Finding Action Tubes

Georgia Gkioxari
UC Berkeley
gkioxari@eecs.berkeley.edu

Jitendra Malik
UC Berkeley
malik@eecs.berkeley.edu

Abstract

We address the problem of action detection in videos. Driven by the latest progress in object detection from 2D images, we build action models using rich feature hierarchies derived from shape and kinematic cues. We incorporate appearance and motion in two ways. First, starting from image region proposals we select those that are motion salient and thus are more likely to contain the action. This leads to a significant reduction in the number of regions being processed and allows for faster computations. Second, we extract spatio-temporal feature representations to build strong classifiers using Convolutional Neural Networks. We link our predictions to produce detections consistent in time, which we call action tubes. We show that our approach outperforms other techniques in the task of action detection.

1. Introduction

In object recognition, there are two traditional problems: whole image classification, "*is there a chair in the image?*", and object detection, "*is there a chair and where is it in the image?*". The two problems have been quantified by the PASCAL Visual Object Challenge [11, 10] and more recently the ImageNet Challenge [8, 7]. The focus has been on the object detection task due to its direct relationship to practical, real world applications. When we turn to the field of action recognition in videos, we find that most work is focused on video classification,"*is there an action present in the video*", with leading approaches [40, 41, 35] trying to classify the video as a whole. In this work, we address the problem of action detection, "*is there an action and where is it in the video*".

Our goal is to build models which can localize and classify actions in video. Figure 1 outlines our approach. Inspired by the recent advances in the field of object detection in images [13], we start by selecting candidate regions and use convolutional networks (CNNs) to classify them. Motion is a valuable cue for action recognition and we utilize it in two ways. We use motion saliency to eliminate regions that are not likely to contain the action. This leads to a big reduction in the number of regions being processed and subsequently in compute time. Additionally, we incorporate kinematic cues to build powerful models for action detection. Figure 2 shows the design of our action models. Given a region, appearance and motion cues are used with the aid of convolutional neural networks to make a prediction. Our experiments indicate that appearance and motion are complementary sources of information and using both leads to significant improvement in performance (Section 4). Predictions from all the frames of the video are linked to produce consistent detections in time. We call the linked predictions in time *action tubes*.

Our detection pipeline is inspired by the human vision system and, in particular, the two-streams hypothesis [14]. The ventral pathway ("*what pathway*") in the visual cortex responds to shape, color and texture while the dorsal pathway ("*where pathway*") responds to spatial transformations and movement. We use convolutional neural networks to computationally simulate the two pathways. The first network, *spatial-CNN*, operates on static cues and captures the appearance of the actor and the environment. The second network, *motion-CNN*, operates on motion cues and captures patterns of movement of the actor and the object (if any) involved in the action. Both networks are trained to discriminate between the actors and the background as well as between actors performing different actions.

We show results on the task of action detection on two publicly available datasets, that contain actions in real world scenarios, UCF Sports [33] and J-HMDB [17]. These are the only datasets suitable for this task, unlike the task of action classification, where more datasets and of bigger size (up to 1M videos) exist. Our approach outperforms all other approaches ([15, 42, 38, 25]) on UCF sports, with the biggest gain observed for high overlap thresholds. In particular, for an overlap threshold of 0.6 our approach shows a relative improvement of 87.3%, achieving mean AUC of 41.2% compared to 22.0% reported by [42]. On the larger J-HMDB, we present an ablation study and show the effect of each component when considered separately. Unfortunately, no other approaches report numbers on this dataset.

Additionally, we show that action tubes yield improved results on action classification on J-HMDB. Using our action detections we are able to achieve an accuracy of 62.5% on J-HMDB, compared to 56.6% reported by [40] and 56.5% achieved by a whole frame video classification technique with CNNs.

The rest of the paper is organized as follows. In Section 2 we mention related work on action classification and action detection in videos. In Section 3 we describe the details of our approach. In Section 4 we report our results on the two datasets.

