets for training. Collecting such datasets, however, is a difficult task given the high costs of manual search and annotation of the video. Notably, the largest action datasets today are still orders of magnitude smaller (UCF101 [36], ActivityNet [7]) compared to large image datasets, they often contain label noise and target specific domains such as sports (Sports1M [20]).

Weakly-supervised learning aims to bypass the need of manually-annotated datasets using readily-available, but possibly noisy and incomplete supervision. Examples of such methods include learning of person names from image captions or video scripts [3, 10, 35, 37]. Learning actions from movies and movie scripts has been approached in [4, 5, 9, 23]. Most of the work on weakly-supervised person and action learning, however, has been limited to one or a few movies. Therefore the power of leveraging large-scale weakly-annotated video data has not been fully explored.

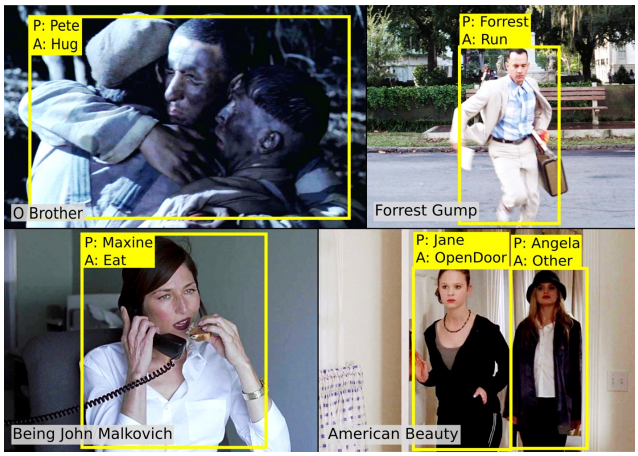

Figure 1: We automatically recognize actors and their actions in a of dataset of 66 movies with scripts as weak supervision.

In this work we aim to scale weakly-supervised learning of actions. In particular, we follow the work of [4] and learn actor names together with their actions from movies and movie scripts. While actors are learned separately for each movie, differently from [4], our method simultaneously learns actions from all movies and movie scripts available for training. Such an approach, however, requires solving a large-scale optimization problem. We address this issue and propose to scale weakly-supervised learning by adapting the Block-Coordinate Frank-Wolfe approach [21]. Our optimization procedure enables action learning from tens of movies and thousands of action samples, readily available from our subset of movies or other recent datasets with movie descriptions [30]. This, in turn, results in large improvements in action recognition.

Besides the optimization, our work introduces a new model for background class in the form of a constraint. It enables better and automatic modeling of the background class (*i.e.* unknown actors and actions). We evaluate our method on 66 movies and demonstrate significant improvements for both actor and action recognition. Example results are illustrated in Figure 1.

[1]Département d'informatique de l'ENS, École normale supérieure, CNRS, PSL Research University, 75005 Paris, France.
[3]Czech Institute of Informatics, Robotics and Cybernetics at the Czech Technical University in Prague.

1.1. Related Work

This section reviews related work on discriminative clustering, Frank-Wolfe optimization and its applications to the weakly-supervised learning of people and actions in video.

Discriminative clustering and Frank-Wolfe. The Frank-Wolfe algorithm [11, 15] allows to minimize large convex problems over convex sets by solving a sequence of linear problems. In computer vision, it has been used in combination with discriminative clustering [2] for action localization [5], text-to-video alignment [1, 6], object co-localization in videos and images [18] or instance-level segmentation [31]. A variant of Frank-Wolfe with randomized block coordinate descent was proposed in [21]. This extension leads to lower complexity in terms of time and memory requirements while preserving the convergence rate. In this work we build on [21] and adapt it for the problem of large-scale weakly-supervised learning of actions from movies.

Weakly-supervised action recognition. Movie scripts are used as a source of weak supervision for temporal action localization in [9]. An extension of this work [5] exploits the temporal order of actions as a learning constraint. Other [22] target spatio-temporal action localization and recognition in video using a latent SVM. A weakly-supervised extension of this method [33] localizes actions without location supervision at the training time. Another recent work [40] proposes a multi-fold Multiple-Instance Learning (MIL) SVM to localize actions given video-level supervision at training time. Closer to us is the work of [4] that improves weakly supervised action recognition by joint action-actor constraints derived from scripts. While the approach in [4] is limited to a few action classes and movies, we propose here a scalable solution and demonstrate significant improvements in action recognition when applied to the large-scale weakly-supervised learning of actions from many movies.

Weakly-supervised person recognition. Person recognition in TV series has been studied in [10, 35] where the authors propose a solution to the problem of associating speaker names in scripts and faces in videos. Speakers in videos are identified by detecting face tracks with lip motion. The method in [8] presents an alternative solution by formulating the association problem using a convex surrogate loss. Parkhi *et al*. [27] present a method for person recognition combining a MIL SVM with a model for the background class. Most similar to our model is the one presented in [4]. The authors propose a discriminative clustering cost under linear constraints derived from scripts to recover the identities and actions of people in movies. Apart from scaling-up the approach of [4] to much larger datasets, our model extends and improves [4] with a new background constraint.

Contributions. In this paper we make the following contributions: (i) We propose an optimization algorithm based on Block-Coordinate Frank-Wolfe that allows scaling up discriminative clustering models [2] to much larger datasets. (ii) We extend the joint weakly-supervised Person-Action model of [4], with a simple yet efficient model of the background class. (iii) We apply the proposed optimization algorithm to scale-up discriminative clustering to an order of magnitude larger dataset, resulting in significantly improved action recognition performance.

2. Discriminative Clustering for Weak Supervision

Assume we want to assign labels (*e.g*. names or action classes) to samples (*e.g*. person tracks in the video). unlike in the standard supervised learning setup, the exact labels of samples are not known at training time. Instead, we are given only partial information that some samples in a subset (or bag) may belong to some of the labels. This ambiguous setup, also known as multiple instance learning, is common, for example, when learning human actions from videos and associated text descriptions.

