Action recognition using saliency learned from recorded human gaze

Daria Stefic and Ioannis Patras

School of Electronic Engineering and Computer Science
Queen Mary, University of London
London, E1 4NS

Abstract

This paper addresses the problem of recognition and localization of actions in image sequences, by utilizing, in the training phase only, gaze tracking data of people watching videos depicting the actions in question. First, we learn discriminative action features at the areas of gaze fixation and train a Convolutional Network that predicts areas of fixation (i.e. salient regions) from raw image data. Second, we propose a Support Vector Machine-based recognition method for joint recognition and localization, in which the bounding box of the action in question is considered as a latent variable. In our formulation the optimization attempts to both minimize the classification cost and maximize the saliency within the bounding box. We show that the results obtained with the optimization where saliency within the bounding box is maximized outperform the results obtained when saliency within the bounding box is not maximized, i.e. when only classification cost is minimized. Furthermore, the results that we obtain outperform the state-of-the-art results on the UCF Sports dataset.

Keywords: action recognition, saliency, Support Vector Machine (SVM), latent variable, 3D Convolutional Neural Network (3D CNN)

1. Introduction

Action recognition in unsegmented images sequences can greatly benefit from attention mechanisms that reduce the influence of background clutter. Early

works on action recognition in this direction, such as the pioneering work of [1], used for this purpose spatiotemporal interest point (STIP) detectors. However, such detectors were designed and not learned. The human visual system on the other hand has built-in attention mechanisms. While their internal workings are not fully understood and transparent, their output, i.e. where human look, can be measured by gaze trackers.

In this paper, following recent works that utilize gaze information as side information for several Computer Vision tasks [2, 3, 4, 5, 6], we address the problem of action recognition and localization using, in the training phase only, gaze information. First, we learn a fixation prediction model, that is a model that learns how to predict where people look when presented with image sequences. We treat this is a supervised binary classification problem, and train a Convolutional Neural Network that takes as input a local 3D spatiotemporal cuboid and returns as an output a (soft) label that could be interpreted as a saliency measure for the cuboid in question. We then learn features by training in a supervised way a 3D convolutional neural network that extracts compact features on local cuboids. Given that humans tend to look at the important and discriminative parts of the action video[7, 8], we train our network only on cuboids extracted around recorded gaze fixations. In order to show that the proposed saliency prediction model and the extracted features can be useful for the problem of action recognition, we use them in a simple action recognition scheme in which local cuboids are classified to one of the action classes independently and the video class is decided according to a majority voting scheme. That is a test video is assigned to the class to which most of its cuboids have been classified to.

Finally, we propose a novel SVM scheme for joint recognition and localization of actions. In our proposed SVM model we are introducing as latent variables the locations of the bounding boxes within which the actions are assumed to take place. Those locations are unknown during both training and testing. During testing/inference the proposed method optimizes with respect to both the video label (i.e. action class) and the location of the bounding boxes a cost that

comprises of two terms. First, a classical SVM misclassification penalty term and second, a term that is related to the saliency within the bounding box. The proposed SVM tries to find the class label and the location of bounding box that optimally balance the minimization of the missclassification cost once a linear classifier is applied on features extracted within the bounding box in question, and the maximization of the sum of the predicted saliencies within the bounding box. The proposed scheme shows improvements over the baseline SVM, over the latent SVM introduced in [9] and over the proposed SVM that does not use the additional saliency cost. It also achieves state-of-the-art performance on the UCF sports dataset.

Figure 1 shows an overview of the inference procedure of our proposed system for a single video. In the first phase (PHASE 1 in Figure 1) we use the outputs of two separate 3D CNNs - one for saliency prediction and another one for feature extraction. The simple majority voting-based classification scheme is depicted as PHASE 2A in Figure 1 and the SVM-based joint action classification-localization is depicted as PHASE 2B.

The main contributions of this paper are the following:

- we present a fully supervised method for learning saliency prediction using human gaze information and show the usefulness of saliency prediction in a majority voting and an SVM framework,
- we present a fully supervised method for learning action features using human gaze information and show that those features outperform commonly used handcrafted features in a majority voting framework,
- we present a latent SVM based method for joint action recognition and localization in which the class label and the bounding boxes are inferred by optimizing a cost function that contains a missclassification penalty term and a term that that is related to the saliency within the bounding box. We show that our joint localization and recognition model that uses predicted saliency for bounding box inference during training is better

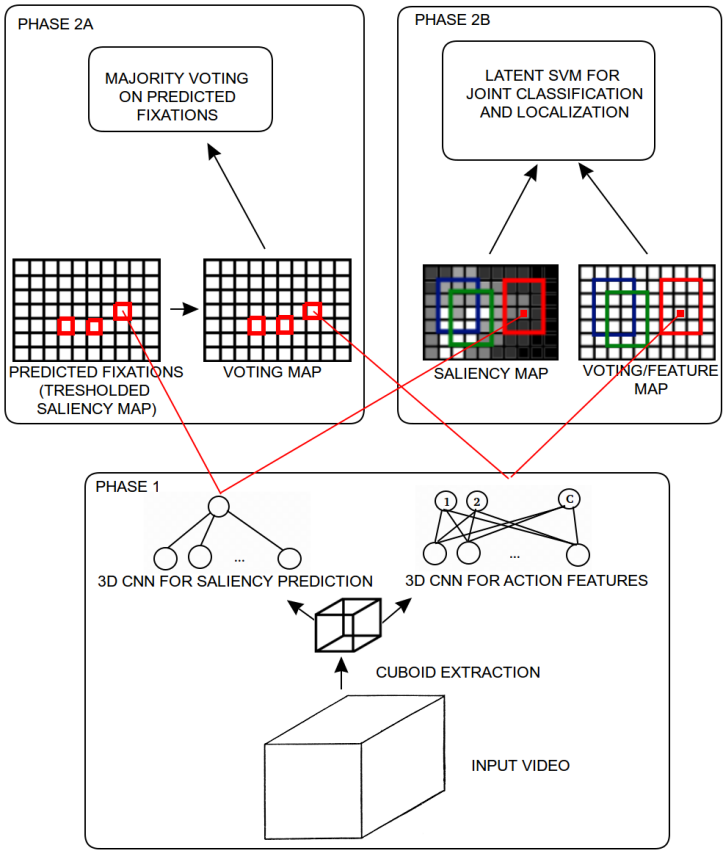

Figure 1: Illustration of two phases in action class inference procedure for the whole video

than both the model that does not use predicted saliency and the model of [9].

The rest of the paper is organized as follows. In section 2, we present related work in action recognition focusing on how saliency has been used to alleviate some of the major challenges. In section 3, we present how we learn features for action recognition and how we learn a saliency predictor. In section 4, we present the proposed SVM framework that uses those action features and a saliency predictor to infer the class of the video and the location of the bounding boxes that contain relevant parts of the video. In section 5, we present experimental

results and in section 6 we give conclusions and future work.

2. Related work

There is a large body of works in the area of action recognition - for recent surveys we refer the reader to [10, 11, 12]. In what follows, we briefly review some the major works in the field and then focus on works that are closer related to the contributions that we make in this paper, namely, works on feature learning, works on joint localization and recognition, focusing on latent SVM formulations, and finally, works that use gaze in the action recognition framework.

In the classical pipeline for human action recognition the first step is feature extraction. Usually features based on shape, such as HOG and Scale-Invariant Feature Transform (SIFT), or motion, such as HOF and Histogram of the Oriented edges of the Motion Boundaries (MoBH), are extracted across the areas detected by local STIP detectors. In such approach, both the feature detector and feature descriptor usually act locally. Lately, tracking and extracting trajectory features has shown very good performance in the action recognition[13, 14, 15, 16, 17, 18]. The second step is feature encoding. Until recently a simple BoW approach was most popular: the features extracted around detected areas are quantized and a BoW representation of the whole video is built. Lately, using Fisher Vector (FV) encoding [19] has shown superior results comparing to the BoW approach, in image recognition[20] and action recognition[21, 22, 16, 17]. In the third step an SVM is used as a classifier on top of the video representation.