2. Related Work

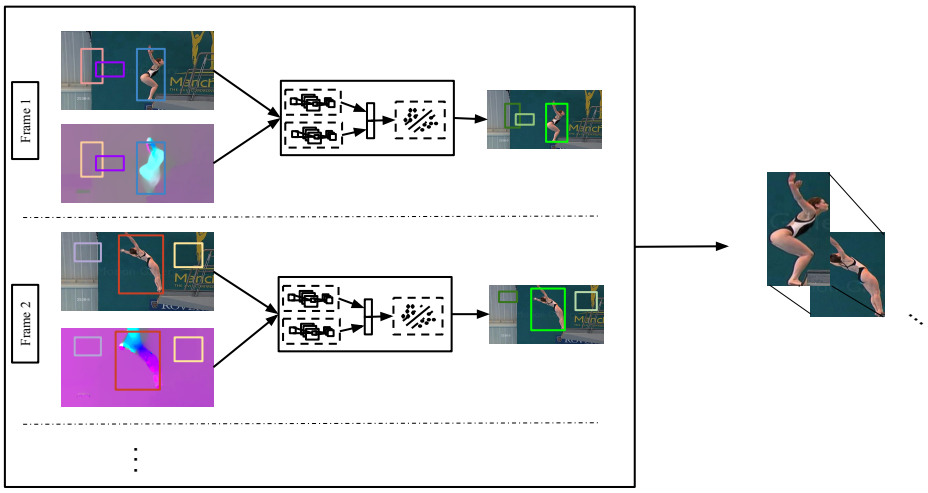
There has been a fair amount of research on action recognition. We refer to [1, 30, 43] for recent surveys in the field. For the task of action classification, recent approaches use features based on shape (e.g. HOG [5], SIFT [28]) and motion (e.g. optical flow, MBH [6]) with high order encodings (e.g. Bag of Words, Fischer vectors) and train classifiers (e.g. SVM, decision forests) to make action predictions. More specifically, Laptev *et al*. [26] extract local features at spatio-temporal interest points which they encode using Bag of Words and train SVM classifiers. Wang *et al*. [40] use dense point trajectories, where features are extracted from regions which are being tracked using optical flow across the frames, instead of fixed locations on a grid space. Recently, the authors improved their approach [41] using camera motion to correct the trajectories. They estimate the camera movement by matching points between frames using shape and motion cues after discarding those that belong to the humans in the frame. The big relative improvement of their approach shows that camera motion has a significant impact on the final predictions, especially when dealing with real world video data. Jain *et al*. [16] make a similar observation.

Following the impressive results of deep architectures, such as CNNs, on the task of handwritten digit recognition [27] and more recently image classification [23] and object detection in images [13], attempts have been made to train deep networks for the task of action classification. Jhuang *et al*. [18] build a feedforward network which consists of a hierarchy of spatio-temporal feature detectors of predesigned motion and shape filters, inspired by the dorsal stream of the visual cortex. Taylor *et al*. [37] use convolutional gated RBMs to learn features for video data in an unsupervised manner and apply them for the task of action classification. More recently, Ji *et al*. [19] build 3D CNNs, where convolutions are performed in 3D feature maps from both spatial and temporal dimensions. Karpathy *et al*. [21] explore a variety of network architectures to tackle the task of action classification on 1M videos. They show that operating on single frames performs equally well than when considering sequences of frames. Simonyan & Zisserman [35] train two separate CNNs to explicitly capture spatial and temporal features. The spatial stream operates on the RGB image while the temporal stream on the optical flow signal. The two stream structure in our network for action detection is similar to their work, but the crucial difference is that their network is for full image classification while our system works on candidate regions and can thus localize the action. Also, the way we do temporal integration is quite different since our work tackles a different problem.

Approaches designed for the task of action classification use feature representations that discard any information regarding the location of the action. However, there are older approaches which are figure centric. Efros *et al*. [9] combine shape and motion features to build detectors suitable for action recognition at low resolution and predict the action using nearest neighbor techniques, but they assume that the actor has already been localized. Schüldt *et al*. [34] build local space-time features to recognize action patters using SVM classifiers. Blank *et al*. [3] use spatio-temporal volume silhouettes to describe an action assuming in addition known background. More recently, per-frame human detectors have been used. Prest *et al*. [31] propose to detect humans and objects and then model their interaction. Lan *et al*. [25] learn spatio-temporal models for actions using figure-centric visual word representation, where the location of the subject is treated as a latent variable and is inferred jointly with the action label. Raptis *et al*. [32] extract clusters of trajectories and group them to predict an action class using a graphical model. Tian *et al*. [38] extend the deformable parts model, introduced by [12] for object detection in 2D images, to video using HOG3D feature descriptors [22]. Ma *et al*. extract segments of the human body and its parts based on color cues, which they prune using motion and shape cues. These parts serve as regions of interest from which features are extracted and subsequently are encoded using Bag of Words. Jain *et al*. [15] produce space-time bounding boxes, starting from super-voxels, and use motion features with Bag of Words to classify the action within each candidate. Wang *et al*. [42] propose a unified approach to discover effective action parts using dynamical poselets and model their relations.

3. Building action detection models

Figure 1 outlines our approach. We classify region proposals using static and kinematic cues (stage a). The classifiers are comprised of two Convolutional Neural Networks (CNNs) which operate on the RGB and flow signal respectively. We make a prediction after using action specific SVM classifiers trained on the spatio-temporal representations produced by the two CNNs. We link the outputs of the classifiers across the frames of the videos (stage b) to produce *action tubes*.



(a) Action detection at every frame of the video

(b) Linked detections in time produce *action tubes*

Figure 1: An outline of our approach. (a) Candidate regions are fed into action specific classifiers, which make predictions using static and motion cues. (b) The regions are linked across frames based on the action predictions and their spatial overlap. *Action tubes* are produced for each action and each video.