To address this challenge of ambiguous and partial labels, we build on the discriminative clustering criterion based on a linear classifier and a quadratic loss (DIFFRAC [2]). This framework has shown promising results in weakly-supervised and unsupervised computer vision tasks [1, 4, 5, 6, 16, 17, 29, 31]. In particular, we use this approach to group samples into linearly separable clusters. Suppose we have N samples to group into K classes. We are given d-dimensional features $X \in \mathbb{R}^{N \times d}$, one for each of the N samples, and our goal is to find a binary matrix $Y \in \{0,1\}^{N \times K}$ assigning each of the N samples to one of the labels, where $Y_{nk} = 1$ if and only if the sample n (*e.g*. a person track in a movie) is assigned to the label k (*e.g*. action class running).

First, suppose the assignment matrix Y is given. In this case finding a linear classifier W can be formulated as a ridge regression problem

$$\min_{W \in \mathbb{R}^{d \times K}} \frac{1}{2N} \|Y - XW\|_{\mathrm{F}}^2 + \frac{\lambda}{2} \|W\|_{\mathrm{F}}^2, \qquad (1)$$

where X is a matrix of input features, Y is the given labels assignment matrix, $\|.\|_{\mathrm{F}}$ is the matrix norm (or Frobenius norm) induced by the matrix inner product $\langle ., . \rangle_{\mathrm{F}}$ (or Frobenius inner product) and λ is a regularization hyper-parameter set to a fixed constant. The key observation is that we can resolve the classifier W^* in closed form as

$$W^*(Y) = (X^\top X + N\lambda I)^{-1} X^\top Y. \qquad (2)$$

In our weakly-supervised setting, however, Y is unknown. Therefore, we treat Y as a latent variable and optimize (1) w.r.t. W and Y. In details, plugging the optimal

solution W^* (2) in the cost (1) removes the dependency on W and the cost can be written as a quadratic function of Y, i.e. $C(Y) = \langle Y, A(X,\lambda)Y\rangle_{\mathrm{F}}$, where $A(X,\lambda)$ is a matrix that only depends on the data X and a regularization parameter λ. Finding the best assignment matrix Y can then be written as the minimization of the cost $C(Y)$:

$$\min_{Y\in\{0,1\}^{N\times K}} \langle Y, A(X,\lambda)Y\rangle_{\mathrm{F}}. \tag{3}$$

Solving the above problem, however, can lead to degenerate solutions [2] unless additional constraints on Y are provided. In section 3, we incorporate weak supervision in the form of constraints on the latent assignment matrices Y. The constraints on Y used for weak supervision generally decompose into small independent blocks. This block structure is the key for our optimization approach that we will present next.

2.1. Large-Scale optimization

The Frank-Wolfe (FW) algorithm has been shown effective for optimizing convex relaxation of (3) in a number of vision problems [1, 4, 5, 6, 19, 31]. It only requires solving linear programs on a set of constraints. Therefore, it avoids costly projections and allows the use of complicated constraints such as temporal ordering [5]. However, the standard FW algorithm is not well suited to solve (3) for a large number of samples N.

First, storing the $N \times N$ matrix $A(X,\lambda)$ in memory becomes prohibitive (*e.g.* the size of A becomes $\geq$ 100GB for $N \geq 200000$). Second, each update of the FW algorithm requires a full pass over the data resulting in a space and time complexity of order N for each FW step.

Weakly supervised learning is, however, largely motivated by the desire of using large-scale data with "cheap" and readily-available but incomplete and noisy annotation. Scaling up weakly-supervised learning to a large number of samples is, therefore, essential for its success. We address this issue and develop an efficient version of the FW algorithm. Our solution builds on the Block-Coordinate Frank-Wolfe (BCFW) [21] algorithm and extends it with a smart block-dependent update procedure as described next. The proposed update procedure is one of the key contribution of this paper.

2.1.1 Block-coordinate Frank-Wolfe (BCFW)

The Block-Coordinate version of the Frank-Wolfe algorithm [21] is useful when the convex feasible set $\mathcal{Y}$ can be written as a Cartesian product of n smaller sets of constraints: $\mathcal{Y} = \mathcal{Y}^{(1)} \times \ldots \times \mathcal{Y}^{(n)}$. Inspired by coordinate descent techniques, BCFW consists of updating one variable block $\mathcal{Y}^{(i)}$ at a time with a reduced Frank-Wolfe step. This method has potentially n times cheaper iterates both in space and time. We will see that most of the weakly-supervised problems exhibit such a block structure on latent variables.

2.1.2 BCFW for discriminative clustering

To benefit from BCFW, we have to ensure that the time and space complexity of the block update does not depend on the total number of samples N (*e.g.* person tracks in all movies) but only depends on the size N_i of smaller blocks of samples i (*e.g.* person tracks within one movie). After a block is sampled, the update consists of two steps. First, the gradient with respect to the block is computed. Then the *linear oracle* is called to obtain the next iterate. As we show below, the difficult part in our case is to efficiently compute the gradient with respect to the block.

Block gradient: a naive approach. Let's denote N_i the size of block i. The objective function f of problem (3) is $f(Y) = \langle Y, A(X,\lambda)Y\rangle_{\mathrm{F}}$, where (see [2])

$$A(X,\lambda) = \frac{1}{2N}(I_N - X(X^\top X + N\lambda I_d)^{-1}X^\top). \tag{4}$$

To avoid storing matrix $A(X,\lambda)$ of size $N\times N$, one can precompute the matrix $P = (X^\top X + N\lambda I_d)^{-1}X^\top \in \mathbb{R}^{d\times N}$. We can write the block gradient with respect to a subset of samples i as follows:

$$\nabla_{(i)} f(Y) = \frac{1}{N}(Y^{(i)} - X^{(i)}PY), \tag{5}$$

where $Y^{(i)} \in \mathbb{R}^{N_i\times K}$ and $X^{(i)} \in \mathbb{R}^{N_i\times d}$ are the label assignment variable and the feature matrix for block i (*e.g.* person tracks in movie i), respectively. Because of the PY matrix multiplication, naively computing this formula has the complexity $\mathcal{O}(NdK)$, where N is the total number of samples, d is the dimensionality of the feature space and K is the number of classes. As this depends linearly on N, we aim to find a more efficient way to compute block gradients, as described next.

Block gradient: a smart update. We now propose an update procedure that avoids re-computation of block gradients and whose time and space complexity at each iteration depends on N_i instead of N. The main intuition is that we need to find a way to store information about all the blocks in a compact form. A natural way of doing so is to maintain the weights of the linear regression parameters $W \in \mathbb{R}^{d\times K}$. From (2) we have $W = PY$. If we are able to maintain the variable W at each iteration with the desired complexity $\mathcal{O}(N_i dK)$, then the block gradient computation (5) can be reduced from $\mathcal{O}(NdK)$ to $\mathcal{O}(N_i dK)$. We now explain how to effectively achieve that.