In such approach local changes can be captured but more complex global spatiotemporal relations and higher level motion patterns are lost by pooling in the feature encoding stage. For that purpose, probabilistic graphical models such as Conditional Random Fields (CRFs)[23, 24, 25, 26], Hidden Markov Models (HMMs)[27, 28, 29, 30], probabilistic Latent Semantic Analysis (pLSA) and Latent Dirichlet Allocation (LDA)[31] can be used. Action grammars[32,

33, 34, 35], models that use graph relations[18, 36] and latent SVMs[37, 38, 39] are also used to model the higher levels of action recognition frameworks.

2.1. Descriptors and feature learning

Much effort has been put in enriching local descriptors and detectors with for example, hierarchical structures [40, 41, 42], local contexts [43, 44], or extending them to 3D [45, 46, 47]. An approach that is focused on *learning* feature representations, namely deep learning, has shown very good performance in various computer vision tasks [48, 49, 50, 51]. In [41, 52] features for action recognition are learned using unsupervised learning algorithms: [41] uses Independent Subspace Analysis and [52] uses Slow Feature Analysis. Supervised deep learning algorithms, such as 3D CNN, have also been successfully applied in the problem of action recognition [53, 54, 51]. In those works 3D CNNs are applied in a holistic manner. However, the inputs to these networks are segmented video sequences. Finally, [55] adopts the two level discriminative learning approach which is conceptually similar to ours. In [55] the discriminative mid-level features are learned using an exemplar SVM on low-level features. Those features are used to build a global video representation which is then fed into a linear SVM. However, the selection of discriminative parts of the videos that are used to train an exemplar SVM is manual.

2.2. Higher level modeling of the video structure

In the last few years, inspired by the work of [9] on part-based object recognition, several works introduce latent variables in the action recognition framework. [37, 39] try to jointly localize and recognize the actions in a latent SVM framework, and [56, 30] use latent variables to model the temporal structure of activities. Finally, a recent work that is close to ours is [38] where recorded human gaze is incorporated in the structured output latent SVM framework. The gaze was used only in the training phase in a form of a loss in the structured prediction to infer the latent variable that determines the bounding boxes of the action.

2.3. Gaze as an interest point detector

A couple of recent works use gaze information for action recognition as a STIP detector[7, 8]. This can be done in two ways: first by using ground truth recorded fixations as a STIP detector in test videos [8], and second by using ground truth fixations to train a detector on a training set and use the predicted fixations as a STIP in test videos [7]. [8] showed that results for action recognition obtained using the former approach in the test video classification procedure outperformed the results obtained using commonly used STIPs. However, this method requires gaze information in test videos and this is not always available.

Other works, such as [7], learns to predict the human fixations on test videos using recorded fixations only on training videos. They pose saliency prediction problem as a binary classification problem, i.e. fixated points are treated as positive examples, and non-fixated points are treated as negative examples. To solve this classification problem, a linear SVM is used on top of manually selected features extracted around points in question. Action recognition results obtained when using fixations predicted by this saliency predictor as a STIP did not outperformed the ones obtained when using ground truth recorded fixations as a STIP, but they did outperform the results obtained when using common STIPs. This is consistent with the results of [8].

3. Learning action features and saliency prediction

The modified SVM-based classifier for joint action recognition and localization that we propose (see Figure 1), utilizes as input the outputs of two 3D CNNs. The first 3D CNN predicts saliency and the second 3D CNN extracts action features. Recorded human fixations are required for training both of those networks. In section 3.1 we will describe the first 3D CNN and how the extracted local action features are used in a global SVM framework. In section 3.2 we will describe the second 3D CNN and how the predicted saliency is used as an additional cost in the proposed SVM framework.

Table 1: Parameters of 3D-CNNs

Parameters	saliency	action features
input cube dimensionality	21x21x10	21x21x10
size of 1st layer filters	4x4x3	4x4x3
1st layer subsampling	2x2x1	2x2x1
number of 1st layer filters	25	50
size of 2nd layer filters	2x2x3	2x2x3
2nd layer subsampling	2x2x1	2x2x1
number of 2nd layer filters	50	100
units in fully connected layer	50	50

3.1. 3D CNN for learning discriminative mid-level local action features

Similar to other works in the literature of Deep Learning we extract features by training a CNN in a supervised way. The architecture of the proposed CNN (Figure 2) has 10 outputs in the last layer, one for each of the 10 action classes and takes as input 3D cuboids extracted around points at which humans fixate. By training this network only on fixated points we learn discriminative features discarding the background clutter. Once the 3D CNN is trained, when presented with a 3D cuboid it outputs a 10 dimensional vector containing the soft class labels for the cuboid in question.

The parameters of the 3D CNN architecture are listed in Table 1. We use $f(x) = \tanh(x)$ to model neurons output and spatial max pooling. In the final layer we use a softmax classifier. The implementation is made using Theano[57, 58]. We have used small filter sizes in order to reduce the number of parameters and make the training easier and a small number of units in the hidden layer in order to prevent overfitting. We have experimented with different number of filters and, as expected, found that the larger the number of filters the better (performance saturates for number of filters reported in Table 1). This will be further discussed in section 5.1.

In the proposed SVM framework for action recognition, we use as features

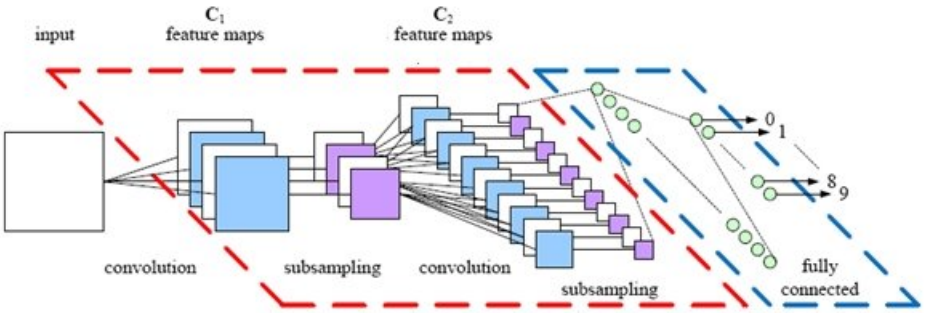

Figure 2: Convolutional neural network architecture used in our experiments, both for learning saliency prediction and action features. However, the number of outputs is different: the network for learning action features has $C = 10$ outputs (as depicted here) and the network for saliency prediction has two outputs. The input and the maps are depicted as 2D, in practice they are 3D.

the outputs of the uppermost softmax layer of the trained 3D CNN, that is, the soft class labels. In order to bring the videos to the same dimensionality, we sample cuboids at a regular 3D grid with a fixed number of points - clearly, we need to use different sampling steps for videos of different resolutions or number of frames. For each cuboid a C dimensional vector, where C is the number of action classes, is extracted after the 3D CNN is applied to it. For joint localization and recognition, we introduce latent variables that are bounding boxes with fixed dimensionality in space and in time, that is, they contain a fixed number of points of the 3D spatiomporal grid. The representation that we used for the whole video is the concatenation of the C-dimensional features at the spatiotemporal bounding box.

