

# Depth2Action: Exploring Embedded Depth for Large-Scale Action Recognition

Yi Zhu and Shawn Newsam

University of California, Merced  
{yzhu25, snewsam}@ucmerced.edu

**Abstract.** This paper performs the first investigation into depth for large-scale human action recognition in video *where the depth cues are estimated from the videos themselves*. We develop a new framework called *depth2action* and experiment thoroughly into how best to incorporate the depth information. We introduce spatio-temporal depth normalization (STDN) to enforce temporal consistency in our estimated depth sequences. We also propose modified depth motion maps (MDMM) to capture the subtle temporal changes in depth. These two components significantly improve the action recognition performance. We evaluate our depth2action framework on three large-scale action recognition video benchmarks. Our model achieves state-of-the-art performance when combined with appearance and motion information thus demonstrating that depth2action is indeed complementary to existing approaches.

**Keywords:** Action Recognition, Embedded Depth

## 1 Introduction

Human action recognition in video is a fundamental problem in computer vision due to its increasing importance for a range of applications such as analyzing human activity, video search and recommendation, complex event understanding, etc. Much progress has been made over the past several years by employing hand-crafted local features such as improved dense trajectories (IDT) [39] or video representations that are learned directly from the data itself using deep convolutional neural networks (ConvNets). However, starting with the seminal two-stream ConvNets method [31], approaches have been limited to exploiting static visual information through frame-wise analysis and/or translational motion through optical flow or 3D ConvNets. Further increase in performance on benchmark datasets has been mostly due to the higher capacity of deeper networks [44,43,23,46] or to recurrent neural networks which model long-term temporal dynamics [24,47,2].

Intuitively, *depth* can be an important cue for recognizing complex human actions. Depth information can help differentiate between action classes that are otherwise very similar especially with respect to appearance and translational motion in the red-green-blue (RGB) domain. For instance, the “CricketShot” and “CricketBowling” classes in the UCF101 dataset are often confused by the



Fig. 1: (a) “CricketBowling” and (b) “CricketShot”. Depth information about the bowler and the batters is key to telling these two classes apart. Our proposed *depth2action* approach exploits the depth information that is embedded in the videos to perform large-scale action recognition. This figure is best viewed in color

state-of-the-art models [44,46]. This makes sense because, as shown in Fig. 1, these classes can be very similar with respect to static appearance, human-object interaction, and in-plane human motion patterns. Depth information about the bowler and the batters is key to telling these two classes apart.

Previous work on depth for action recognition [3,40,50,45] uses depth information obtained from *depth sensors* such as Kinect-like devices and thus is not applicable to large-scale action recognition in RGB video. We instead estimate the depth information *directly from the video itself*. This is a difficult problem which results in noisy depth sequences and so a major contribution of our work is how to effectively extract the subtle but informative depth cues. To our knowledge, our work is the first to perform large-scale action recognition based on depth information embedded in the video data.

Our novel contributions are as follows: (i) We introduce *depth2action*, a novel approach for human action recognition using depth information embedded in videos. It is shown to be complementary to existing approaches which exploit spatial and translational motion information and, when combined with them, achieves state-of-the-art performance on three popular benchmarks. (ii) We propose STDN to enforce temporal consistency and MDMM to capture the subtle temporal depth cues in noisy depth sequences. (iii) We perform a thorough investigation on how best to extract and incorporate the depth cues including: image- versus video-based depth estimation; multi-stream 2D ConvNets versus 3D ConvNets to jointly extract spatial and temporal depth information; ConvNets as feature extractors versus end-to-end classifiers; early versus late fusion of features for optimal prediction; and other design choices.

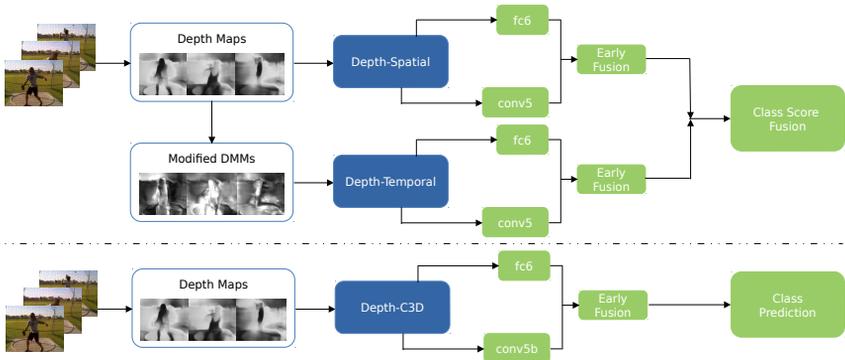


Fig. 2: **Depth2Action framework.** Top: Our *depth two-stream* model. Depth maps are estimated on a per-frame basis and input to a depth-spatial net. Modified depth motion maps (MDMMs) are derived from the depth maps and input to a depth-temporal net. Features are extracted, concatenated and input to two support vector machine (SVM) classifiers, to obtain the final prediction scores. Bottom: Our *depth-C3D* framework which is similar except the depth maps are input to a single depth-C3D net which jointly captures spatial and temporal depth information. This figure is best viewed in color

## 2 Related Work

There exists an extensive body of literature on human action recognition. We review only the most related work.