3.1. Regions of interest

Given a frame, the number of possible regions that contain the action is enormous. However, the majority of these candidates are not descriptive and can be eliminated without loss in performance. There has been a lot of work on generating useful region proposals based on color, texture, edge cues ([39, 2]). We use selective search [39] on the RGB frames to generate approximately 2K regions per frame. Given that our task is to localize the actor, we discard the regions that are void of motion, using the optical flow signal. As a result, the final regions we consider are those that are salient in shape and motion. One could use more complicated techniques, such as action saliency detectors trained on human eye fixations and low level cues [29].

Our motion saliency algorithm is extremely simple. We view the normalized magnitude of the optical flow signal f_m as a heat map at the pixel level. If R is a region, then $f_m(R) = \frac{1}{|R|}\sum_{i \in R} f_m(i)$ is a measure of how motion salient R is. R is discarded if $f_m(R) < \alpha$.

For $\alpha = 0.3$, approximately 85% of boxes are discarded, with a loss of only 4% in recall on J-HMDB, for an overlap threshold of 0.5. Despite the small loss in recall, this step is of great importance for the algorithm's time complexity. It takes approximately 11s to process an image with 2K boxes, with the majority of the time being consumed in extracting features for the boxes (for more details see [13]). This means that a video of 100 frames would require 18min to process! This is prohibitive, especially for a dataset of thousands of videos. Eliminating regions which are unlikely to contain the action reduces the compute time significantly.

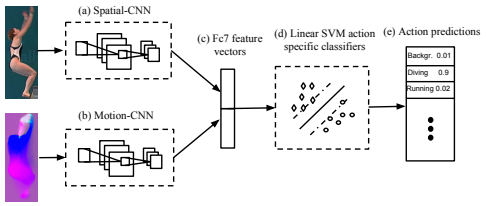


Figure 2: We use action specific SVM classifiers on spatio-temporal features. The features are extracted from the fc7 layer of two CNNs, *spatial-CNN* and *motion-CNN*, which were trained to detect actions using static and motion cues, respectively.

3.2. Action specific classifiers

We use discriminative action classifiers on spatio-temporal features to make predictions for each region. The features are extracted from the final layer of the CNNs which are trained to discriminate among different actions as well as between actions and the background. We use a linear SVM with hard negative mining to train the final classifiers. Figure 2 shows how spatial and motion cues are combined and fed into the SVM classifier.

3.2.1 CNNs for action detection

We train two Convolutional Neural Networks for the task of action detection. The first network, *spatial-CNN*, takes as

input RGB frames and captures the appearance of the actor as well as cues from the scene. The second network, *motion-CNN*, operates on the optical flow signal and captures the movement of the actor. Spatio-temporal features are extracted by combining the output from the intermediate layers of the two networks. Action specific SVM classifiers are trained on the spatio-temporal features and are used to make predictions at the frame level. Figure 2 schematically outlines the procedure. Subsequently, we link the detections in time to produce temporarily consistent action predictions, which we call *action tubes*.

We train spatial-CNN and motion-CNN similar to R-CNN [13]. Regions of interest are computed at every frame of the video, as described above. At train time, the regions which overlap more than 50% with the ground truth are considered as positive examples, and the rest are negatives. The networks are carefully initialized to avoid overfitting.

The architecture of spatial-CNN and motion-CNN is identical and follows [23] and [44]. Assume $C(k, n, s)$ is a convolutional layer with kernel size $k \times k$, n filters and a stride of s, $P(k, s)$ a max pooling layer of kernel size $k \times k$ and stride s, N a normalization layer, RL a rectified linear unit, $FC(n)$ a fully connected layer with n filters and $D(r)$ a dropout layer with dropout ratio r. The architecture of our networks follows: $C(7, 96, 2) - RL - P(3, 2) - N - C(5, 384, 2) - RL - P(3, 2) - N - C(3, 512, 1) - RL - C(3, 512, 1) - RL - C(3, 384, 1) - RL - P(3, 2) - FC(4096) - D(0.5) - FC(4096) - D(0.5) - FC(|A|+1)$. The final fully connected layer has number of outputs as many as the action classes plus one for the background class. During training a softmax loss layer is added at the end of the network.

Network details The architecture of our CNNs is inspired by two different network designs, [23] and [44]. Our network achieves 17% top-5 error on the ILSVRC-2012 validation set for the task of classification.

Weight initialization Proper initialization is a key for training CNNs, especially in the absence of data.
spatial-CNN: We want spatial-CNN to accurately localize people performing actions in 2D frames. We initialize spatial-CNN with a model that was trained on the PASCAL VOC 2012 detection task, similar to [13]. This model has learned feature representations necessary for accurately detecting people under various appearance and occlusion patterns, as proven by the high person detection AP reported on the VOC2012 test set.
motion-CNN: We want motion-CNN to capture motion patterns. We train a network on single frame optical flow images for the task of action classification. We use the UCF101 dataset (split 1) [36], which contains 13320 videos of 101 different actions. Our single frame optical flow model achieves an accuracy of 72.2% on split 1, similar to 73.9% reported by [35]. The 1.7% difference can be attributed to the differences in the network's architectures. Indeed, the network used in [35] yields 13.5% top-5 error on the ILSVRC-2012 validation set, compared to the 17% top-5 error achieved by our network. This model is used to initialize motion-CNN when trained on smaller datasets, such as UCF Sports and J-HMDB.