Formally, the representation of a video $\mathbf{x}^i$, denoted by $\mathbf{R}(\mathbf{x}^i, \mathbf{bb}^i)$, is a concatenation of features extracted at bounding boxes per frame $\mathbf{bb}^i = [bb_1 \ldots bb_t \ldots]^T$, $t = 1, \ldots, T$, where T is the number of frames. That is, $\mathbf{R}(\mathbf{x}^i, \mathbf{bb}^i)$ consists of

the concatenated features $\mathbf{r}(\mathbf{x}_t^i, bb_t^i)$ for each frame t. Formally:

$$\mathbf{R}(\mathbf{x}^i, \mathbf{bb}^i) = [\mathbf{r}(\mathbf{x}_1^i, bb_1^i)^T \ \ ... \ \ \mathbf{r}(\mathbf{x}_t^i, bb_t^i)^T \ \ ...]^T, \ t = 1, \ ..., \ T, \tag{1}$$

where $\mathbf{r}(\mathbf{x}_t^i, bb_t^i)$ is the concatenation of features extracted by the pretrained 3D-CNN within the bounding box bb_t^i in frame t, that is:

$$\mathbf{r}(\mathbf{x}_t^i, bb_t^i) = [... \ \mathbf{a}(p; \theta_a)^T \ ...]^T, \quad p \in bb_t^i, \tag{2}$$

where p is a point within the bounding box and $\mathbf{a}(p; \theta_a)$ is a vector of features extracted at point p using the 3D CNN (with parameters θ_a) trained for action cube classification. The feature vector $\mathbf{a}(p; \theta_a)$ is the output of the softmax layer of the 3D CNN and contains the probabilities that the cuboid p belongs to each of the class c. That is:

$$\mathbf{a}(p; \theta_a) = [P(Y = 1|p, \theta_a), \ ..., \ P(Y = c|p, \theta_a), \ ..., \ ]^T, \tag{3}$$

where c is an action class. Clearly, the dimensionality of $\mathbf{a}(p; \theta_a)$ is equal to the number of classes.

3.2. 3D CNN for saliency prediction

Very recently, deep learning has been applied for saliency prediction, either in an unsupervised way [59] or in a supervised way using recorded ground truth fixations obtained by gaze tracking [60, 61, 62, 63]. In this paper we adopt the supervised approach and use a 3D CNN. The network acts as a binary classifier that classifies cuboids as being fixations or not.

The architecture for saliency prediction is the same as the one for learning action features depicted in Fig. 2, the only difference being that it has only two outputs in the last layer and fewer number of filters in both layers. The parameters of this 3D CNN architecture are listed in Table 1. The output of the softmax layer of the network for saliency prediction will be incorporated in the total cost of the SVM classifier, as described in the following paragraph.

In our SVM framework we want to avoid choosing bounding boxes $\mathbf{bb}^i$ with low concentration of saliency. Hence, for each video $\mathbf{x}^i$, we add a cost which is

defined in terms of the saliency concentration in the inferred bounding boxes. The saliency concentration within the bounding box $\mathbf{bb}^i$ for the video $\mathbf{x}^i$ is defined as:

$$M(\mathbf{x}^i, \mathbf{bb}^i) = \sum_{t=1}^{T} m(\mathbf{x}_t^i, bb_t^i), \tag{4}$$

where $m(\mathbf{x}_t^i, bb_t^i)$ is the saliency concentration at bounding box bb_t^i at frame t defined as:

$$m(\mathbf{x}_t^i, bb_t^i) = \frac{\sum_{p \in bb_t^i} s(p; \theta_s)}{\sum_{p \in bb_t^i} 1} \tag{5}$$

where $s(p; \theta_s)$ is the estimated saliency of a point p using the parameters θ_s of the 3D CNN that is trained for saliency prediction, that is:

$$s(p; \theta_s) = P(Y = +1 | p, \theta_s), \tag{6}$$

where $P(Y = +1 | p, \theta_s)$ is the output of the softmax layer of the network for saliency prediction, that is the probability of a point p being a fixation. Hence, $s(p; \theta_s) \in [0, 1]$.

4. SVM formulation

First, in section 4.1, we will present the proposed SVM-based classifier, the parameters over which its cost function is optimized, and how the representation of a video described in section 3.1 and the saliency cost described in 3.2 are incorporated in the total cost. Second, in section 4.2, we will present the optimization procedure for this type of SVM, and how the inference of bounding boxes and video class is incorporated in it. Third, in section 4.3, we will present how the classification of a video is performed. Finally, in section 4.4, we will present how a special case of our model compares to [9].

4.1. Cost function

Here, we define the problem formally. Let $\mathbf{x}^i$ be a video of T frames and $\mathbf{bb}^i = [bb_1^i, \ ..., \ bb_t^i, \ ..., \ bb_T^i]^T$ bounding boxes per frame, that ideally contain

discriminative information for action classification. Let $\mathbf{R}(\mathbf{x}^i, \mathbf{bb}^i)$ be a representation of the video in question, as described in section 3.1. Typical systems, such as [41, 43], assume that the information $\mathbf{bb}^i$ is given, and adopt a video classification scheme such as $\mathbf{w}^T\mathbf{R}(\mathbf{x}^i, \mathbf{bb}^i) + b$, where $\mathbf{w}, b$ are learned using, for example, max-margin learning. For example, $\mathbf{bb}^i$ can be given in a form of a STIP detector that is used in order to sample the cuboids around salient points. The representation $\mathbf{R}(\mathbf{x}^i, \mathbf{bb}^i)$ that is built using those points is then fed into an SVM classifier and only SVM parameters are learned. By contrast, we treat the locations of the bounding boxes $\mathbf{bb}^i$ as latent variables and solve the optimization problem in which we are searching not only for the optimal values of the standard SVM parameters, but also for the optimal locations of the bounding boxes.

Given a set of labeled videos $D = \{(\mathbf{x}_1, y_1),\ ...,\ (\mathbf{x}_M, y_M)\}$ where $y_i \in \{-1, 1\}$ and M is the number of videos, we are solving the following optimization problem:

$$\min_{\mathbf{w}, b, \{\mathbf{bb}^i\}_{i=1}^M} L_D(\mathbf{w}, b, \{\mathbf{bb}^i\}_{i=1}^M), \tag{7}$$

where

$$L_D(\mathbf{w}, b, \{\mathbf{bb}^i\}_{i=1}^M) = \frac{1}{2}(\mathbf{w}^T\mathbf{w} + b^2) + \tag{8}$$

$$\sum_{i=1}^{M} \big[\max(0, 1 - y^i f(\mathbf{w}, b; \mathbf{x}^i, \mathbf{bb}^i)) - \lambda M(\mathbf{x}^i, \mathbf{bb}^i)\big]. \tag{9}$$

In the above equation $f(\mathbf{w}, b; \mathbf{x}^i, \mathbf{bb}^i)$ is the scoring function for a video $\mathbf{x}^i$:

$$f(\mathbf{w}, b; \mathbf{x}^i, \mathbf{bb}^i) = \mathbf{w}^T\mathbf{R}(\mathbf{x}^i, \mathbf{bb}^i) + b, \tag{10}$$

and $\mathbf{w}$ are concatenated weights per frame:

$$\mathbf{w} = [\mathbf{w}_1^T,\ \ ...,\ \ \mathbf{w}_t^T,\ \ ...,\ \mathbf{w}_T^T]^T. \tag{11}$$

Note that the additional cost $M(\mathbf{x}^i, \mathbf{bb}^i)$ related to the saliency of a bounding box areas is regularized by the parameter λ.

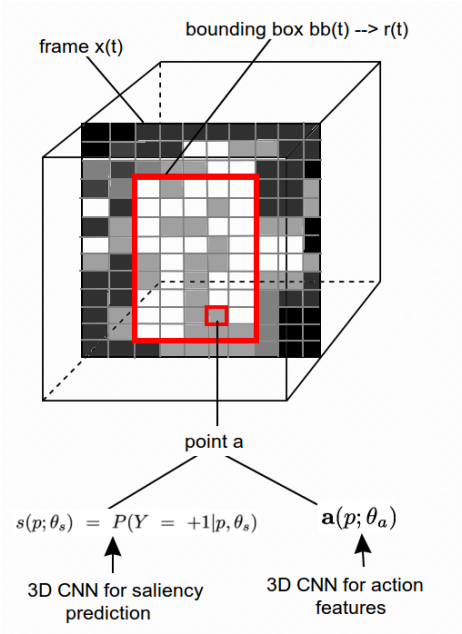

Figure 3: Illustration of the proposed SVM. In each frame $\mathbf{x}_t$ bounding box bb_t is selected based on saliency concentration and features extracted across that bounding box. Based on the selected bounding box a frame representation $\mathbf{r}_t(\mathbf{x}_t, bb_t)$ is built by concatenating features across the bounding box. A video representation $\mathbf{R}(\mathbf{x}, \mathbf{bb})$ is further built by concatenating frame representations.