**Deep ConvNets:** Improved dense trajectories [39] dominated the field of video analysis for several years until the two-stream ConvNets architecture introduced by Simonyan and Zisserman [31] achieved competitive results for action recognition in video. In addition, motivated by the great success of applying deep ConvNets in image analysis, researchers have adapted deep architectures to the video domain either for feature representation [43,48,37,52,35] or end-to-end prediction [13,44,24,47].

While our framework shares some structural similarity with these works, it is distinct and complementary in that it exploits *depth* for action recognition. All the works above are based on appearance and translational motion in the RGB domain. We note there has been some work that exploits audio information [26]; however, not all videos come with audio and our approach is complementary to this work as well.

**RGB-D Based Action Recognition:** There is previous work on action recognition in RGB-D data. Chen et al. [3] use depth motion maps (DMM) for real-time human action recognition. Yang and Tian [50] cluster hypersurface normals in depth sequences to form a super normal vector (SNV) representation. Very recently, Wang et al. [45] apply weighted hierarchical DMM and deep ConvNets to achieve state-of-the-art performance on several benchmarks. Our work is different from approaches that use RGB-D data in several key ways:

(i) *Depth information source and quality*: These methods use depth information obtained from depth sensors. Besides limiting their applicability, this results in depth sequences that have much higher fidelity than those which can be estimated from RGB video. Our estimated depth sequences are too noisy for recognition techniques designed for depth-sensor data. Taking the difference between consecutive frames in our depth sequences only amplifies this noise making techniques such as STOP features [38], SNV representations [50], and DMM-based framework [3,45], for example, ineffective.

(ii) *Benchmark datasets*: RGB-D benchmarks such as MSRAction3D [19], MSRDailyActivity3D [42], MSRGesture3D [41], MSROnlineAction3D [51] and MSRActionPairs3D [27] are much more limited in terms of the diversity of action classes and the number of samples. Further, the videos often come with other meta data like skeleton joint positions. In contrast, our benchmarks such as UCF101 contain large numbers of action classes and the videos are less constrained. Recognition is made more difficult by the large intra-class variation.

We note that we take inspiration from [49,45] in designing our modified DMMs. The approaches in these works use RGB-D data and are not appropriate for our problem, though, since they construct multiple depth sequences using different geometric projections, and our videos are too long and our estimated depth sequences too noisy to be characterized by a single DMM.

In summary, our depth2action framework is novel compared to previous work on action recognition. An overview of our framework can be found in Fig. 2.

### 3 Methodology

Since our videos do not come with associated depth information, we need to extract it directly from the RGB video data. We consider two state-of-the-art approaches to *efficiently* extract depth maps from the individual video frames. We enforce temporal consistency in these sequences through inter-frame normalization. We explore different ConvNets architectures to extract spatial and temporal depth cues from the normalized depth sequences.

#### 3.1 Depth Extraction

Extracting depth maps from video has been studied for some time now [53,34,29]. Most approaches, however, are not applicable since they either require stereo video or additional information such as geometric priors. There are a few works [22] which extract depth maps from monocular video alone but they are computationally too expensive which does not scale to problems like ours.

We therefore turn to frame-by-frame depth extraction and enforce temporal consistency through a normalization step. Depth from images has made much progress recently [20,1,14,4] and is significantly more efficient for extracting depth from video. We consider two state-of-the-art approaches to extract depth from images, [20] and [4], based on their accuracy and efficiency.

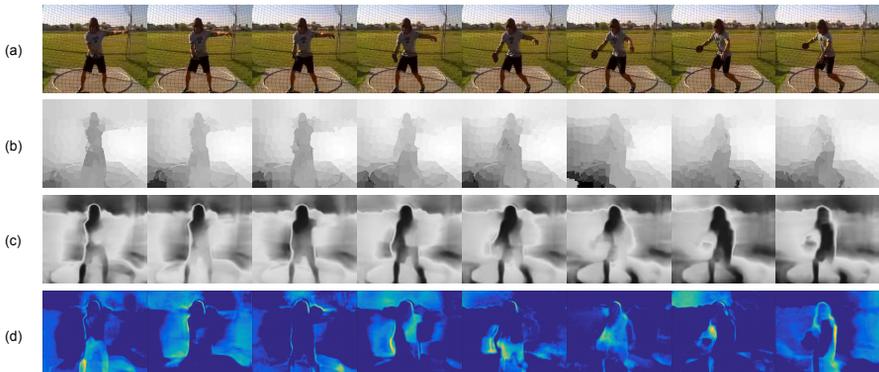


Fig. 3: Depth maps estimated from the video `v_ThrowDiscus_g05_c02.avi` in the UCF101 dataset. (a): raw RGB frames; (b): depth maps extracted using [20]; (c): depth maps extracted using [4]; (d): the absolute difference between consecutive depth maps in (c). Blue indicates smaller values and yellow larger ones. This figure is best viewed in color