Processing of input data We preprocess the input for each of the networks as follows
spatial-CNN: The RGB frames are cropped to the bounds of the regions of interest, with a padding of 16 pixels, which is added in each dimension. The average RGB values are subtracted from the patches. During training, the patches are randomly cropped to 227×227 size, and are flipped horizontally with a probability of 0.5.
motion-CNN: We compute the optical flow signal for each frame, according to [4]. We stack the flow in the x-, y-direction and the magnitude to form a 3-dimensional image, and scale it by a constant ($s = 16$). During training, the patches are randomly cropped and flipped.

Parameters We train spatial-CNN and motion-CNN with backpropagation, using Caffe [20]. We use a learning rate of 0.001, a momentum of 0.9 and a weight decay of 0.0005. We train the networks for 2K iterations. We observed more iterations were unnecessary, due to the good initialization of the networks.

3.2.2 Training action specific SVM classifiers

We train action specific SVM classifiers on spatio-temporal features, which are extracted from an intermediate layer of the two networks. More precisely, given a region R, let $\phi_s(R)$ and $\phi_m(R)$ be the feature vectors computed after the 7th fully connected layer in spatial-CNN and motion-CNN respectively. We combine the two feature vectors $\phi(R) = [\phi_s(R)^T \ \phi_m(R)^T]^T$ to obtain a spatio-temporal feature representation for R. We train SVM classifiers $\mathbf{w}_\alpha$ for each action $\alpha \in A$, where ground truth regions for α are considered as positive examples and regions that overlap less than 0.3 with the ground truth as negative. During training, we use hard negative mining.

At test time, each region R is a associated with a score vector $score(R) = \{\mathbf{w}_\alpha^T \phi(R) : \alpha \in A\}$, where each entry is a measure of confidence that action α is performed within the region.

3.3. Linking action detections

Actions in videos are being performed over a period of time. Our approach makes decisions on a single frame level. In order to create temporally coherent detections, we link the results from our single frame approach into unified detections along time.

Assume two consecutive frames at times t and $t+1$, respectively, and assume R_t is a region at t and R_{t+1} at $t+1$. For an action α, we define the linking score between those regions to be

$$s_\alpha(R_t, R_{t+1}) = \mathbf{w}_\alpha^T \phi(R_t) + \mathbf{w}_\alpha^T \phi(R_{t+1}) + \lambda \cdot ov(R_t, R_{t+1}) \quad (1)$$

where $ov(R, \hat{R})$ is the intersection-over-union of two regions R and $\hat{R}$ and λ is a scalar. In other words, two regions are strongly linked if their spatial extent significantly overlaps and if they score high under the action model.

For each action in the video, we seek the optimal path

$$\bar{R}_\alpha^* = \underset{\bar{R}}{\operatorname{argmax}} \frac{1}{T} \sum_{t=1}^{T-1} s_\alpha(R_t, R_{t+1}) \quad (2)$$

where $\bar{R}_\alpha = [R_1, R_2, ..., R_T]$ is the sequence of linked regions for action α. We solve the above optimization problem using the Viterbi algorithm. After the optimal path is found, the regions in $\bar{R}_\alpha^*$ are removed from the set of regions and Eq. 2 is solved again. This is repeated until the set of regions is empty. Each path from Eq. 2 is called an *action tube*. The score of an action tube $\bar{R}_\alpha$ is defined as $S_\alpha(\bar{R}_\alpha) = \frac{1}{T} \sum_{t=1}^{T-1} s_\alpha(R_t, R_{t+1})$.

4. Results

We evaluate our approach on two widely used datasets, namely UCF Sports [33] and J-HMDB [17]. On UCF sports we compare against other techniques and show substantial improvement from state-of-the-art approaches. We present an ablation study of our CNN-based approach and show results on action classification using our action tubes on J-HMDB, which is a substantially larger dataset than UCF Sports.

Datasets UCF Sports consists of 150 videos with 10 different actions. There are on average 10.3 videos per action for training, and 4.7 for testing [1]. J-HMDB contains about 900 videos of 21 different actions. The videos are extracted from the larger HMDB dataset [24], consisting of 51 actions. Contrary to J-HMDB, UCF Sports has been widely used by scientists for evaluation purposes. J-HMDB is more interesting and should receive much more attention than it has in the past.

Metrics. To quantify our results, we report Average-Precision at a frame level, *frame-AP*, and at the video level, *video-AP*. We also plot ROC curves and measure AUC, a metric commonly used by other approaches. None of the AP metrics have been used by other methods on this task. However, we feel they are informative and provide a direct link between the tasks of action detection and object detection in images.