In Fig.3 we illustrate the working of our proposed SVM, that is the selection of the bounding box and the build of the input feature representation based on learned saliency and action features.

4.2. Learning

In order to minimize the cost function $L_D(\mathbf{w}, b, \{\mathbf{bb}^i\}_{i=1}^M)$, over two subsets of the parameters, namely $\mathbf{w}$ and b on the one hand and $\{\mathbf{bb}^i\}_{i=1}^M$ on the other, we use a block coordinate descent method. We iteratively alternate between optimizing the cost function with respect to the SVM parameters $\mathbf{w}, b$ keeping the bounding box parameters $\{\mathbf{bb}^i\}_{i=1}^M$ fixed (step 2) and optimizing the cost function with respect to the bounding box parameters $\{\mathbf{bb}^i\}_{i=1}^M$ keeping the

SVM parameters $\mathbf{w}, b$ fixed (step 3). Step 2 results to a convex optimization problem, more specifically an SVM-like problem that we solve with a gradient descent method. Step 3 is an optimization problem that can be solved by enumeration of the positions of the $\{\mathbf{bb}^i\}_{i=1}^M$ - an efficient exact solution is possible given that we do not consider interdependencies in subsequent frames (see Eq. (19) - Eq. (23)). Therefore, each step gives optimal solutions with respect to the subset of the parameters and the procedure converges to a local minimum.

The full procedure consists of the following steps:

Step 1. Initialization

We initialize the bounding box $(bb_t^i)^*$ at frame t as the one that maximizes the saliency:

$$(bb_t^i)^* = \underset{bb_t^i}{\operatorname{argmax}}\, m(\mathbf{x}_t^i, bb_t^i), \tag{12}$$

i.e. we are choosing the most salient areas of the videos. Note that this solution is actually the solution of a special case of our model when in the objective function the parameter $\lambda = +\infty$.

Step 2. Optimization with respect to $\mathbf{w}, b$

In this step we solve for $\mathbf{w}, b$ while keeping $\{\mathbf{bb}^i\}_{i=1}^M$ fixed. This results in a convex optimization problem that we solve by stochastic gradient descent. The subgradient of the objective function with respect to $\mathbf{w}$ is computed as follows:

$$\bigtriangledown_w L_D(\mathbf{w}, b, \{\mathbf{bb}^i\}_{i=1}^M) = \mathbf{w} + C\sum_i h_{\mathbf{w}}(\mathbf{w}, \mathbf{x}_i, y_i), \tag{13}$$

where

$$h_{\mathbf{w}}(\mathbf{w}, \mathbf{x}_i, y_i) = \begin{cases} 0, & \text{if} \quad y_i f(\mathbf{w}, b; \mathbf{x}^i, (\mathbf{bb}^i)^*) \geq 1, \\ y_i \mathbf{R}(\mathbf{x}_i, (\mathbf{bb}_i)^*), & \text{otherwise.} \end{cases} \tag{14}$$

and the subgradient with respect to b:

$$\bigtriangledown_b L_D(\mathbf{w}, b, \{\mathbf{bb}^i\}_{i=1}^M) = b + C\sum_i h_b(b, \mathbf{x}_i, y_i), \tag{15}$$

where

$$h_b(b, \mathbf{x}_i, y_i) = \begin{cases} 0, & \text{if} \quad y_i f(\mathbf{w}, b; \mathbf{x}^i, (\mathbf{bb}^i)^*) \geq 1, \\ y_i, & \text{otherwise.} \end{cases} \tag{16}$$

The full gradient descent algorithm for optimizing $L_D(\mathbf{w}, b, \{\mathbf{bb}^i\}_{i=1}^M)$ over $\mathbf{w}, b$ is summarized in Algorithm 1.

Algorithm 1 Stochastic gradient descent for $[\mathbf{w}, b]$ optimization

pick a random example i

use $(\mathbf{bb}^i)^*$ computed in the previous step to calculate $f(\mathbf{w}, b; \mathbf{x}^i, (\mathbf{bb}^i)^*)$

if $y_i f(\mathbf{w}, b; \mathbf{x}^i, (\mathbf{bb}^i)^*) \geq 1$ **then**

$\mathbf{w} \leftarrow \mathbf{w} - \alpha\mathbf{w}$,

$b \leftarrow b - \alpha b$

else

$\mathbf{w} \leftarrow \mathbf{w} - \alpha(\mathbf{w} - Cy^i\mathbf{R}(\mathbf{x}^i, (\mathbf{bb}^i)^*))$,

$b \leftarrow b - \alpha(b - Cy^i)$

end if

where α is learning rate.

The learning rate α is set to $0.05/it$, where it is the iteration index.

Step 3. Optimization with respect to bb

In this step we optimize $L_D(\mathbf{w}, b, \{\mathbf{bb}^i\}_{i=1}^M)$ with respect to $\{\mathbf{bb}^i\}_{i=1}^M$ by doing inference for every bounding box $\mathbf{bb}^i$ independently in the following way:

$$(\mathbf{bb}^i)^* = \underset{\mathbf{bb}^i}{\operatorname{argmin}} \big[\max(0, 1 - y^i f(\mathbf{w}, b; \mathbf{x}^i, \mathbf{bb}^i)) - \lambda M(\mathbf{x}^i, \mathbf{bb}^i) \big] \quad (17)$$

$$= \underset{\mathbf{bb}^i}{\operatorname{argmax}} (y^i f(\mathbf{w}, b; \mathbf{x}^i, \mathbf{bb}^i) + \lambda M(\mathbf{x}^i, \mathbf{bb}^i)) \quad (18)$$

Here we can see how the search for a bounding box balances between good feature response $f(\mathbf{w}, b; \mathbf{x}^i, \mathbf{bb}^i)$ across the bounding box area and a high saliency concentration $M(\mathbf{x}^i, \mathbf{bb}^i)$ of the same bounding box. Further, by applying $f(\mathbf{w}, b; \mathbf{x}^i, \mathbf{bb}^i) = \mathbf{w}^T \mathbf{R}(\mathbf{x}^i, \mathbf{bb}^i) + b$, the inference can be written as:

$$\underset{\mathbf{bb}^i}{\operatorname{argmax}} (y^i (\mathbf{w}^T \mathbf{R}(\mathbf{x}^i, \mathbf{bb}^i) + b) + \lambda M(\mathbf{x}^i, \mathbf{bb}^i)). \quad (19)$$

Since $\mathbf{R}(\mathbf{x}^i, \mathbf{bb}^i)$ is concatenation of features per frame, we get:

$$\underset{\mathbf{bb}^i}{\operatorname{argmax}} \big[y^i (\sum_{t=1}^{T} w_t^T \mathbf{r}(\mathbf{x}_t^i, bb_t^i) + b) + \lambda \sum_{t=1}^{T} m(\mathbf{x}_t^i, bb_t^i) \big] \quad (20)$$

$$= \underset{\mathbf{bb}^i}{\operatorname{argmax}} \big\{ \sum_{t=1}^{T} \big[y^i w_t^T \mathbf{r}(\mathbf{x}_t^i, bb_t^i) + \lambda m(\mathbf{x}_t^i, bb_t^i) \big] + y^i b \big\}. \quad (21)$$

We can ignore $y^i b$:

$$= \sum_{t=1}^{T} \underset{bb_t^i}{\operatorname{argmax}} \big[y^i w_t^T \mathbf{r}(\mathbf{x}_t^i, bb_t^i) + \lambda m(\mathbf{x}_t^i, bb_t^i) \big], \quad (22)$$

and, therefore, the inference of an optimal bounding box positions across the whole videos comes down to inference of the optimal bounding box position per frame t:

$$(bb_t^i)^* = \underset{bb_t^i}{\operatorname{argmax}} \big[y^i w_t^T \mathbf{r}(\mathbf{x}_t^i, bb_t^i) + \lambda m(\mathbf{x}_t^i, bb_t^i) \big]. \quad (23)$$

Step 4. Algorithm iteration

After step 3 the algorithm iterates between step 2 and step 3 until it reaches the maximum number of iterations.