**Deep Convolutional Neural Fields (DCNF) [20]:** This work jointly explores the capacity of deep ConvNets and continuous CRFs to estimate depth from an image. Depth is predicted through maximum a posterior (MAP) inference which has a closed-form solution. We apply the implementation kindly provided by the authors [20] but discard the time consuming “inpainting” procedure which is not important for our application. Our modified implementation takes only 0.09s per frame to extract a depth map.

**Multi-scale Deep Network [4]:** Unlike DCNF above, this method does not utilize super-pixels and thus results in smoother depth maps. It uses a sequence of scales to progressively refine the predictions and to capture image details both globally and locally. Although the model can also be used to predict surface normals and semantic labels within a common ConvNets architecture, we only use it to extract depth maps. Our modified implementation takes only 0.01s per frame to extract a depth map.

Fig. 3 visually compares the per-frame depths maps generated by the two approaches. We observe that 1) [4] (Fig. 3c) results in smoother maps since it does not utilize super-pixels like [20] (Fig. 3b), and 2) [4] preserves structural details, such as the border between the sky and the trees, better than [20] due to its multi-scale refinement. An ablation study (see supplemental materials) shows [4] results in better action recognition performance so we use it to extract per-frame depth maps for the rest of the paper.

### 3.2 Spatio-Temporal Depth Normalization

We now have depth sequences. While this makes our problem similar to work on action recognition from depth-sensor data such as [45], these methods are not

applicable for a number of reasons. First, their inputs are point clouds which allows them to derive depth sequences from multiple perspectives for a single video as well as augment their training data through virtual camera movement. We only have a single fixed viewpoint. Second, their depth information has much higher fidelity since it was acquired with a depth sensor. Ours is prohibitively noisy to use a single 2D depth motion map to represent an entire video as is done in [45]. We must develop new methods.

The first step is to reduce the noise by enforcing temporal consistency under the assumption that depth does not change significantly between frames. We introduce a temporal normalization scheme which constrains the furthest part of the scene to remain approximately the same throughout a clip. We find this works best when applied separately to three horizontal spatial windows and so we term the method spatio-temporal depth normalization (STDN). Specifically, let  $\mathbf{x}$  be a frame. We then take  $n$  consecutive frames  $[\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, \dots, \mathbf{x}_{t_n}]$  to form a volume (clip) which is divided spatially into three equal-sized subvolumes that represent the top, middle, and bottom parts [25]. We take the 95<sup>th</sup> percentile of the depth distribution as the furthest scene element in each subvolume. The 95<sup>th</sup> percentile of the corresponding window in each frame is then linearly scaled to equal this furthest distance.

We also investigated other methods to enforce temporal consistency including intra-frame normalization, temporal averaging (uniform as well as Gaussian) with varying temporal window sizes, and warping. None performed as well as the proposed STDN (see supplemental materials).

### 3.3 ConvNets Architecture Selection

Recent progress in action recognition based on ConvNets can be attributed to two models: a two-stream approach based on 2D ConvNets [31,44] which separately models the spatial and temporal information, and 3D ConvNets which jointly learn spatio-temporal features [11,37]. These models are applied to RGB video sequences. We explore and adapt them for our depth sequences.

**2D ConvNets:** In [31], the authors compute a spatial stream by adapting 2D ConvNets from image classification [15] to action recognition. We do the same here except we use depth sequences instead of RGB video sequences. We term this our *depth-spatial stream* to distinguish it from the standard spatial stream which we will refer to as RGB-spatial stream for clarity. Our depth-spatial stream is pre-trained on the ILSVRC-2012 dataset [30] with the VGG-16 implementation [32] and fine-tuned on our depth sequences. [31] also computes a temporal stream by applying 2D ConvNets to optical flow derived from the RGB video. We could similarly compute optical flow from our depth sequences but this would be redundant (and very noisy) so we instead propose a different depth-temporal stream below in section 3.4.

**3D ConvNets:** In [11,37], the authors show that 2D ConvNets “forget” the temporal information in the input signal after every convolution operation. They propose 3D ConvNets which analyze sets of contiguous video frames organized

as clips. We apply this approach to clips of depth sequences. We term this *depth-C3D* to distinguish it from the standard 3D ConvNets which we will refer to as RGB-C3D for clarity. Our depth-C3D net is pre-trained using the Sports-1M dataset [13] and fine-tuned on our depth sequences.

### 3.4 Depth-Temporal Stream

Here, we look to augment our depth-spatial stream with a depth-temporal stream. We take inspiration from work on action recognition from depth-sensor data and adapt depth motion maps [49] to our problem. In [49], a single 2D DMM is computed for an entire sequence by thresholding the difference between consecutive depth maps to get per-frame (binary) motion energy and then summing this energy over the entire video. A 2D DMM summarizes where depth motion occurs.