At each iteration t of the algorithm, we only update a block i of Y while keeping all other blocks fixed. We denote the direction of the update by $D_t \in \mathbb{R}^{N\times K}$ and the step size by γ_t. With this notation the update becomes

$$Y_{t+1} = Y_t + \gamma_t D_t. \tag{6}$$

The update rule for the weight variable W_t can now be written as follows:

$$\begin{aligned} W_{t+1} &= P(Y_t + \gamma_t D_t) \\ W_{t+1} &= W_t + \gamma_t P D_t, \end{aligned} \tag{7}$$

Recall that at iteration t, BCFW only updates block i, therefore D_t has non zero value only at block i. In block notation we can therefore write the matrix product PD_t as:

$$\left[P^{(1)}, \cdots, P^{(i)}, \cdots, P^{(n)}\right] \times \begin{bmatrix} 0 \\ D_t^{(i)} \\ 0 \end{bmatrix} = P^{(i)} D_t^{(i)}, \tag{8}$$

where $P^{(i)} \in \mathbb{R}^{d\times N_i}$ and $D_t^{(i)} \in \mathbb{R}^{N_i \times K}$ are the i-th blocks of matrices P and D_t, respectively. The outcome is an update of the following form

$$W_{t+1} = W_t + \gamma_t P^{(i)} D_t^{(i)}, \tag{9}$$

where the computational complexity for updating W has been reduced to $\mathcal{O}(N_i dK)$ compared to $\mathcal{O}(NdK)$ in the standard update.

We have designed a Block-Coordinate Frank-Wolfe update with time and space complexity depending only on the size of the blocks and not the entire dataset. This allows to scale discriminative clustering to problems with a very large number of samples. The pseudo-code for the algorithm is summarized in Algorithm 1. Next, we describe an application of this large-scale discriminative clustering algorithm to weakly-supervised person and action recognition in movies.

Algorithm 1 BCFW for Discriminative Clustering [2]

Initiate Y_0, $P := (X^\top X + N\lambda I_d)^{-1} X^\top$, $W_0 = PY_0$, $g_i = +\infty$, $\forall i$.
for $t = 1 \dots N_{iter}$ **do**
  $i \leftarrow$ sample from distribution proportional to g [26]
  $\nabla_{(i)} f(Y_t) \leftarrow \frac{1}{N}(Y_t^{(i)} - X^{(i)} W_t)$ # Block gradient
  $Y_{min} \leftarrow \arg\min_{x \in \mathcal{Y}^{(i)}} \langle \nabla_{(i)} f(Y_t), x \rangle_{\mathrm{F}}$ # Linear oracle
  $D \leftarrow Y_{min} - Y^{(i)}$
  $g_i \leftarrow -\langle D, \nabla_{(i)} f(Y_t) \rangle_{\mathrm{F}}$ # Block gap
  $\gamma \leftarrow \min(1, \frac{g_i}{\frac{1}{N}\langle D, D - X^{(i)} P^{(i)} D \rangle_{\mathrm{F}}})$ # Line-search
  $W_{t+1} \leftarrow W_t + \gamma P^{(i)} D$ # W update
  $Y_{t+1}^{(i)} \leftarrow Y_t^{(i)} + \gamma D$ # Block update
end for

3. Weakly-supervised Person-Action model

We now describe an application of our large-scale discriminative clustering algorithm with weak-supervision. The goal is to assign to each person track a name and an action. Both names and actions are mined from movie scripts. For a given movie i, we assume to have N_i automatically extracted person tracks as well as the parsing of a movie script into person names and action classes. We also assume that scripts and movies have been roughly aligned in time. In such a setup we can assign labels (*e.g.* a name or an action) from a script section to a subset of tracks $\mathcal{N}$ from the corresponding time interval of a movie (see Figure 2 for example). In the following, we explain how to convert such form of weak supervision into a set of constraints on latent variables corresponding to the names and actions of people. We will also show how these constraints easily decompose into blocks. We denote Z the latent variable assignment matrix for person names and T for actions.

3.1. Weak-supervision as constraints

We use linear constraints to incorporate weak supervision from movie scripts. In detail, we define constraints on subsets of person tracks that we call "bags". In the following we explain the procedure for construction of bags together with the definition of the appropriate constraints.

'At least one' constraint. Suppose a script reveals the presence of a person p in some time interval of the movie. We construct a set $\mathcal{N}$ with all person tracks in this interval. As first proposed by [4], we model that *at least one* track in $\mathcal{N}$ is assigned to person p by the following constraint

$$\sum_{n\in\mathcal{N}} Z_{np} \geq 1. \tag{10}$$

We can apply the same type of constraint when solving for action assignment T.

Person-Action constraint. Scripts can also provide information that a person p is performing an action a in a scene. In such cases we can formulate stricter and more informative constraints as follows. We construct a set $\mathcal{N}$ containing all person tracks appearing in this scene. Following [4], we formulate a joint constraint on presence of a person performing a specific action as

$$\sum_{n\in\mathcal{N}} Z_{np} T_{na} \geq 1. \tag{11}$$

Mutual exclusion constraint. We also model that each person track can only be assigned to exactly one label. This

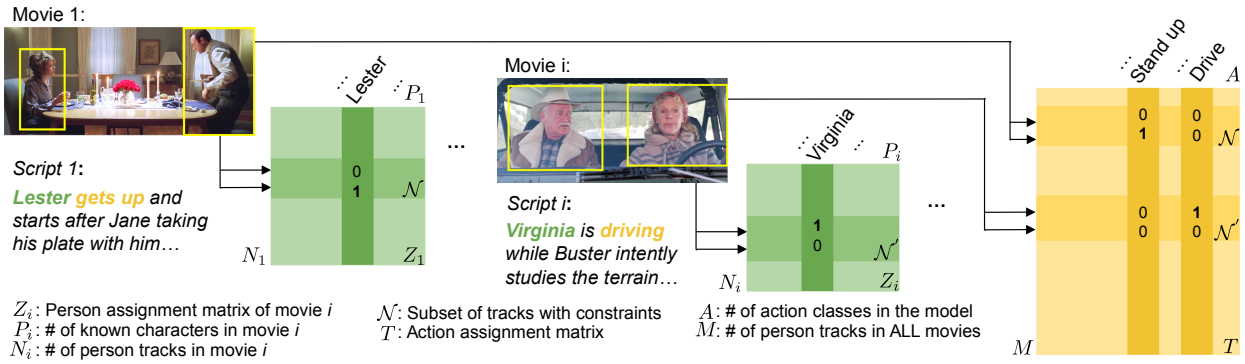

Figure 2: Overview of the Person-Action weakly supervised model, see text for detailed explanations.

restriction can be formalized by the mutual exclusion constraint

$$Z1_P = 1_N, \tag{12}$$

for Z (*i.e.* rows sum up to 1). Same constraint holds for T.