As our bounding box search space is constrained to fixed size bounding boxes, the time complexity of one iteration of our algorithm is $O(WHT)$ (for a single training example), where W is width, H is height and T is the number of frames of a video. The complexity of our method is the same as in [38]. In practice we sample a fixed number of points across x-axis, y-axis and frames as described in 3.1. We sample 20 points across the x-axis, 10 points across the y-axis and 5 frames across the video. Therefore, in our case $W = 20$, $H = 10$ and $T = 5$.

4.3. Classification

Once learning is performed, we end up with C binary classifiers that are trained in *one-vs.-all* manner. Each of those classifiers parametrized by (w_c, b_c) can be used in order to determine whether a video described by $\mathbf{x}^i$ depicts the action c by solving the following optimization problem (for clarity, we omit the index i):

$$y_c^*, \mathbf{bb}_c^* = \underset{y \in \{+1,-1\}, \mathbf{bb}}{\operatorname{argmax}} [y f(\mathbf{w}_c, b_c, \mathbf{bb}; \mathbf{x}) + \lambda M(\mathbf{x}, \mathbf{bb})]. \tag{24}$$

In order to solve the multiclass classification problem, we find the label c^* that gives the maximum response $f(\mathbf{w}_c, b_c; \mathbf{x}, \mathbf{bb}_c^*)$ for the video in question. That is, we solve:

$$c^* = \underset{c \in \{1,\ ...,\ C\}}{\operatorname{argmax}} f(\mathbf{w}_c, b_c; \mathbf{x}, \mathbf{bb}_c^*), \tag{25}$$

where $\mathbf{bb}_c^*$ is given by 24.

Finally, let us note that when the bounding boxes are fixed, the classification decisions are not influenced by the saliency costs. That is, the binary classification in eq. 24 becomes a standard SVM classifier, that is:

$$y_c^* = \underset{y \in \{+1,-1\}}{\operatorname{argmax}} [y f(\mathbf{w}_c, b_c; \mathbf{x}, \mathbf{bb}_c^*)], \tag{26}$$

or

$$y_c^* = \operatorname{sgn} f(\mathbf{w}_c, b_c; \mathbf{x}, \mathbf{bb}_c^*). \tag{27}$$

4.4. Comparison with latent SVM presented in [9]

In this section we will show the relation of our model to the latent SVM proposed in [9]. If we set $\lambda = 0$, the cost function of our model takes the following form:

$$\min_{\mathbf{w},b,\{\mathbf{bb}^i\}_{i=1}^M} \frac{1}{2}(\mathbf{w}^T\mathbf{w} + b^2) + \sum_{i=1}^{M} \big[\max(0, 1 - y^i f(\mathbf{w}, b; \mathbf{x}^i, \mathbf{bb}^i))\big], \tag{28}$$

while the one of latent SVM[9] is defined as follows:

$$\min_{\mathbf{w},b} \frac{1}{2}(\mathbf{w}^T\mathbf{w} + b^2) + \sum_{i=1}^{M} \big[\max(0, 1 - y^i \max_{\mathbf{bb}^i} f(\mathbf{w}, b; \mathbf{x}^i, \mathbf{bb}^i))\big]. \tag{29}$$

When searching for $(y^i)^*$ and $(\mathbf{bb}^i)^*$ our model searches over all possible combinations of $(y^i)^*$ and $(\mathbf{bb}^i)^*$ (analogous to eq. 24 when $\lambda = 0$; for clarity, in the rest of the section we omit index i):

$$y^*, \mathbf{bb}^* = \underset{y\in\{+1,-1\},\mathbf{bb}}{\operatorname{argmax}} \; yf(\mathbf{w}, b, \mathbf{bb}; \mathbf{x}). \tag{30}$$

In contrast to this, in [9] the search for y^* and $\mathbf{bb}^*$ is performed in two separate steps:

$$\mathbf{bb}^* = \underset{\mathbf{bb}}{\operatorname{argmax}} \, f(\mathbf{w}, b, \mathbf{bb}; \mathbf{x}), \tag{31}$$

and

$$y^* = \underset{y\in\{+1,-1\}}{\operatorname{argmax}} \; yf(\mathbf{w}, b, \mathbf{bb}^*; \mathbf{x}). \tag{32}$$

The drawback of this model is that it is not possible to incorporate the saliency cost under the $\max_{\mathbf{bb}}$ term as it would add negative saliency in the cost for negative examples. On the other hand, this model searches for the strongest bounding box area response first and classifies it afterwards. By doing so, the models avoids choosing the strong negative response and classifying it as negative, as it can happen in our model. In our model we are trying to avoid this by adding saliency cost.

5. Results

We evaluate our method on the UCF sports dataset [64] using human eye movements recorded for each video of this dataset [7]. It contains 150 videos depicting 10 sports actions classes. The actions are recorded in different scenes and viewpoints, and some of the sequences contain more than one human. The human eye movements were collected from 16 human subjects.

Most works on UCF sports dataset use leave-one-out protocol, however, we follow the one from [38]. In this protocol the dataset is split in 103 training examples and 47 test examples. We follow this protocol for a couple of reasons. First, this is the work closest to ours and second, in [37], where this protocol was introduced, it has been shown that there is a strong scene correlation among videos in certain classes. Also, in some works that use LOO cross validation the parameter setting is unclear [37].

Additionally, to validate the proposed SVM approach we use Olympic sports dataset [56]. For this dataset there are no recorded gaze fixations available, therefore it is not possible to validate our learned features and saliency prediction in the MV framework. However, it is possible and we will show the usefulness of saliency prediction in the SVM framework. The Olympic sports dataset contains videos of athletes practicing different sports. There are 783 videos of 16 sports action classes. Videos are collected from YouTube and only video class annotations are available. We use the suggested split for training and testing available on the dataset webpage.

When predicting saliency on Olympic sports dataset we use 3D CNN for saliency prediction that is trained on UCF sports dataset. As features we use the one from [65], which are also deeply learned using a CNN. However, they are learned on ILSVRC-2012 dataset, which contains only static 2D images, so they do not capture motion. Feature representation is built in the same way as in the experiments on the UCF sports dataset. The only minor difference is that during bounding box search W and H are both set to 7. That is due to the architectural properties of the network used for feature extraction (for more

details see [65]).

5.1. Majority voting based video classification

In Table 2 we present the results obtained by a simple majority voting scheme on the UCF sports dataset in order to illustrate two things. First, in order to show a good discriminatory power of the features learned with CNN we compare the results obtained by simple majority voting scheme to the ones obtained with the BoW approach. In BoW approach cuboids are densely sampled (interestingly, in realistic videos, dense sampling has been shown to be a better sampling strategy than using any kind of STIP[43]) and HOG, HOF and HoMB descriptors are used - for more details see [38]. In our approach the votes of the cuboids are sampled across the whole video in three different ways: without using saliency, i.e. randomly, using saliency predicted by our 3D CNN network trained for saliency prediction or using ground truth saliency. The vote of a cuboid is

$$c^* = \underset{c}{\operatorname{argmax}}\, \mathbf{a}_c(p;\theta_a) = \underset{c}{\operatorname{argmax}}\, P(Y=c|p,\theta_a) \tag{33}$$

where c is an action class. In the setup where ground truth saliency is used, we simply use only the votes of the recorded fixations points. In the setup with predicted saliency we are using the output of a pretrained saliency predictor (3D-CNN) which gives as an output the probability of a cuboid being salient, that is $s(p;\theta_s) = P(Y=+1|p,\theta_s)$. A cuboid is classified as a salient if $s(p;\theta_s) \geq 0.5$ and in that case we count its vote, otherwise we discard it. We can see that our simple majority voting scheme, that is without the additional quantization step and training an SVM classifier that are used in the BoW approach, yields much better results, both when using predicted saliency and when using ground truth saliency. Even when using no saliency, the result is comparable with the global BoW setting. This result shows that our discriminatively learned features compare well to the handcrafted ones.