We instead calculate the motion energy as the absolute difference between consecutive depth maps *without thresholding* in order to retain the subtle motion information embedded in our noisy depth sequences. We also accumulate the motion energy over clips instead of entire sequences since the videos in our dataset are longer and less-constrained compared to the depth-sensor sequences in [19,27,42,41,51] and so our depth sequences are too noisy to be summarized over long periods. In many cases, the background would simply dominate.

We compute one modified depth motion map (MDMM) for a clip of  $N$  depth maps as

$$\text{MDMM}_{t_{start}} = \sum_{t_{start}}^{t_{start}+N} |\text{map}^{t_{start}+1} - \text{map}^{t_{start}}|, \quad (1)$$

where  $t_{start}$  is the first frame of the clip,  $N$  is the duration of the clip, and  $\text{map}^t$  is the depth map at frame  $t$ . Multiple MDMMs are computed for each video. Each MDMM is then input to a 2D ConvNet for classification. We term this our *depth-temporal stream*. We combine it with our depth-spatial stream to create our *depth two-stream* (see Fig. 2). Similar to the depth-spatial stream, the depth-temporal stream is pre-trained on the ILSVRC-2012 dataset [30] with the VGG-16 network [32] and fine-tuned on the MDMMs.

We also consider a simpler temporal stream by taking the absolute difference between adjacent depth maps and inputting this difference sequence to a 2D ConvNet. We term this our *baseline depth-temporal stream*. Fig. 3d shows an example sequence of this difference. It does a good job at highlighting changes in the depth despite the noisiness of the image-based depth estimation.

### 3.5 ConvNets: Feature Extraction or End-to-End Classification

The ConvNets in our depth two-stream and depth-C3D models default to end-to-end classifiers. We investigate whether to use them instead as feature extractors followed by SVM classifiers. This also allows us to investigate early versus late fusion. We use our depth-spatial stream for illustration.

Features are extracted from two layers of our fine-tuned ConvNets. We extract the activations of the first fully-connected layer (fc6) on a per-frame basis.

These are then averaged over the entire video and L2-normalized to form a 4096-dim video-level descriptor. We also extract activations from the convolutional layers as they contain spatial information. We choose the `conv5` layer, whose feature dimension is  $7 \times 7 \times 512$  (7 is the size of the filtered images of the convolutional layer and 512 is the number of convolutional filters). By considering each convolutional filter as a latent concept, the `conv5` features can be converted into  $7^2$  latent concept descriptors (LCD) [48] of dimension 512. We also adopt a spatial pyramid pooling (SPP) strategy [7] similar to [48]. We apply principle component analysis (PCA) to de-correlate and reduce the dimension of the LCD features to 64 and then encode them using vectors of locally aggregated descriptors (VLAD) [10]. This is followed by intra- and L2-normalization to form a 16384-dim video-level descriptor.

Early fusion consists of concatenating the `fc6` and `conv5` features for input to a single multi-class linear SVM classifier [5] (see Fig. 2). Late fusion consists of feeding the features to two separate SVM classifiers and computing a weighted average of their probabilities. The optimal weights are selected by grid-search.

## 4 Experiments

The goal of our experiments is two-fold. First, to explore the various design options described in section 3 Methodology. Second, to show that our `depth2action` framework is complementary to standard approaches to large-scale action recognition based on appearance and translational motion and achieves state-of-the-art results when combined with them.

### 4.1 Datasets

We perform experiments on three widely-used publicly-available action recognition benchmark datasets, UCF101 [33], HMDB51 [16], and ActivityNet [8].

**UCF101** is composed of realistic action videos from YouTube. It contains 13320 videos in 101 action classes. It is one of the most popular benchmark datasets because of its diversity in terms of actions and the presence of large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc. **HMDB51** is composed of 6766 videos in 51 action classes extracted from a wide range of sources. It contains both original videos as well as stabilized ones, but we only use the original videos. Both UCF101 and HMDB51 have a standard three split evaluation protocol and we report the average recognition accuracy over the three training and test splits. As suggested by the authors in [8], we use **ActivityNet** release 1.2 for our experiments due to the noisy crowdsourced labels in release 1.1. The second release consists of 4819 training, 2383 validation, and 2480 test videos in 100 activity classes. Though the number of videos and classes are similar to UCF101, ActivityNet is a much more challenging benchmark because it has greater intra-class variance and consists of longer, untrimmed videos. The evaluation metric we used in this paper is top-1 accuracy for all three datasets.