Background class constraint. One of our contributions is a novel way of coping with the background class. As opposed to previous work [4], our approach allows us to have background model that does not require any external data. Also it does not require a specific background class classifier as in [27].

Our background class constraint can be seen as a way to supervise people and actions that are not mentioned in scripts. We observe that tracks that are not subject to constraints from Eq. (10) and tracks that belong to crowded shots are likely to belong to the background class. Let us denote by $\mathcal{B}$ the set of such tracks. We impose that at least a certain fraction $\alpha \in [0, 1]$ of tracks in $\mathcal{B}$ must belong to the background class. Assuming that person label $p = 1$ corresponds to the background, we obtain the following linear constraint (similar constraint can be defined for actions on T):

$$\sum_{n \in \mathcal{B}} Z_{n1} \geq \alpha \mid \mathcal{B} \mid . \tag{13}$$

3.2. Person-Action model formulation

Here we summarize the complete formulation of the person and action recognition problems.

Solving for names. We formulate the person recognition problem as discriminative clustering, where X_1 are face descriptors:

$$\min_{Z \in \{0,1\}^{N \times P}} \langle Z, A(X_1, \lambda) Z \rangle_{\mathrm{F}}, \quad \text{(Discriminative cost)} \tag{14}$$

$$\text{such that} \quad \begin{cases} \sum_{n \in \mathcal{N}} Z_{np} \geq 1, & \text{(At least one)} \\ \sum_{n \in \mathcal{B}} Z_{n1} \geq \alpha \mid \mathcal{B} \mid, & \text{(Background)} \\ Z1_P = 1_N. & \text{(Mutual exclusion)} \end{cases}$$

Solving for actions. After solving the previous problem for names separately for each movie, we vertically concatenate all person name assignment matrices Z. We also define a single action assignment variable T in $\{0, 1\}^{M \times A}$, where M is the total number of tracks across all movies and X_2 are action descriptors (details given later). We formulate our action recognition problem as a large QP:

$$\min_{T \in \{0,1\}^{M \times A}} \langle T, A(X_2, \mu) T \rangle_{\mathrm{F}}, \quad \text{(Discriminative cost)} \tag{15}$$

$$\text{such that} \quad \begin{cases} \sum_{n \in \mathcal{N}} T_{na} \geq 1, & \text{(At least one)} \\ \sum_{n \in \mathcal{N}} Z_{np} T_{na} \geq 1, & \text{(Person-Action)} \\ \sum_{n \in \mathcal{B}} T_{n1} \geq \beta \mid \mathcal{B} \mid, & \text{(Background)} \\ T1_A = 1_M. & \text{(Mutual exclusion)} \end{cases}$$

Block-Separable constraints. The set of linear constraints on the action assignment matrix T is block separable since each movie has it own set of constraints, i.e. there are no constraints spanning multiple movies. Therefore, we can fully demonstrate here the power of our large-scale discriminative clustering optimization (Algorithm 1).

4. Experimental Setup

4.1. Dataset

Our dataset is composed of 66 Hollywood feature-length movies (see the list in Appendix) that we obtained from either BluRay or DVD. For all movies, we downloaded their scripts (on `www.dailyscript.com`) that we temporally aligned with the videos and movie subtitles using the method described in [23]. The total number of frames in all 66 movies is 11,320,252. The number of body tracks detected across all movies (see 4.3 for more details) is $M = 201874$.

4.2. Text pre-processing

To provide weak supervision for our method we process movie scripts to extract occurrences of the 13 most frequent action classes: `Stand Up`, `Eat`, `Sit Down`, `Sit Up`,

`Hand Shake`, `Fight`, `Get Out of Car`, `Kiss`, `Hug`, `Answer Phone`, `Run`, `Open Door` and `Drive`. To do so, we collect a corpus of movie scripts different from the set of our 66 movies and train simple text-based action classifiers using linear SVM and a TF-IDF representation of words composed of uni-grams and bi-grams. After retrieving actions in our target movie scripts, we also need to identify who is performing the action. We used spaCy [14] to parse every sentence classified as describing one of the 13 actions and get every subject for each action verb.

4.3. Person detection and Features

Face tracks. To obtain tracks of faces in the video, we run the multi-view face detector [25] based on the DPM model [12]. We then extract face tracks using the same method as in [10, 35]. For each detected face, we compute facial landmarks [35] followed by the face alignment and resizing of face images to 224x224 pixel. We use pre-trained vgg-face features [28] to extract descriptors for each face. We kept the features of dimension 4096 computed by the network at the last fully-connected layer that we L_2 normalized. For each face track, we choose the top K (in practice, we choose K=5) faces that have the best facial landmark confidence. Then we represent each track by averaging the features of the top K faces.

Body tracks. To get the person body tracks, we run the Faster-RCNN network with VGG-16 architecture fine-tuned on VOC 07 [32]. Then we track bounding boxes using the same tracker as used to obtain face tracks. To get person identity for body tracks, we greedily link each body track to one face track by maximizing a spatio-temporal bounding box overlap measure. However if a body track does not have an associated face track as the actor's face may look away from the camera, we cannot obtain its identity. Such tracks can be originating from any actor in the movie. To capture motion features of each body track, we compute bag-of-visual-words representation of dense trajectory descriptors [38] inside the bounding boxes defined by the body track. We use 4000 cluster centers for each of the HOF, MBHx and MBHy channels. In order to capture appearance of each body track we extract ResNet-50 [13] pre-trained on ImageNet. For each body bounding box, we compute the average RoI-pooled [32] feature map of the last convolutional layer within the bounding box, which yields a feature vector of dimension 2048 for each box. We extract a feature vector every 10th frame, average extracted feature vectors over the duration of the track and L_2 normalize. Finally, we concatenate the dense trajectory descriptor and the appearance descriptor resulting in a 14028-dimensional descriptor for each body track.