We can see that even when using ground truth saliency, the result is 85.00%, which is around 3% above the state-of-the-art and around 1.5% above our best

Method	MAP
Global BoW [38]	64.29
BoW with spatial split [38]	65.95
BoW with temporal split [38]	69.64
MV (no saliency)	64.31
MV (with predicted saliency)	74.17
MV (with ground truth saliency)	85.00

Table 2: Results obtained using majority voting scheme on the UCF sports dataset. The measure is mean per class classification accuracy.

result obtained with SVM-based approach. Those results are reported in Table 5 and further discussed in section 5.2. Note that those results reported in Table 5 are obtained by frameworks in which only predicted saliency is allowed. However, one of the reasons that the result of 85.00% obtained using ground truth saliency is still lacking is due to the fact that even ground truth saliency can be misleading - for example, people tend to look a lot at faces, which are not discriminatory for any action. This affects the results in two ways: first, as the action features are used only on fixations, they may capture some irrelevant head movements and because of that, the learned features are lacking. Second, during sampling the ground truth fixations, the same irrelevant movements are captured, which all together can lead to misclassification.

The second thing we want to illustrate in this majority voting scheme is the fact that there is a large improvement in results when using any kind of saliency prediction, either ground truth or predicted, over the results without saliency prediction, i.e. random dense sampling. As mentioned in the beginning of this section, the latter were shown to be superior over sampling the points detected with any of the STIPs[43]. This shows the efficacy of our 3D CNN-based saliency predictor.

In table 3 we present results we obtain by varying the hyperparameters of 3D CNN for action recognition. We varied the number of filters in both

layers, the number of fully connected units and the depth of the 3D CNN for action features. The size of the filters is the same for all networks, that is 4x4x3 in the first layer and 2x2x3 in the second layer. The only exception is network 10, which has larger first layer filters, that is 8x8x5. We can observe the following. First, we can see that adding a layer improves the results. Second, larger number of filters is beneficial (in our case especially first layer filters - compare networks 0 and 1). Third, larger number of units in a fully connected layer leads to overfitting (network 4). Fourth, even in an one layer network training with larger filter size seems to be challenging (network 10). Our results verify general findings in the deep learning literature (we mention those in the beginning of section 3.1). However, we did not investigate the impact that different hyperparameters would have in the SVM framework: in further SVM experiments we use the largest network, that is network 0. The hyperparameters of 3D CNN for saliency prediction had less impact on the action recognition results in a majority voting framework so we omit those.

In Figure 4 we can see some examples of well estimated saliency maps and voting maps. Those maps are obtained by sampling points across the video with a variable step size, as described in section 3.1. For each point, its saliency value (see eq. 6) and voting vector (see eq. 3) are obtained. Correct votes in the voting maps are the ones for which c^* (see eq. 33) corresponds to the ground truth and those are marked with white. The incorrect ones, i.e. the ones that cast vote for any other action than the correct one, are marked with black. In Figure 5 we see examples of misclassification of a video action class, and we notice that the misclassification is mostly due to errors in the voting scheme rather than in the saliency prediction. Those videos exhibit large change in scale and we can see that in those videos it is hard to notice the movements even with bare eye.

5.2. SVM-based video classification

In this section we present the results obtained with the method presented in section 4. As a baseline, in Table 5 we report the results obtained without

Network	1st layer filters	2nd layer filters	fully connected units	result
0	50	100	50	85.11
1	50	50	50	85.11
2	25	100	50	76.60
3	50	100	25	78.72
4	50	100	100	78.72
5	25	50	50	78.72
6	50	50	25	78.72
7	50	-	50	74.47
8	50	-	25	72.34
9	10	-	50	68.09
10	50	-	50	68.09

Table 3: Results obtained in a majority voting framework when neural network for action features is learned using different hyperparamters. Here, as saliency prediction ground truth fixations are used. The measure is classification accuracy per video, as opposed to the mean per class classification accuracy reported in table 2.

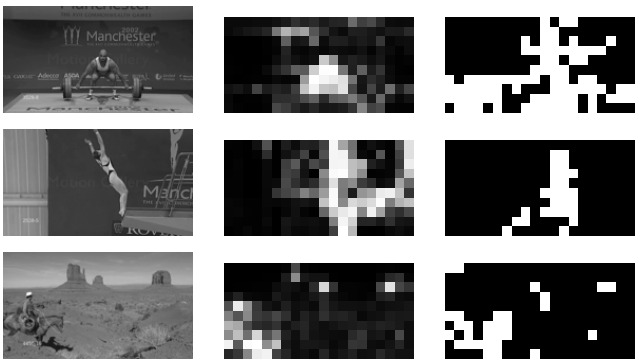

Figure 4: Examples of good saliency prediction and voting maps: (a) original frame, (b) predicted saliency map, (c) voting map.

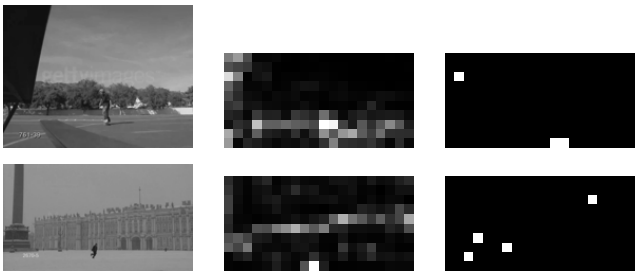

Figure 5: Examples of bad voting maps: (a) original frame, (b) predicted saliency map, (c) voting map.

optimizing with respect to the bounding box, i.e. when the bounding box at each frame is the whole frame. Those results are significantly lower than the ones obtained when using bounding boxes, except when $\lambda = 0$. This illustrates the importance of both searching for discriminative areas in the video and the importance of adding the saliency cost.

Furthermore, we would like to illustrate the importance of adding the saliency cost term in our SVM framework by varying the value of the parameter λ in the set of values: $\{0.0, 2.5.5.0, 7.5, 10.0, 15.0, 17.5, 20.0\}$. The results obtained on the test set of UCF sports dataset are presented in Fig. 6. We observe that increasing λ significantly improves the results and that the peak is reached at $\lambda = 2.5$. For λ>17.5 the performance drops and then, as $\lambda \to +\infty$ it saturates. The highest performance on the test set was obtained for $\lambda = 2.5$ (85.24%), however with cross validation we obtained $\lambda = 5.0$, so in Table 5 we report the results obtained with that value of λ (83.57%). The value of the SVM parameter C in Eq. (13) was also obtained by cross validation; $C = 0.1$. Parameters λ and C were obtained by cross validation simultaneously and the cross validation was 2-folded. Note that when using no saliency, i.e. for $\lambda = 0.0$, the results are worse than the ones obtained with majority voting using saliency prediction (see Table 2). In this case the initial weights were obtained using saliency information, so this also illustrates the importance of saliency being incorporated in the optimization procedure.

	d	g	k	l	ri	ru	s	sB	sSA	w
diving	100	0	0	0	0	0	0	0	0	0
golf	0	83.3	0	0	0	0	0	0	0	16.7
kicking	16.7	0	66.7	0	0	0	0	0	0	16.7
lifting	0	0	0	100	0	0	0	0	0	0
riding	0	0	0	0	100	0	0	0	0	0
running	0	0	0	0	0	75	0	0	0	0
skateboarding	0	0	0	0	0	0	25	0	0	75
swing-bench	0	0	0	0	0	0	0	100	0	0
swing-SA	0	0	0	0	0	0	0	0	100	0
walking	0	14.3	0	0	0	0	0	0	0	85.7

Table 4: Confusion matrix obtained for $\lambda = 5.0$ on the UCF sports dataset

For the Olympic sports dataset we perform cross validation only over C parameter and use $\lambda = 5.0$ obtained for UCF sports dataset. The results we obtain on this dataset also show that it is beneficial to use saliency, even if it is learned on a different dataset. We have seen that on the UCF sports best results are achieved for smaller λ. Therefore, it is interesting to note that on the Olympic sports dataset better results are achieved when $\lambda = +\infty$, even though the saliency is learned on the UCF sports. This is probably due to the fact that in the experiments on UCF sports we use features trained on this dataset, and in the experiments on the Olympic sports we use features that are not trained on the Olympic sports dataset.However, it also implies that our learned saliency is general enough to improve the result on a different dataset comparing to the method that does not use saliency.