- **frame-AP** measures the area under the precision-recall curve of the detections for each frame (similar to the PASCAL VOC detection challenge [11]). A detection is correct if the intersection-over-union with the ground truth at that frame is greater than σ and the action label is correctly predicted.
- **video-AP** measures the area under the precision-recall curve of the action tubes predictions. A tube is correct if the mean per frame intersection-over-union with the ground truth across the frames of the video is greater than σ and the action label is correctly predicted.
- **AUC** measures the area under the ROC curve, a metric previously used on this task. An action tube is correct under the same conditions as in *video-AP*. Following [38], the ROC curve is plotted until a false positive rate of 0.6, while keeping the top-3 detections per class and per video. Consequently, the best possible AUC score is 60%.

4.1. Results on UCF sports

In Figure 3, we plot the ROC curve for $\sigma = 0.2$ (red). In Figure 4 we plot the average AUC for different values of σ. We plot the curves as produced by the recent state-of-the-art approaches, Jain *et al*. [15], Wang *et al*. [42], Tian *et al*. [38] and Lan *et al*. [25]. Our approach outperforms all other techniques by a significant margin for all values of σ, showing the most improvement for high values of overlap, where other approaches tend to perform poorly. In particular, for $\sigma = 0.6$, our approach achieves an average AUC of 41.2% compared to 22.0% by [42].

Table 1 shows frame-AP (second row) and video-AP (third row) for an interestion-over-union threshold of $\sigma = 0.5$. Our approach achieves a mean AP of 68.1% at the frame level and 75.8% at the video level, with excellent performance for most categories. *Running* is the only action for which the action tubes fail to detect the actors (11.7 % video-AP) , even though our approach is able to localize them at the frame level (54.9% frame-AP). This is due to the fact that the test videos for *Running* contain multiple actors next to each other and our simple linking algorithm fails to consistently associate the detections with the correct actors, because of the proximity of the subjects and the presence of camera motion. In other words, the action tubes for *Running* contain the action but the detections do not always correspond to the same person. Indeed, if we make our evaluation agnostic to the instance, video-AP for *Running* is 83.8%. Tracking objects in a video is a very interesting but rather orthogonal problem to action localization and is beyond the scope of this work.

[1] The split was proposed by [25]

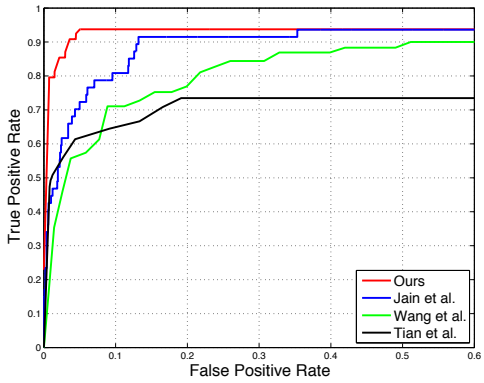


Figure 3: ROC curves on UCF Sports for an intersection-over-union threshold of $\sigma = 0.2$. Red shows our approach. We manage to reach a high true positive rate at a much smaller false positive rate, compared to the other approaches shown on the plot.

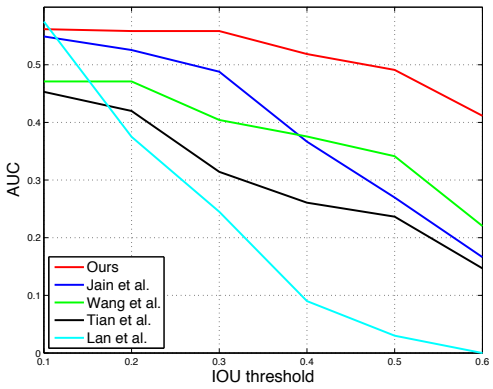


Figure 4: AUC on UCF Sports for various values of intersection-over-union threshold of σ (x-axis). Red shows our approach. We consistently outperform other approaches, with the biggest improvement being achieved at high values of overlap ($\sigma \geq 0.4$).

AP (%)	Diving	Golf	Kicking	Lifting	Riding	Running	Skateboarding	Swing1	Swing2	Walking	*mAP*
frame-AP	75.8	69.3	54.6	99.1	89.6	54.9	29.8	88.7	74.5	44.7	68.1
video-AP	100	91.7	66.7	100	100	11.7	41.7	100	100	45.8	75.8

Table 1: AP on the UCF Sports dataset for an intersection-over-union threshold of $\sigma = 0.5$. *frame-AP* measures AP of the action detections at the frame level, while *video-AP* measures AP of the predicted action tubes.

Figure 7 shows examples of detected action tubes on UCF sports. Each block corresponds to a different video. The videos were selected from the test set. We show the highest scoring action tube for each video. Red boxes indicate the detections in the corresponding frames. The predicted label is overlaid.