Method	Acc.	Multi-Class AP	Background AP
Cour *et al.* [8]	48%	63%	–
Sivic *et al.* [35]	49%	63%	–
Bojanowski *et al.* [4]	57%	75%	51%
Parkhi *et al.* [27]	74%	93%	75%
Our method	**83%**	**94%**	**82%**

Table 1: Comparison on the Casablanca benchmark [4].

Episode	1	2	3	4	5
Sivic *et al.* [35]	90	83	70	86	85
Parkhi *et al.* [27]	**99**	90	94	96	**97**
Ours	98	**98**	**98**	**97**	**97**

Table 2: Comparison on the Buffy benchmark [35] using AP.

α	0	0.1	0.2	0.3	0.4	0.5	0.75	1.0
Accuracy	58	58	70	82	**84**	83	76	55
AP	86	87	90	**94**	**94**	93	85	58

Table 3: Sensitivity to hyper-parameter α (13) on Casablanca.

5. Evaluation

5.1. Evaluation of person recognition

We compare our person recognition method to several other methods on the Casablanca benchmark from [4] and the Buffy benchmark from [35]. All methods are evaluated on the same inputs: same face tracks, scripts and characters. Table 1 shows the Accuracy (Acc.) and Average Precision (AP) of our approach compared to other methods on the Casablanca benchmark [4]. In particular we compare to Parkhi *et al.* [27] which is a strong baseline using the same CNN face descriptors as in our method. We also show the AP of classifying the background character class (Background AP). We compare in Table 2 our approach to other methods [27, 35] reporting results on season 5 of the TV series "Buffy the Vampire Slayer". Both of these methods [27, 35] use speaker detection to mine additional strong (but possibly incorrect) labels from the script, which we also incorporate (as additional bags) to make the comparison fair. Our method demonstrates significant improvement over the previous results. It also outperforms other methods on the task of classifying background characters. Finally, Table 3 shows the sensitivity to hyper-parameter α from the background constraint (13) on the Casablanca benchmark. Note that in contrast to other methods, our background model does not require supervision for the background class. This clearly demonstrates the advantage of our proposed background model. For all experiments the hyper-parameter α of the background constraint (13) was set to 30%. Figure 5 illustrates our qualitative results for character recognition in different movies.

METHOD	# movies	Joint-Model	St.U.	E.	S.D.	Si.U.	H.S.	F.	G.C.	K.	H.	A.	R.	O.D.	D.	mAP
(a) Random	∅	No	0.9	0.1	0.7	0.1	0.1	0.6	0.2	0.3	0.5	0.2	1.8	0.8	0.4	0.5
(b) Script only	∅	No	3.0	4.3	5.5	2.8	4.7	2.5	1.6	11.3	4.2	1.4	13.7	3.1	3.0	4.7
(c) Fully-supervised	4	No	21.2	0.2	22.2	0.9	0.6	7.3	1.4	1.9	**4.5**	2.0	33.2	18.5	6.3	9.3
(d) Few training movies	5	Yes	22.6	9.6	15.6	**8.1**	9.7	6.1	1.0	6.0	2.1	4.2	44.0	16.2	15.9	12.4
(e) No Joint Model	66	No	10.7	7.0	17.1	7.3	**18.0**	**12.6**	**2.0**	**14.9**	3.6	**5.8**	24.4	14.2	24.9	12.5
(f) Full setup	66	Yes	**27.0**	**9.8**	**28.2**	6.7	7.8	5.9	1.0	12.9	1.7	5.7	**56.3**	**21.3**	**29.7**	**16.4**

Table 4: Average Precision of actions evaluated on 5 movies. (St.U: `Stand Up`, E.: `Eat`, S.D: `Sit Down`, Si.U.: `Sit Up`, H.S: `Hand Shake`, F.: `Fight`, G.C.: `Get out of Car`, K.: `Kiss`, H.: `Hug`, A.: `Answer Phone`, R.: `Run`, O.D.: `Open Door`, D.: `Drive`)

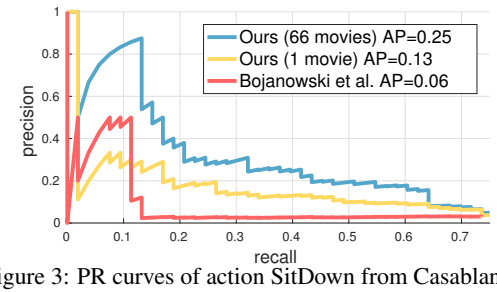

Figure 3: PR curves of action SitDown from Casablanca.

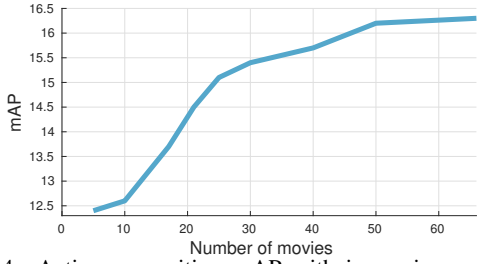

Figure 4: Action recognition mAP with increasing number of training movies.

5.2. Evaluation of action recognition

First, we compare our method to Bojanowski *et al.* 2013 [4]. Their evaluation uses different body tracks than ours, we design here an algorithm-independent evaluation setup. We compare our model using the Casablanca movie and the `Sit Down` action. For the purpose of evaluation, we have manually annotated all person tracks in the movie and then manually labeled whether or not they contain the `Sit Down` action. Given this ground truth, we assess the two models in a similar way as typically done in object detection. Figure 3 shows a precision-recall curve evaluating recognition of the `Sit Down` action. We show our method trained on Casablanca only (as done in [4]) and then on all 66 movies. Our method trained on Casablanca is already better than [4]. The improvement becomes even more evident when training our method on all 66 movies.

To evaluate our method on all 13 action classes, we use five movies (American Beauty, Casablanca, Double Indemnity, Forrest Gump and Fight Club). For each of these movies we have manually annotated all person tracks produced by our tracker according to 13 target action classes and the background action class. We assume that each track corresponds to at most one target action. In rare cases where this assumption is violated, we annotate the track by one of the correct action classes.