In the same table, that is Table 5, we compare our results to the state-of-the-art. The work that is closest to ours is presented in [38]. In their work the recored human gaze fixations are incorporated, in the training phase only, in the form of structural loss of a structured output latent SVM that is used as an action classifier. That makes the gaze inference necessary during the opti-

mization. By contrast, in our method the saliency prediction depends only on the output of the pretrained 3D CNN, i.e. there is no top-down inference. The works of [37, 39] also use latent SVM, however no saliency data has been used in either training or testing. [66] and [18] also do not use saliency data. Another major difference in comparison to [38] is that as a feature representation they use BoW per frame. By contrast, we use feature concatenation, which seems to be a better representation, as inside the discriminative area of a bounding box the spatial relations should not be disregarded. Furthermore, [38] reports the results obtained when inferring one and two discriminative regions to illustrate the importance of adding flexibility in the choice of discriminative regions. We can see that even when using two discriminative regions (for which they obtain best results), their results were worse than ours even though we use only a single bounding box. In the confusion matrix presented in Table 4 we can see that the action kicking which contains additional object of interest (the ball) has the lowest accuracy.

The state-of-the-art on Olympic sports dataset is obtained using trajectory based features and Fisher Vectors as a higher level video representation [17]. [1] As we mention in the beginning of the related work section, those are the methods that in general currently hold the state-of-the-art on the action recognition datasets. We can see that our results are lower. We intent to investigate how such representations can be incorporated in our framework in our future work.

The main limitation of our approach is that our representation is not invariant to larger translation and scale variations: the choice of fixed size bounding box makes our representation sensitive to scale changes, and using the fixed feature concatenation without any kind of pooling makes it sensitive to translations. That is a problem especially for periodic actions, such as skateboarding and running - in Table 4 we can see that accuracies obtained for those actions are lower.

[1] Note that the results of the approach of [17] we report are reported in [67], not in [17].

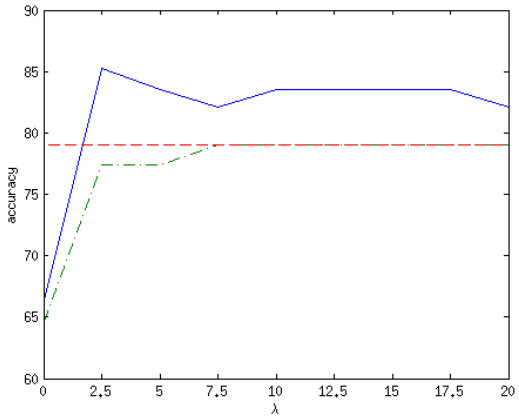

Figure 6: The effect of different λ values on the UCF sports test set. Dash-dot green line represents the results obtained before the optimization, i.e. using the initial weights and different values of λ, full blue line represents the results after the optimization, and red dashed line represents the result obtained using the initial weights and the initial bounding boxes.

5.2.1. Comparison with latent SVM results

In section 4.4 we showed how our model in a special case when $\lambda = 0$ compares to the one of [9]. Here, in Table 5 we report how their results compare. As we are interested in comparing only the performance of the latent SVM model [9] to ours, the features that are used are the ones that we used in all our experiments, i.e. the ones obtained by 3D CNN. That is, we do not implement the deformable parts model on top of which latent SVM is built, as presented in [9].

We can see that the model of [9] achieves slightly better results than our model in case of $\lambda = 0$. This was expected as our model can pick up on the noise of strong negative bounding box responses, while [9] searches for the strong bounding box area response first and classifies it afterwards. However, with added saliency cost, our model outperforms it by a large margin. Hence, it seems that adding saliency cost acts as a much better regularization scheme for not picking the irrelevant background clutter.

Method	UCF sports	Olympic sports
Lan *et al.* [37]	73.1	-
Shapovalova *et al.* [39]	75.3	-
Raptis *et al.* [18]	79.4	-
Jain *et al.* [66]	80.24	-
Shapovalova *et al.* [38] (1 region)	77.98	-
Shapovalova *et al.* [38] (2 regions)	82.14	-
Niebles *et al.* [56]	-	72.1
Laptev *et al.* [68]	-	62.0
Peng *et al.* [17]	-	93.8
Ours (no bounding box)	67.62	60.45
Ours ($\lambda = +\infty$)	79.05	67.16
Ours ($\lambda = 0.0$)	66.31	59.7
Ours ($\lambda = 5.0$)	83.57	64.18
Latent SVM	68.57	61.19

Table 5: Comparison of the results obtained with SVM to the state-of-the-art. The measure for UCF sports dataset is mean per class classification accuracy. The measure for Olympic sports dataset is mean average precision.

6. Conclusion

In this paper we have shown how saliency prediction learned from recorded human fixations can alleviate the problem of action recognition: first, by training a saliency predictor, and second, by training a discriminative mid level feature extractor on recorded human fixations. We have shown the efficacy of both the saliency predictor and the feature extractor in a simple majority voting framework. Furthermore, we have developed an SVM framework which incorporates the saliency cost from saliency prediction and representation built from learned action features. In this framework we have shown the importance of using saliency through saliency cost term and achieved state-of-the-art results on the UCF sports dataset.

References

[1] I. Laptev, T. Lindeberg, Space-time interest points, in: Proceedings of International Conference on Computer Vision, 2003.

[2] A. Klami, C. Saunders, T. Campos, S. Kaski, Can relevance of images be inferred from eye movements?, in: Proceedings of International Conference on Multimedia Information Retrieval, 2008.

[3] Y. Zhang, H. Fu, Z. Liang, Z. Chi, D. Feng, Eye movement as an interaction mechanism for relevance feedback in a content-based image retrieval system, in: Proceedings of the Eye Tracking Research and Application Symposium, 2010.

[4] S. Vrochidis, I. Patras, I. Kompatsiaris, Exploiting gaze movements for automatic video annotation, in: Proceedings of International Workshop on Image Analysis for Multimedia Interactive Services, 2012.

[5] M. Sadeghi, G. Tien, G. Hamarneh, M. Atkins, Hands-free interactive image segmentation using eyegaze, in: Proceedings of SPIE 7260, Medical Imaging 2009: Computer-Aided Diagnosis, 2009.

[6] A. Fathi, Y. Li, J. M. Rehg, Learning to recognize daily actions using gaze, in: Proceedings of European Conference on Computer Vision, 2012.

[7] S. Mathe, C. Sminchisescu, Dynamic eye movement datasets and learnt saliency models for visual action recognition, in: Proceedings of European Conference on Computer Vision, 2012.

[8] E. Vig, M. Dorr, D. Cox, Space-variant descriptor sampling for action recognition based on saliency and eye movements, in: Proceedings of European Conference on Computer Vision, 2012.

[9] P. Felzenschwalb, R. Girshick, D. McAllester, D. Ramannan, Object detection with discriminatively trained part based models, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (9) (2009) 1627–1645.

[10] J. Aggarwal, M. S. Ryoo, Human activity analysis: A review, ACM Computing Surveys 43.

[11] R. Poppe, A survey on vision-based human action recognition, Image Vision Computing 28 (2010) 976–990.

[12] D. Weinland, R. Ronfard, E. Boyer, A survey of vision-based methods for action representation, segmentation and recognition, Computer Vision and Image Understanding 115 (2011) 224–241.

[13] H. Wang, A.Klaser, C.Schmid, C. Liu, Action recognition by dense trajectories, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, 2011.

[14] H. Wang, A.Klaser, C.Schmid, C. Liu, Action recognition with improved trajectories, in: Proceedings of International Conference on Computer Vision, 2013.

[15] J. Sun, X. Wu, S. Yan, L. Cheong, T. Chua, J. Li, Hierarchical spatio-temporal context modeling for action recognition, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, 2009.

[16] X. Peng, L. Wang, X. Wang, Y. Qiao, Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice, in: arXiv preprint arXiv:1405.4506, 2014.

[17] X. Peng, C. Zou, Y. Qiao, Q. Peng, Action recognition with stacked fisher vectors, in: Proceedings of European Conference on Computer Vision, 2014.