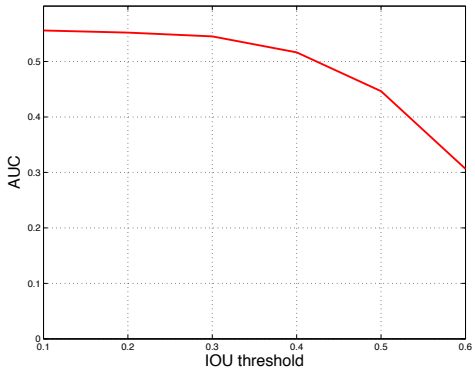


Figure 5: AUC on J-HMDB for different values of intersection-over-union threshold (averaged over the three splits).

4.2. Results on J-HMDB

We report frame-AP and video-AP for the 21 actions of J-HMDB. We present an ablation study of our approach by evaluating the performance of the two networks, *spatial-CNN* and *motion-CNN*. Table 2 shows the results for each method and for each action category.

As shown in the ablation study, it is apparent that the combination of spatial and motion-CNN performs significantly better for almost all actions. In addition, we can make some very useful observations. There are specific categories for which one signal matters more than the other. In particular, motion seems to be the most important for actions such as *Clap*, *Climb Stairs*, *Sit*, *Stand* and *Swing Baseball*, while appearance contributes more for actions such as *Catch*, *Shoot Gun* and *Throw*. Also, we notice that even though motion-CNN performs on average a bit worse than spatial-CNN at the frame level (24.3% vs. 27.0% respectively), it performs significantly better at the video level (45.7% vs. 37.9% respectively). This is due to the fact that the flow frames are not very informative when considered separately, however they produce a stronger overall prediction after the temporal smoothing provided by our linking algorithm.

Figure 5 shows the AUC for different values of the intersection-over-union threshold, averaged over the three splits on J-HMDB. Unfortunately, comparison with other approaches is not possible on this dataset, since no other approaches report numbers or have source code available.

Figure 8 shows examples of action tubes on J-HMDB. Each block corresponds to a different video. The videos are selected from the split 1 test set. We show the highest scoring action tube for each video. Red boxes indicate the detections in the corresponding frames. The predicted label is overlaid.

frame-AP (%)	brush_hair	catch	clap	climb_stairs	golf	jump	kick_ball	pick	pour	pullup	push	run	shoot_ball	shoot_bow	shoot_gun	sit	stand	swing_baseball	throw	walk	wave	*mAP*
spatial-CNN	55.8	**25.5**	25.1	24.0	77.5	1.9	5.3	21.4	68.6	71.0	15.4	6.3	4.6	41.1	**28.0**	9.4	8.2	19.9	**17.8**	29.2	11.5	27.0
motion-CNN	32.3	5.0	35.6	30.1	58.0	**7.8**	2.6	16.4	55.0	72.3	8.5	6.1	3.9	47.8	7.3	24.9	26.3	**36.3**	4.5	22.1	7.6	24.3
full	**65.2**	18.3	**38.1**	**39.0**	**79.4**	7.3	**9.4**	**25.2**	**80.2**	**82.8**	**33.6**	**11.6**	**5.6**	**66.8**	27.0	**32.1**	**34.2**	33.6	15.5	**34.0**	**21.9**	**36.2**
video-AP (%)																						
spatial-CNN	67.1	**34.4**	37.2	36.3	93.8	7.3	14.4	29.6	80.2	93.9	17.4	10.0	8.8	71.2	**45.8**	17.7	11.6	38.5	20.4	40.5	19.4	37.9
motion-CNN	66.3	16.0	**60.0**	51.6	88.6	**18.9**	10.8	23.9	83.4	96.7	18.2	17.2	**14.0**	84.4	19.3	**72.6**	**61.8**	**76.8**	17.3	46.7	14.3	45.7
full	**79.1**	33.4	53.9	**60.3**	**99.3**	18.4	**26.2**	**42.0**	**92.8**	**98.1**	**29.6**	**24.6**	13.7	**92.9**	42.3	67.2	57.6	66.5	**27.9**	**58.9**	**35.8**	**53.3**

Table 2: Results and ablation study on J-HMDB (averaged over the three splits). We report *frame-AP* (top) and *video-AP* (bottom) for the spatial and motion component and their combination (full). The combination of the spatial- and motion-CNN performs significantly better under both metrics, showing the significance of static and motion cues for the task of action recognition.

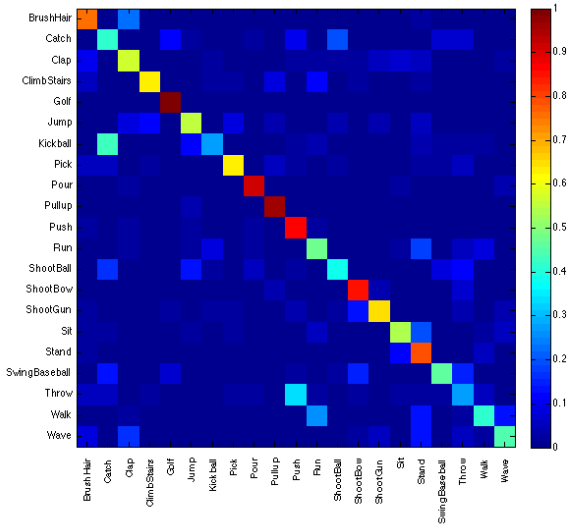


Figure 6: The confusion matrix on J-HMDB for the classification task, when using action tubes to predict a label for each video.