Method	R@1	R@5	R@10	Median Rank
Yu *et al.* [41]	3.6%	14.7%	23.9%	**50**
Levi *et al.* [24]	4.7%	15.9%	23.4%	64
Our baseline	**7.3%**	**19.2%**	**27.1%**	52

Table 5: Baseline comparison against winners of the LSMDC2016 movie clip retrieval challenge

In Table 4 we compare results of our model to different baselines. The first baseline (**a**) corresponds to the random assignment of action classes. The second baseline (**b**) Script only uses information extracted from the scripts: each time an action appears in a bag, all person tracks in this bag are then simply annotated with this action. Baseline (**c**) is using our action descriptors but trained in a fully supervised set-up on a small subset of annotated movies. To demonstrate the strength of this baseline we have used the same action descriptors on the LSMDC2016[1] movie clip retrieval challenge. This is the largest public benchmark [30] related to our work that considers movie data (but without person localization as we do in our work). Table 5 shows our features employed in simple CCA method as done in [24] achieving state-of-the-art on this benchmark. The fourth baseline (**d**) is our method train only *using the five evaluated movies*. The fifth baseline (**e**) is our model without the joint person-action constraint (11), but still trained on all 66 movies. Finally, the last result (**f**) is from our model using all the 66 training movies and person-action constraints (11). Results demonstrate that optimizing our model on more movies brings the most significant improvement to the final results. We confirm the idea from [4] that adding the information of who is performing the action in general helps identifying actions. However we also notice it is not always true for actions with interacting people such as: `Fight`, `Hand Shake`, `Hug` or `Kiss`. Knowing who is doing the action does not seems to help for these actions. Figure 4 shows improvements in action recognition when gradually increasing the number of training movies. Figure 6 shows qualitative results of our model on different movies. The statistics about the ground truth and constraints together with additional results are provided in **Appendix**.

[1] https://sites.google.com/site/describingmovies/lsmdc-2016

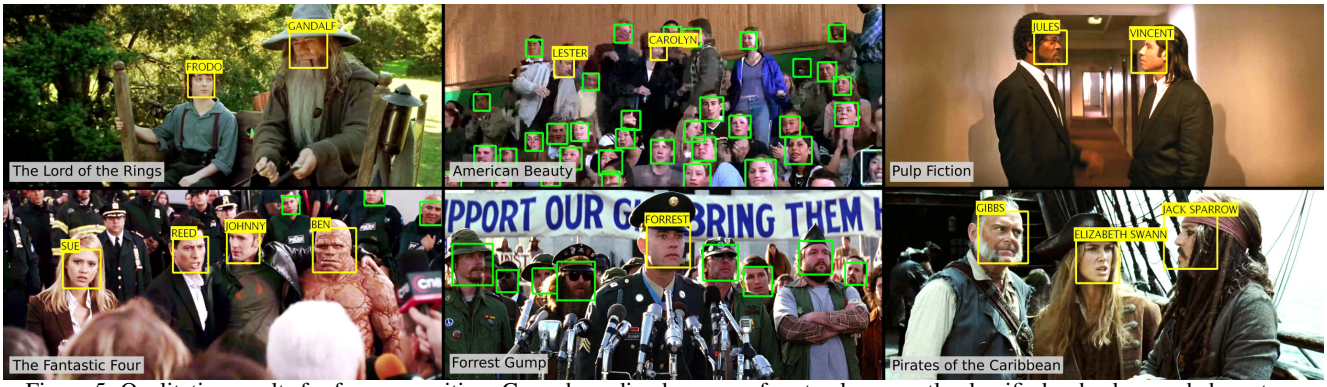

Figure 5: Qualitative results for face recognition. Green bounding boxes are face tracks correctly classified as background characters.