[18] M. Raptis, I. Kokkinos, S. Soatto, Discovering disriminative action parts from mid-level video representation, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, 2012.

[19] F. Perronnin, J. Sanchez, T. Mensink, Improving the fisher kernel for large-scale image classification, in: Proceedings of European Conference on Computer Vision, 2010.

[20] K. Chatfield, V. Lempitsky, A. Vedaldi, A. Zisserman, The devil is in the details: an evaluation of recent feature encoding methods, in: Proceedings of British Machine Vision Conference, 2011.

[21] D. Oneata, J. Verbeek, C. Schmid, Action and event recognition with fisher vectors on a compact feature set, in: Proceedings of International Conference on Computer Vision, 2013.

[22] H. Wang, D. Oneata, J. Verbeek, C. Schmid, A robust and efficient video representation for action recognition, in: arXiv preprint arXiv:1504.05524, 2015.

[23] A. Quattoni, S. Wang, L. Morency, M. Collins, T. Darrell, Hidden-state conditional random fields, IEEE TPAMI.

[24] C. Sminchisescu, A. Kanaujia, M. D., Conditional models for contextual human motion recognition, CVIU.

[25] Y. Song, L. Morency, R. Davis, Action recognition by hierarchical sequence summarization, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, 2013.

[26] Y. Wang, , M. G., Hidden part models for human action recognition: Probabilistic versus max margin, IEEE TPAMI.

[27] D. Q. P. T. V. Duong, H. H. Bui, S. Venkatesh, Activity recognition and abnormality detection with the switching hidden semi-markov model, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, 2005.

[28] S. Hongeng, R. Nevatia, Large-scale event detection using semi-hidden markov models, in: Proceedings of International Conference on Computer Vision, 2003.

[29] P. Natarajan, R. Nevatia, Coupled hidden semi markov models for activity recognition, in: Proceedings of IEEE workshop on Motion and Video Computing (WMVC), 2007.

[30] K. Tang, L. Fei-Fei, D. Koller, Learning latent temporal structure for complex event detection, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, 2012.

[31] J. C. Niebles, H. Wang, L. Fei-Fei, Unsupervised learning of human action categories using spatial-temporal words, Internation Journal of Computer Vision 79 (3) (2008) 299–318.

[32] J. K. Aggarwal, M. S. Ryoo, Recognition of composite human activities through context-free grammar based representation, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, 2006.

[33] H. Pirsiavash, D. Ramanan, Parsing videos of actions with segmental grammars, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, 2014.

[34] H. Kuehne, A. Arslan, T. Serre, The language of actions: Recovering the syntax and semantics of goal-directed human activities, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, 2014.

[35] N. N. Vo, A. F. Bobick, From stochastic grammar to bayes network: Probabilistic parsing of complex activity, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, 2014.

[36] S. Assari, A. Zamir, M. Shah, Video classification using semantic concept co-occurrences, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, 2014.

[37] T. Lan, Y. Wnag, G. Mori, Discriminative figure-centric models for joint action localization and recognition, in: Proceedings of International Conference on Computer Vision, 2011.

[38] N. Shapovalova, M. Raptis, L. Sigal, G. Mori, Action is in the eye of the beholder: Eye-gaze driven model for spatio-temporal action localization, in: Proceedings of Neural Information Processing Systems Conference, 2013.

[39] N. Shapovalova, A. Vahdat, K. Cannons, T. Lan, G. Mori, Similarity constrained latent support vector machine: An application to weakly supervised action classification, in: Proceedings of European Conference on Computer Vision, 2012.

[40] A. Gilbert, J. Illingworth, R. Bowden, Action recognition using mined hierarchical compound features, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (5) (2011) 883–897.

[41] Q. V. Le, W. Y. Zou, S. Y. Yeung, A. Y. Ng, Learning hierarchihal invariant spatio-temporal features for action recognition with independent subspace analysis, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, 2011.

[42] A. Kovashka, K. Grauman, Learning a hierarchy of discriminative space-time neighborhood features for human action recognition, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, 2010.

[43] H. Wang, U. M. M., A. Klaser, I. Laptev, C. Schmid, Evaluation of local spatio-temporal features for action recognition, in: Proceedings of British Machine Vision Conference, 2009.

[44] S. Christian, I. Laptev, B. Caputo, Recognizing human actions: A local svm approach, in: Proceedings of International Conference on Pattern Recognition, 2004.

[45] C. F. T. P. H. Sapienza, M., Learning discriminative space-time actions from weakly labelled videos, in: Proceedings of British Machine Vision Conference, 2012.

[46] J. Yuan, L. Zicheng, Y. Wu, Discriminative subvolume search for efficient action detection, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, 2009.

[47] S. Koelstra, I. Patras, The fast-3d spatio-temporal interest region detector, in: Workshop on Image Analysis for Multimedia Interactive Services, 2009.

[48] A. Krizhevsky, I. Sutskever, G. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of Neural Information Processing Systems Conference, 2012.

[49] H. Lee, R. Grosse, , R. Ranganath, A. Ng, Convolutional deep belief networks for for scalable unsupervised learning of hiererchical representations, in: Proceedings of International Conference on Machine Learning, 2009.

[50] C. Szegedy, A. Toshev, D. Erhan, Deep neural networks for object detection, in: Proceedings of Neural Information Processing Systems Conference, 2013.

[51] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, 2014.

[52] L. Sun, K. Jia, T. Chan, Y. Fang, G. Wang, S. Yan, Dl-sfa: Deeply-learned slow feature analysis for action recognition, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, 2014.

[53] S. Ji, W. Xu, M. Yang, K. Yu, 3d convolutional neural networks for human action recognition, IEEE Transactions on Pattern Recognition and Machine Intelligence 35 (1) (2013) 221–231.

[54] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, A. Baskurt, Sequential deep learning for human action recognition, in: Proceedings of Workshop on Human Behaviour Understanding, 2011.

[55] D. Tran, L. Torresani, Exmoves: Classifier-based features for scalable action recognition, in: Proceedings of International Conference on Learning Representations, 2014.

[56] J. Niebles, C. Chen, L. Fei-Fei, Modeling temporal structure of decomposable motion segments of activity classification, in: Proceedings of European Conference on Computer Vision, 2010.

[57] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, Y. Bengio, Theano: new features and speed improvements, Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop (2012).

[58] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, Y. Bengio, Theano: a CPU and GPU math expression compiler, in: Proceedings of the Python for Scientific Computing Conference (SciPy), 2010.

[59] Y. Lin, S. Kong, D. Wang, Y. Zhuang, Saliency detection within a deep convolutional architecture, in: Proceedings of AAAI Workshop on Cognitive Computing for Augmented Human Intelligence, 2014.

[60] E. Eleonora Vig, M. Dorr, D. Cox, Large-scale optimization of hierarchical features for saliency prediction in natural images, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, 2014.

[61] M. Kmmerer, L. Theis, M. Bethge, Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet, in: arXiv:1411.1045, 2014.

[62] C. Shen, M. Song, Q. Zhao, Learning high-level concepts by training a deep network on eye fixations, in: Deep Learning and Unsupervised Feature Learning Workshop, in conjunction with NIPS, 2012.

[63] C. Shen, Q. Zhao, Learning to predict eye fixations for semantic contents using multi-layer sparse network, Neurocomputing 138 (2014) 61–68.

[64] M. Rodriguez, J. Ahmed, M. Shah, Action mach: A spatio-temporal maximum average correlation height filter for action recognition, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, 2008.

[65] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proceedings of International Conference on Learning Representations, 2015.

[66] M. Jain, J. Gemert, H. Jegou, P. Bouthemy, C. Snoek, Action localization with tubelets from motion, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, 2014.

[67] Z. Lan, M. Lin, X. Li, A. Hauptmann, B. Raj, Beyond gaussian pyramid: Multi-skip feature stacking for action recognition, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, 2015.

[68] I. Laptev, M. Marszalek, C. Schmid, B. Rozengeld, Learning realistic human actions from movies, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, 2008.