Table 1: Recognition performance of our proposed configurations on three benchmark datasets. (a): Our spatio-temporal depth normalization (STDN) indicated by (N) is shown to improve performance for all configurations on all datasets. (b): Using the ConvNets to extract features is better than using them as end-to-end classifiers. Also, early fusion of features is better than late fusion of SVM probabilities. See the text for discussion on depth two-stream versus depth-C3D

(a) Effectiveness of STDN

(b) Features or End-to-End Classifier

Model	UCF101	HMDB51	ActivityNet
Depth-Spatial	58.8%	37.9%	35.9%
Depth-Spatial (N)	59.1%	38.3%	36.4%
Depth-Temporal Baseline	61.8%	40.6%	38.2%
Depth-Temporal Baseline (N)	63.3%	42.0%	39.8%
Depth-Temporal	63.9%	42.6%	39.7%
Depth-Temporal (N)	65.1%	43.5%	40.9%
Depth Two-Stream	65.6%	44.2%	42.7%
Depth Two-Stream (N)	<b>67.0%</b>	<b>45.4%</b>	44.2%
Depth-C3D	61.7%	40.9%	45.9%
Depth-C3D (N)	63.8%	42.8%	<b>47.4%</b>

Model	UCF101	HMDB51	ActivityNet
Depth Two-Stream	67.0%	45.4%	44.2%
Depth Two-Stream fc6	68.2%	46.5%	45.3%
Depth Two-Stream conv5	70.1%	48.2%	47.0%
Depth Two-Stream Early	<b>72.5%</b>	<b>49.7%</b>	49.6%
Depth Two-Stream Late	70.9%	48.9%	48.7%
Depth-C3D	63.8%	42.8%	47.4%
Depth-C3D fc6	64.9%	43.9%	47.9%
Depth-C3D conv5b	66.7%	45.0%	49.1%
Depth-C3D Early	69.5%	46.6%	<b>52.1%</b>
Depth-C3D Late	67.8%	45.7%	51.0%

## 4.2 Implementation Details

We use the Caffe toolbox [12] to implement the ConvNets. The network weights are learned using mini-batch stochastic gradient descent (256 frames for two-stream ConvNets and 30 clips for 3D ConvNets) with momentum (set to 0.9).

**Depth Two-Stream:** We adapt the VGG-16 architecture [32] and use ImageNet models as the initialization for both the depth-spatial and depth-temporal net training. As in [44], we adopt data augmentation techniques such as corner cropping, multi-scale cropping, horizontal flipping, etc. to help prevent overfitting, as well as high dropout ratios (0.9 and 0.8 for the fully connected layers). The input to the depth-spatial net is the per-frame depth maps, while the input to the depth-temporal net is either the depth difference between adjacent frames (in the baseline case) or the MDMMs. For generating the MDMMs, we set  $N$  in equation 1 to 10 frames as a subvolume. For the depth-spatial net, the learning rate decreases from 0.001 to 1/10 of its value every 15K iterations, and the training stops after 66K iterations. For the depth-temporal net, the learning rate starts at 0.005, decreases to 1/10 of its value every 20K iterations, and the training stops after 100K iterations.

**Depth-C3D:** We adopt the same architecture as in [37]. The Depth-C3D net is pre-trained on the Sports-1M dataset [13] and fine-tuned on estimated depth sequences. During fine-tuning, the learning rate is initialized to 0.005, decreased to 1/10 of its value every 8K iterations, and the training stops after 34K iterations. Dropout is applied with a ratio of 0.5.

Note that since the number of training videos in the HMDB51 dataset is relatively small, we use ConvNets fine-tuned on UCF101, except for the last layer, as the initialization (for both 2D and 3D ConvNets). The fine-tuning stage starts with a learning rate of  $10^{-5}$  and converges in one epoch.

### 4.3 Results

**Effectiveness of STDN:** Table 1(a) shows the performance gains due to our proposed normalization. STDN improves recognition performance for all approaches on all datasets. The gain is typically around 1-2%. We set the normalization window ( $n$  in section 3.2) to 16 frames for UCF101 and ActivityNet, and 8 frames for HMDB51. We further observe that (i) Depth-C3D benefits from STDN more than depth two-stream. This is possibly because the input to depth-C3D is a 3D volume of depth sequences while the input to depth two-stream is the individual depth maps. Temporal consistency is important for the 3D volume. (ii) Depth-temporal benefits from STDN more than depth-spatial. This is expected since the goal of the normalization is to improve the temporal consistency of the depth sequences and only the depth-temporal stream “sees” multiple depth-maps at a time. From now on, all results are based on depth sequences that have been normalized.

**Depth Two-Stream versus Depth-C3D:** As shown in Table 1(a), depth two-stream performs better than depth-C3D for UCF101 and HMDB51, while the opposite is true for ActivityNet. This suggests that depth-C3D may be more suitable for large-scale video analysis. Though the second release of ActivityNet has a similar number of action clips as UCF101, in general, the video duration is much (30 times) longer than that of UCF101. Similar results for 3D ConvNets versus 2D ConvNets was observed in [21]. The computational efficiency of depth-C3D also makes it more suitable for large-scale analysis. Although our depth-temporal net is much faster than the RGB-temporal net (which requires costly optical flow computation), depth-two stream is still significantly slower than depth-C3D. We therefore recommend using depth-C3D for large-scale applications.