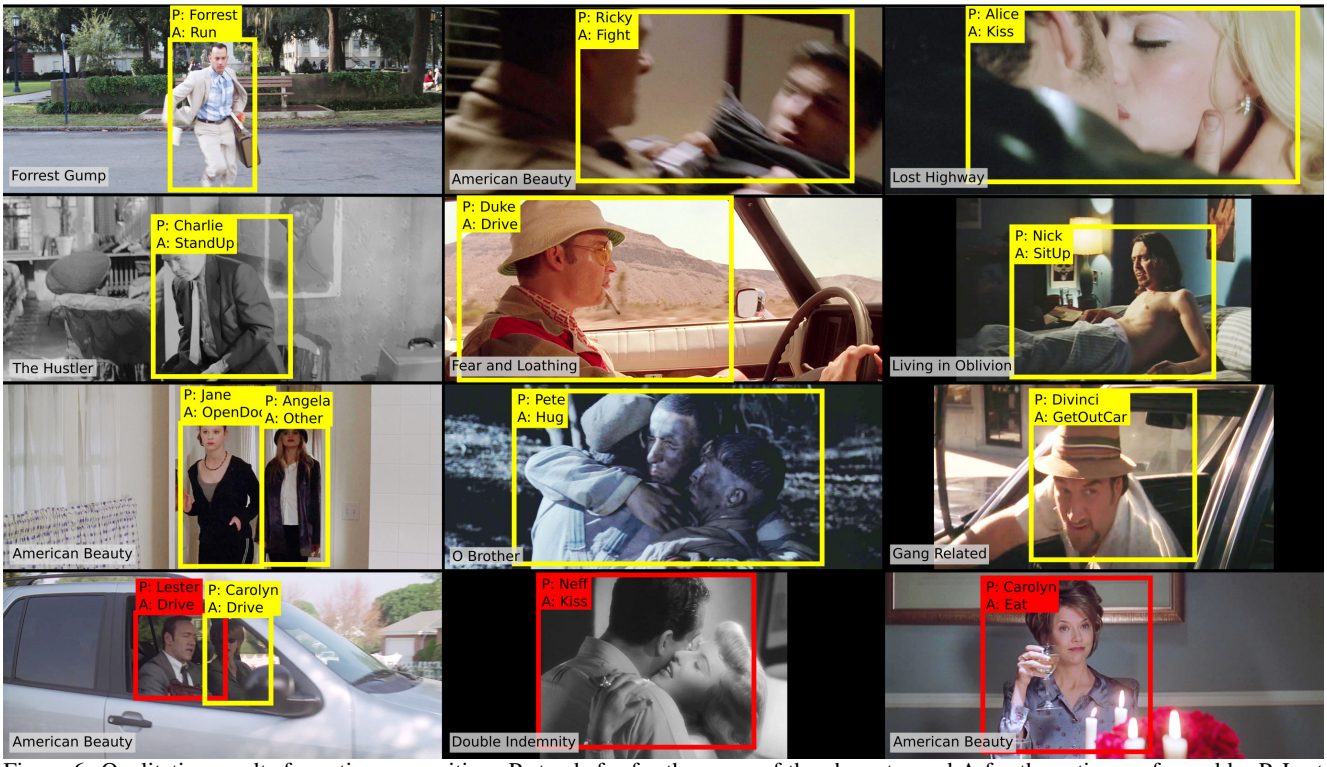

Figure 6: Qualitative results for action recognition. P stands for for the name of the character and A for the action performed by P. Last row (in red) shows mislabeled tracks with high confidence (*e.g.* hugging labeled as kissing, sitting in a car labeled as driving).

6. Conclusion

We have proposed an efficient online optimization method based on the Block-Coordinate Frank-Wolfe algorithm. We use this new algorithm to scale-up discriminative clustering model in the context of weakly-supervised person and action recognition in feature-length movies. Moreover, we have proposed a novel way of handling the background class, which does not require collecting background class data as required by the previous approaches, and leads to better performance for person recognition. In summary, the proposed model significantly improves action recognition results on 66 feature-length movies. The significance of the technical contribution goes beyond the problem of person-action recognition as the proposed optimization algorithm can scale-up other problems recently tackled by discriminative clustering. Examples include: unsupervised learning from narrated instruction videos [1], text-to-video alignment [6], co-segmentation [16], co-localization in videos and images [18] or instance-level segmentation [31], which can be now scaled-up to an order of magnitude larger datasets.

Acknowledgments. This work has been supported by ERC grants ACTIVIA (no. 307574) and LEAP (no. 336845), CIFAR Learning in Machines & Brains program, ESIF, OP Research, development and education Project IMPACT No. CZ.02.1.01/0.0/0.0/15_003/0000468 and a Google Research Award.

References

[1] J.-B. Alayrac, P. Bojanowski, N. Agrawal, I. Laptev, J. Sivic, and S. Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *CVPR*, 2016. 2, 3, 8

[2] F. Bach and Z. Harchaoui. Diffrac: a discriminative and flexible framework for clustering. In *NIPS*, 2007. 2, 3, 4

[3] T. L. Berg, A. C. Berg, J. Edwards, M. Maire, R. White, Y.-W. Teh, E. Learned-Miller, and D. A. Forsyth. Names and faces in the news. In *CVPR*, volume 2, pages II–848, 2004. 1

[4] P. Bojanowski, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Finding Actors and Actions in Movies. In *ICCV*, 2013. 1, 2, 3, 4, 5, 6, 7

[5] P. Bojanowski, R. Lajugie, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Weakly supervised action labeling in videos under ordering constraints. In *ECCV*, 2014. 1, 2, 3

[6] P. Bojanowski, R. Lajugie, E. Grave, F. Bach, I. Laptev, J. Ponce, and C. Schmid. Weakly-supervised alignment of video with text. In *ICCV*, 2015. 2, 3, 8

[7] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970, 2015. 1

[8] T. Cour, B. Sapp, C. Jordan, and B. Taskar. Learning from ambiguously labeled images. In *CVPR*, 2009. 2, 6

[9] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *ICCV*, 2009. 1, 2

[10] M. Everingham, J. Sivic, and A. Zisserman. "Hello! My name is... Buffy" – Automatic Naming of Characters in TV Video. In *BMVC*, 2006. 1, 2, 6

[11] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 1956. 2

[12] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models, release 5. 6

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 6

[14] M. Honnibal and M. Johnson. An improved non-monotonic transition system for dependency parsing. In *EMNLP*, 2015. 6

[15] M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML*, 2013. 2

[16] A. Joulin, F. Bach, and J. Ponce. Discriminative Clustering for Image Co-segmentation. In *CVPR*, 2010. 2, 8

[17] A. Joulin, F. Bach, and J. Ponce. Multi-class cosegmentation. In *CVPR*, 2012. 2

[18] A. Joulin, K. Tang, and L. Fei-Fei. Efficient image and video co-localization with frank-wolfe algorithm. In *ECCV*, 2014. 2, 8

[19] A. Joulin, K. Tang, and L. Fei-Fei. Efficient image and video co-localization with Frank-Wolfe algorithm. In *ECCV*, 2014. 3

[20] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014. 1

[21] S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher. Block-coordinate frank-wolfe optimization for structural SVMs. In *ICML*, 2013. 1, 2, 3

[22] T. Lan, Y. Wang, and G. Mori. Discriminative figure-centric models for joint action localization and recognition. In *ICCV*, 2011. 2

[23] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 1, 5

[24] G. Levi, D. Kaufman, L. Wolf, and T. Hassner. Video Description by Combining Strong Representation and a Simple Nearest Neighbor Approach. In *ECCV LSMDC2016 Workshop*, 2016. 7

[25] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. In *ECCV*, 2014. 6

[26] A. Osokin, J.-B. Alayrac, I. Lukasewitz, P. Dokania, and S. Lacoste-Julien. Minding the gaps for block frank-wolfe optimization of structured svms. In *ICML*, 2016. 4

[27] O. Parkhi, E. Rahtu, and A. Zisserman. It's in the bag: Stronger supervision for automated face labelling. In *ICCV Workshop*, 2015. 2, 5, 6

[28] O. Parkhi, A. Vedaldi, and A. Zisserman. Deep Face Recognition, British Machine Vision Conference. In *BMVC*, 2015. 6

[29] V. Ramanathan, A. Joulin, P. Liang, and L. Fei-Fei. Linking people in videos with "their" names using coreference resolution. In *ECCV*, 2014. 2

[30] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele. A dataset for movie description. In *CVPR*, 2015. 1, 7

[31] G. Seguin, P. Bojanowski, R. Lajugie, and I. Laptev. Instance-level video segmentation from object tracks. In *CVPR*, 2016. 2, 3, 8

[32] R. Shaoqing, H. Kaiming, G. Ross, and S. Jian. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 6