Action Classification Our approach is not limited to action detection. We can use the action tubes to predict an action label for the whole video. In particular, we can predict the label l for a video by picking the action with the maximum action tube score

$$l = \underset{\alpha \in A}{\operatorname{argmax}} \max_{\bar{R} \in \{\bar{R}_\alpha\}} S_\alpha(\bar{R}) \qquad (3)$$

where $S_\alpha(\bar{R})$ is the score of the action tube $\bar{R}$ as defined by Eq. 2.

If we use Eq. 3 as the prediction, our approach yields an accuracy of 62.5%, averaged over the three splits of J-HMDB. Figure 6 shows the confusion matrix.

In order to show the impact of the action tubes in the above result, we create a baseline model for action classification, similar to [35]. We use spatial and motion-CNNs in a classification setting, where full frames are used as input instead of regions. The weights of the CNNs are initialized from networks trained on UCF 101 (split1) for the task of action classification. We average the class probabilities as produced by the softmax layers of the CNNs, instead of training SVM classifiers (We observed major overfitting problems when training SVM classifiers on top of the combined fc7 features). We average the outputs of spatial- and motion-CNNs, with weights 1/3 and 2/3 respectively, and pick the action label with the maximum score after averaging the frames of the videos. This approach yields an accuracy of 56.5% averaged over the three splits of J-HMDB. This compares to 56.6% achieved by [40]. Table 3 summarizes the results for action classification on J-HMDB. It is quite evident that focusing on the actor is beneficial for the task of video classification, while a lot of information is being lost when the whole scene is analyzed in an orderless fashion.

Accuracy (%)	Wang *et al.* [40]	CNN (1/3 spatial, 2/3 motion)	Action Tubes
J-HMDB	56.6	56.5	**62.5**

Table 3: Classification accuracy on J-HMDB (averaged over the three splits). CNN (third column) shows the result of the weighted average of spatial and motion-CNN on the whole frames, while Action Tubes (fourth column) shows the result after using the scores of the predicted action tubes to make decisions for the video's label.

5. Conclusions

We propose an approach to action detection using convolutional neural networks on static and kinematic cues. We experimentally show that our action models perform state-of-the-art on the task of action localization. From our ablation study it is evident that appearance and motion cues are complementary and their combination is mandatory for accurate predictions across the board.

However, there are two problems closely related to action detection that we did not tackle. One is, as we mention in Section 4, the problem of tracking. For example, in a track field it is important to recognize that the athletes are running but also to be able to follow each one throughout the race. For this problem to be addressed, we need compelling datasets that contain videos of multiple actors, unlike the existing ones where the focus is on one or two actors. Second, camera motion is a factor which we did