**ConvNets for Feature Extraction versus End-to-End Classification:** Table 1(b) shows that treating the ConvNets as feature extractors performs significantly better than using them for end-to-end classification. This agrees with the observations of others [2,37,54]. We further observe that the VLAD encoded conv5 features perform better than fc6. This improvement is likely due to the additional discriminative power provided by the spatial information embedded in the convolutional layers. Another attractive property of using feature representations is that we can manipulate them in various ways to further improve the performance. For instance, we can employ different (i) encoding methods: Fisher vector [25], VideoDarwin [6]; (ii) normalization techniques: rank normalization [18]; and (iii) pooling methods: line pooling [54], trajectory pooling [43,54], etc.

**Early versus Late Fusion:** Table 1(b) also shows that early fusion of features through concatenation performs better than late fusion of SVM probabilities. Late fusion not only results in a performance drop of around 1.0% but also requires a more complex processing pipeline since multiple SVM classifiers need to be trained. UCF101 benefits from early fusion more than the other two datasets. This might be due to the fact that UCF101 is a trimmed video dataset and so the content of individual videos varies less than in the other two datasets. Early fusion of multiple layers’ activations is typically more robust to noisy data.

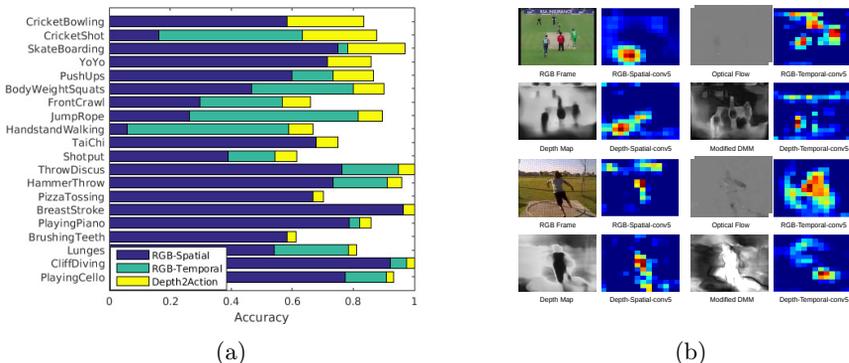


Fig. 4: (a) Recognition results on the first split of UCF101. Plot showing the classes for which our proposed depth2action framework (yellow) outperforms RGB-spatial (blue) and RGB-temporal (green) streams. (b) Visualizing the convolutional feature maps of four models: RGB-spatial, RGB-temporal, depth-spatial, and depth-temporal. Pairs of inputs and resulting feature maps are shown for each model for two actions, “CricketBowling” and “ThrowDiscus”. This figure is best viewed in color

**Depth2Action:** We thus settle on our proposed depth2action framework. For medium-scale video datasets like UCF101 and HMDB51, we perform early fusion of conv5 and fc6 features extracted using a depth two-stream configuration. For large-scale video datasets like ActivityNet, we perform early fusion of conv5b and fc6 features extracted using a depth-C3D configuration. These two models are shown in Fig. 2.

#### 4.4 Discussion

**Class-Specific Results:** We investigate the specific classes for which depth information is important. To do this, we compare the per-class performance of our depth2action framework with standard methods that use appearance and translational motion in the RGB domain. We first compute the performances of an RGB-spatial stream which takes the RGB video frames as input and an RGB-temporal stream which takes optical flow (computed in the RGB domain) as input. We then identify the classes for which our depth2action performs better than both the RGB-spatial and RGB-temporal streams. We compute these results for the first split of the UCF101 dataset. Fig. 4a shows the 20 classes for which our depth2action framework performs best (in order of decreasing improvement). For example, for the class CricketShot, RGB-spatial achieves an accuracy of around 0.18, RGB-temporal achieves around 0.62, while our depth2action achieves around 0.88. (For those classes where RGB-spatial performs better than RGB-temporal, we simply do not show the performance



Fig. 5: Sample video frames of action classes that benefit from depth information. Left: UCF101. Right: HMDB51. This figure is best viewed in color

of RGB-temporal.) Depth2action clearly represents a complementary approach especially for classes where the RGB-spatial and RGB-temporal streams perform relatively poorly such as CricketBowling, CricketShot, FrontCrawl, HammerThrow, and HandStandWalking. Recall from Fig. 1 that CricketBowling and CricketShot are very similar with respect to appearance and translational motion. These are shown to be the two classes for which depth2action provides the most improvement, achieving respectable accuracies of above 0.8.

Sample video frames from classes in the UCF101 (left) and HMDB51 (right) datasets which benefit from depth information are shown in Fig. 5 (see supplemental materials for more samples).