[33] N. Shapovalova, A. Vahdat, K. Cannons, T. Lan, and G. Mori. Similarity constrained latent support vector machine: An application to weakly supervised action classification. In *ECCV*, 2012. 2

[34] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014. 1

[35] J. Sivic, M. Everingham, and A. Zisserman. "Who are you?" - Learning person specific classifiers from video. In *CVPR*, 2009. 1, 2, 6

[36] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 1

[37] M. Tapaswi, M. Bauml, and R. Stiefelhagen. "Knock! Knock! Who is it?" probabilistic person identification in tv-series. In *CVPR*, 2012. 1

[38] H. Wang and C. Schmid. Action Recognition with Improved Trajectories. In *ICCV*, 2013. 1, 6

[39] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, pages 4305–4314, 2015. 1

[40] P. Weinzaepfel, X. Martin, and C. Schmid. Towards weakly-supervised action localization. *arXiv preprint arXiv:1605.05197*, 2016. 2

[41] Y. Yu, H. Ko, J. Choi, and G. Kim. Video captioning and retrieval models with semantic attention. In *ECCV LSMDC2016 Workshop*, 2016. 7

A. Appendix

This appendix contains details and supplementary results to the main paper.

A.1. Slack variables

To account for imprecise information in movie scripts, we add slack variables to our constraints. We penalize the values of slack variables with the L_2 penalty. The slack-augmented constraints are defined as:

$$\sum_{n \in \mathcal{N}} Z_{np} \geq 1 - \xi, \tag{16}$$

$$\sum_{n \in \mathcal{N}} T_{na} \geq 1 - \xi, \tag{17}$$

$$\sum_{n \in \mathcal{N}} Z_{np}\, T_{na} \geq 1 - \xi, \tag{18}$$

where ξ is the slack variable.

A.2. Lower bound

In practice, we noticed that modifying the value of the lower bound in constraints (16), (17), (18) from 1 to a higher value can significantly improve the performance of the algorithm. The constraints we use become:

$$\sum_{n \in \mathcal{N}} Z_{np} \geq \alpha_1 - \xi, \tag{19}$$

$$\sum_{n \in \mathcal{N}} T_{na} \geq \alpha_2 - \xi, \tag{20}$$

$$\sum_{n \in \mathcal{N}} Z_{np}\, T_{na} \geq \alpha_2 - \xi, \tag{21}$$

where $\alpha_1, \alpha_2 \in \mathbb{R}_+$ are hyper-parameters.

A.3. Combining face and body tracks

Let's denote $a_1, a_2, ..., a_n$, n faces tracks in the current shot and $b_1, b_2, ..., b_m$ the m body tracks in this same shot (we assume $m \geq n$). We want to model that each face track is associated to at most one body track but a body track does not necessary have a face track, as the face of a person may not always be visible. Let's also define the following overlap measure O between a face track a and a body track b. If $\mathcal{A}$ is a set of all frames of the track a and $a(t)$, $b(t)$ are bounding boxes of tracks a and b at frame t, we have:

$$O(a, b) = \sum_{t \in \mathcal{A}} \frac{Area(a(t) \cap b(t))}{Area(a(t))}. \tag{22}$$

We compute the overlap for all possible pairs $O(a_i, b_j)$, where $i \in [1, n]$ and $j \in [1, m]$. Then we associate each face track a_i with the body track b_j that maximizes $O(a_i, b_j)$. Finally, for each body track b_j we either do not have any associated face track (then the body track won't have a match) or have multiple face tracks a_i associated to it. In the latter case, we match the body track b_j with the face track a_i that maximizes $O(a_i, b_j)$.

β	0	0.1	0.2	0.3	0.4	0.5	0.6	0.75	0.8
mAP	15.0	15.7	15.9	15.8	16.6	16.1	16.2	16.0	15.5

Table 6: Influence of the hyper-parameter β (13) for action recognition.

A.4. Sensitivity to the background constraint hyperparameter for action recognition

Table 6 shows the low sensitivity of the action recognition results to the β (15) hyper-parameter on the action recognition results.

A.5. Additional dataset statistics

Table 7 provides the number of action constraints we extracted from 66 movie scripts. It also shows the number of ground truth intervals for each action we obtained by an exhaustive manual annotation of human actions in five testing movies.

Our dataset contains the following 66 movies: American Beauty, As Good As It Gets, Being John Malkovich, Big Fish, Bringing Out the Dead, Bruce the Almighty, Casablanca, Charade, Chasing Amy, Clerks, Crash, Dead Poets Society, Double Indemnity, Erin Brockovich, Fantastic Four, Fargo, Fear and Loathing in Las Vegas, Fight Club, Five Easy Pieces, Forrest Gump, Gandhi, Gang Related, Get Shorty, Hudsucker Proxy, I Am Sam, Independence Day, Indiana Jones and the Last Crusade, It Happened One Night, Jackie Brown, Jay and Silent Bob Strike Back, LA Confidential, Legally Blonde, Light Sleeper, Little Miss Sunshine, Living in Oblivion, Lone Star, Lost Highway, Men In Black, Midnight Run, Misery, Mission to Mars, Moonstruck, Mumford, Ninotchka, O Brother, Pirates of the Caribbean Dead Mans Chest, Psycho, Pulp Fiction, Quills, Raising Arizona, Rear Window, Reservoir Dogs, The Big Lebowski, The Butterfly Effect, The Cider House Rules, The Crying Game, The Godfather, The Graduate, The Grapes of Wrath, The Hustler, The Lord of the Rings The Fellowship of the Ring, The Lost Weekend, The Night of the Hunter, The Pianist, The Princess Bride, Truman Capote.

ACTION	# movies	Other	St.U.	E.	S.D.	Si.U.	H.S.	F.	G.C.	K.	H.	A.	R.	O.D.	D.	Total
Ground truth	5	14532	146	24	112	19	28	90	26	47	74	28	277	131	59	15593
Constraints	66	$\emptyset$	237	85	146	46	49	70	81	244	44	99	156	208	169	1634

Table 7: Action recognition ground truth and constraint statistics. (St.U: `Stand Up`, E.: `Eat`, S.D: `Sit Down`, Si.U.: `Sit Up`, H.S: `Hand Shake`, F.: `Fight`, G.C.: `Get out of Car`, K.: `Kiss`, H.: `Hug`, A.: `Answer Phone`, R.: `Run`, O.D.: `Open Door`, D.: `Drive`)