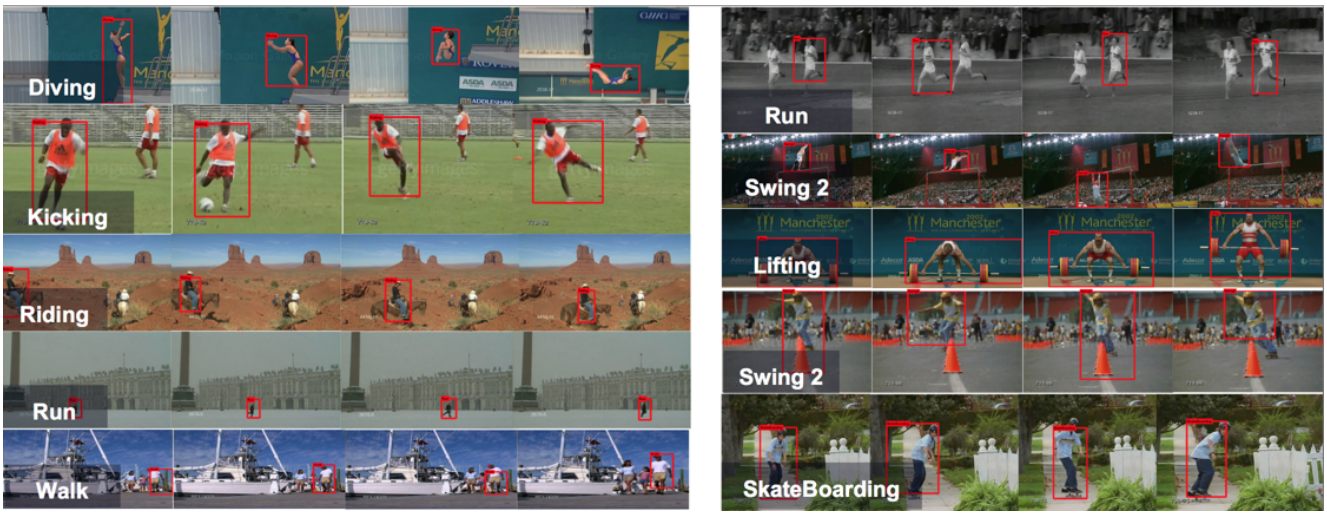


Figure 7: Examples from UCF Sports. Each block corresponds to a different video. We show the highest scoring action tube detected in the video. The red box indicates the region and the predicted label is overlaid. We show 4 frames from each video. The top example on the right shows the problem of tracking, while the 4th example on the right is a wrong prediction, with the true label being *Skate Boarding*.

Figure 8: Examples from J-HMDB. Each block corresponds to a different video. We show the highest scoring action tube detected in the video. The red box indicates the region and the predicted label is overlaid. We show 4 frames from each video. The 2nd example on the left and the two bottom ones on the right are wrong predictions, with true labels being *catch*, *sit* and *run* respectively.

not examine, despite strong evidence that it has a significant impact on performance [41, 16]. Efforts to eliminate the effect of camera movement, such as the one proposed by [41], might further improve our results.

Acknowledgements

This work was supported by the Intel Visual Computing Center and the ONR SMARTS MURI N000140911051. The GPUs used in this research were generously donated by the NVIDIA Corporation.

References

[1] J. Aggarwal and M. Ryoo. Human activity analysis: A review. *ACM Computing Surveys*, 2011. 2

[2] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, 2014. 3

[3] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, 2005. 2

[4] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004. 4

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 2

[6] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, 2006. 2

[7] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Competition 2012 (ILSVRC2012). http://www.image-net.org/challenges/LSVRC/2012/. 1

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[9] A. A. Efros, A. C. Berg, G., and J. Malik. Recognizing action at a distance. In *ICCV*, 2003. 2

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results. http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html. 1

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV*, 2010. 1, 5

[12] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *TPAMI*, 2010. 2

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1, 2, 3, 4

[14] M. A. Goodale and A. D. Milner. Separate visual pathways for perception and action. *Trends in Neurosciences*, 15:20–25, 1992. 1

[15] M. Jain, J. Gemert, H. Jegou, P. Bouthemy, and C. G. M. Snoek. Action localization with tubelets from motion. In *CVPR*, 2014. 1, 2, 5

[16] M. Jain, H. Jegou, and P. Bouthemy. Better exploiting motion for better action recognition. In *CVPR*, 2013. 2, 8

[17] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. Black. Towards understanding action recognition. In *ICCV*, 2013. 1, 5

[18] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *ICCV*, 2007. 2

[19] S. Ji, W. Hu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. In *PAMI*, 2013. 2

[20] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 4

[21] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classica- tion with convolutional neural networks. In *CVPR*, 2014. 2

[22] A. Klaser, M. Marszalek, C. Schmid, and A. Zisserman. Human Focused Action Localization in Video. In *ECCV*, 2010. 2

[23] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 2, 4

[24] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011. 5

[25] T. Lan, Y. Wang, and G. Mori. Discriminative figure-centric models for joint action localization and recognition. In *ICCV*, 2011. 1, 2, 5

[26] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 2

[27] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1989. 2

[28] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 2

[29] S. Mathe and C. Sminchisescu. Dynamic Eye Movement Datasets and Learned Saliency Models for Visual Action Recognition. In *ECCV*, 2012. 3

[30] R. Poppe. A survey on vision-based human action recognition. *Image Vision Computing*, 2010. 2

[31] A. Prest, V. Ferrari, and C. Schmid. Explicit modeling of human-object interactions in realistic videos. *PAMI*, 2012. 2

[32] M. Raptis, I. Kokkinos, and S. Soatto. Discovering discriminative action parts from mid-level video representations. In *CVPR*, 2012. 2

[33] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach: a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008. 1, 5

[34] C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *ICPR*, 2004. 2

[35] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 1, 2, 4, 7

[36] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human action classes from videos in the wild. In *CRCV-TR-12-01*, 2012. 4

[37] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *ECCV*, 2010. 2

[38] Y. Tian, R. Sukthankar, and M. Shah. Spatiotemporal deformable part models for action detection. In *CVPR*, 2013. 1, 2, 5

[39] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *IJCV*, 2013. 3

[40] H. Wang, A. Kläser, C. Schmid, and C. Liu. Action recognition by dense trajectories. In *CVPR*, 2011. 1, 2, 7

[41] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. 1, 2, 8

[42] L. Wang, Y. Qiao, and X. Tang. Video action detection with relational dynamic-poselets. In *ECCV*, 2014. 1, 2, 5

[43] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 2011. 2

[44] M. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 4