**Visualizing Depth2Action:** We visualize the convolutional feature maps (conv5) to better understand how depth2action encodes depth information and how this encoding is different from that of RGB two-stream models. Fig. 4b shows pairs of inputs and resulting feature maps for four models: RGB-spatial, RGB-temporal, depth-spatial, and depth-temporal. (The feature maps are displayed using a standard heat map in which warmer colors indicate larger values.) The top four pairs are for “CricketBowling” and bottom four pairs are for “ThrowDiscus” (see supplemental materials for more action classes).

In general, the depth feature maps are sparser and more accurate than the RGB feature maps, especially for the temporal streams. The depth-spatial stream correctly encodes the bowler and the batter in “CricketBowling” and the discus thrower in “ThrowDiscus” as being salient while the RGB-stream gets distracted by other parts of the scene. The depth-temporal stream clearly identifies the progression of the bowler into the scene in “CricketBowling” and the movement of the discus thrower’s leg into the scene in “ThrowDiscus” as being salient while the RGB-temporal stream is distracted by translational movement throughout the scene. These results demonstrate that our proposed depth2action approach does indeed focus on the correct regions in classes for which depth is important.

**What about IDT?** We compare our depth2action framework with improved dense trajectories (IDT) computed from RGB video. IDT has been shown to be

Table 2: Comparison of RGB two-stream, IDT computed from RGB video, and depth2action, and their combinations for the UCF101 dataset.  $\Delta$  indicates the performance increase with respect to RGB two-stream taken as the baseline

Model	split01	split02	split03	Average	$\Delta$
RGB Two-Stream Baseline	90.8%	92.0%	91.3%	91.4%	0
IDT	83.1%	85.9%	85.1%	84.7%	-
Depth2Action	71.9%	73.0%	72.5%	72.5%	-
RGB Two-Stream+IDT	91.9%	92.6%	91.8%	92.1%	0.8%
RGB Two-Stream+Depth2Action	91.7%	92.5%	91.8%	92.0%	0.7%
Depth2Action+IDT	84.3%	86.6%	86.4%	85.8%	-
RGB Two-Stream+Depth2Action+IDT	92.5%	93.8%	92.8%	93.0%	1.8%

the best hand-crafted features for action recognition [39]. It is known to perform well under various camera motions (e.g. pan, tilt and zoom) and zoom can be considered global depth change.

While the top part of Table 2 shows that IDT outperforms depth2action, which is not surprising due to how noisy our estimated depth maps are, we turn our attention to the performance obtained by combining these two approaches with an RGB two-stream model. Rows four and five show that the performance achieved by combining depth2action with RGB two-stream is on par with the combination of IDT with RGB two-stream (and both perform significantly better than IDT). The last column shows the improvement over RGB two-stream alone. This demonstrates that although depth2action is not as effective as IDT when taken alone, *it is as complementary to RGB two-stream as IDT*. This point is even more significant given the fact that IDT requires several orders of magnitude more computation time and storage space (mainly to extract and store the features) than depth2action. The combination of depth2action and RGB two-stream is much preferred over that of IDT and RGB two-stream for large-scale analysis. The last row of Table 2 shows the results of combining all three approaches. We again get an improvement. This result turns out to be state-of-the-art for this dataset and stresses the importance and complementarity of jointly exploiting appearance, translational motion, and depth for action recognition.

#### 4.5 Comparison with State-of-the-art

Table 3 compares our approach with a large number of recent state-of-the-art published results on the three benchmarks. For UCF101 and HMDB51, the reported performance is the mean recognition accuracy over the standard three splits. The last row shows the performance of combining depth2action with RGB two-stream for UCF101 and HMDB51, and RGB C3D for ActivityNet, and also IDT features. We achieve state-of-the-art results on all three datasets through this combination, again stressing the importance of appearance, motion, and depth for action recognition.

Table 3: Comparison with the state-of-the-art. \* indicates the results are from our implementation of the method. Two-stream and C3D here is RGB based

Algorithm	UCF101	Algorithm	HMDB51	Algorithm	ActivityNet
Srivastava et al. [35]	84.3%	Srivastava et al. [35]	44.1%	Wang & Schmid [39]	61.3%*
Wang & Schmid [39]	85.9%	Oneata et al. [25]	54.8%	Simonyan & Zisserman [31]	67.1%*
Simonyan & Zisserman [31]	88.0%	Wang & Schmid [39]	57.2%	Tran et al. [37]	69.4%*
Jain et al. [9]	88.5%	Simonyan & Zisserman [31]	59.1%		
Ng et al. [24]	88.6%	Sun et al. [36]	59.1%		
Lan et al. [17]	89.1%	Jain et al. [9]	61.4%		
Zha et al. [52]	89.6%	Fernando et al. [6]	63.7%		
Tran et al. [37]	90.4%	Lan et al. [17]	65.1%		
Wu et al. [47]	91.3%	Wang et al. [43]	65.9%		
Wang et al. [43]	91.5%	Peng et al. [28]	66.8%		
Depth2Action	72.5%	Depth2Action	49.7%	Depth2Action	52.1%
+Two-Stream	92.0%	+Two-Stream	67.1%	+C3D	71.2%
+IDT+Two-Stream	<b>93.0%</b>	+IDT+Two-Stream	<b>68.2%</b>	+IDT+C3D	<b>73.4%</b>

We note that since there are no published results<sup>1</sup> for release 1.2 of ActivityNet, we report the results from our implementations of IDT [39], RGB two-stream [31] and RGB C3D [37].

## 5 Conclusion

We introduced *depth2action*, the first investigation into depth for large-scale human action recognition where the depth cues are derived from the videos themselves rather than obtained using a depth sensor. This greatly expands the applicability of the method. Depth is estimated on a per-frame basis for efficiency and temporal consistency is enforced through a novel normalization step. Temporal depth information is captured using modified depth motion maps. A wide variety of design options are explored. Depth2action is shown to be complementary to standard approaches based on appearance and translational motion, and achieves state-of-the-art performance on three benchmark datasets when combined with them.

In addition to advancing state-of-the-art performance, the depth2action framework is a rich research problem. It bridges the gap between the RGB- and RGB-D-based action recognition communities. It consists of numerous interesting sub-problems such as fine-grained action categorization, depth estimation from single images/video, learning from noisy data, etc. The estimated depth information could also be used for other applications such as object detection/segmentation, event recognition, and scene classification. We will make our trained models and estimated depth maps publicly available for future research.

**Acknowledgements.** This work was funded in part by a National Science Foundation CAREER grant, #IIS-1150115, and a seed grant from the Center for Information Technology in the Interest of Society (CITRIS). We gratefully acknowledge the support of NVIDIA Corporation through the donation of the Titan X GPU used in this work.

<sup>1</sup> The up-to-date leaderboard is at <http://activity-net.org/evaluation.html>.

## References

1. Baig, M.H., Torresani, L.: Coupled Depth Learning. In: WACV (2016)
2. Ballas, N., Yao, L., Pal, C., Courville, A.: Delving Deeper into Convolutional Networks for Learning Video Representations. In: ICLR (2016)
3. Chen, C., Liu, K., Kehtarnavaz, N.: Real-time Human Action Recognition Based on Depth Motion Maps. *Journal of Real-Time Image Processing* (2013)
4. Eigen, D., Fergus, R.: Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture. In: ICCV (2015)
5. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research* (2008)
6. Fernando, B., Gavves, E., M., J.O., Ghodrati, A., Tuytelaars, T.: Modeling Video Evolution for Action Recognition. In: CVPR (2015)
7. He, K., Zhang, X., Ren, S., Sun, J.: Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. TPAMI (2015)
8. Heilbron, F.C., Escorcia, V., Ghanem, B., Niebles, J.C.: ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding. In: CVPR (2015)
9. Jain, M., van Gemert, J.C., Snoek, C.G.M.: What do 15,000 Object Categories Tell Us about Classifying and Localizing Actions? In: CVPR (2015)
10. Jegou, H., Perronnin, F., Douze, M., Sanchez, J., Perez, P., Schmid, C.: Aggregating Local Image Descriptors into Compact Codes. TPAMI (2012)
11. Ji, S., Xu, W., Yang, M., Yu, K.: 3D Convolutional Neural Networks for Human Action Recognition. TPAMI (2012)
12. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional Architecture for Fast Feature Embedding. arXiv preprint arXiv:1408.5093 (2014)
13. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale Video Classification with Convolutional Neural Networks. In: CVPR (2014)
14. Kong, N., Black, M.J.: Intrinsic Depth: Improving Depth Transfer with Intrinsic Images. In: ICCV (2015)
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. NIPS (2012)
16. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: HMDB: A Large Video Database for Human Motion Recognition. In: ICCV (2011)
17. Lan, Z., Lin, M., Li, X., Hauptmann, A.G., Raj, B.: Beyond Gaussian Pyramid: Multi-skip Feature Stacking for Action Recognition. In: CVPR (2015)
18. Lan, Z., Yu, S.I., Hauptmann, A.G.: Improving Human Activity Recognition Through Ranking and Re-ranking. arXiv preprint arXiv:1512.03740 (2015)
19. Li, W., Zhang, Z., Liu, Z.: Action Recognition Based on A Bag of 3D Points. In: CVPR (2010)
20. Liu, F., Shen, C., Lin, G.: Deep Convolutional Neural Fields for Depth Estimation from a Single Image. In: CVPR (2015)
21. Liu, L., Zhou, Y., Shao, L.: DAP3D-Net: Where, What and How Actions Occur in Videos? arXiv preprint arXiv:1602.03346 (2016)
22. Liu, M., Salzmann, M., He, X.: Structured Depth Prediction in Challenging Monocular Video Sequences. arXiv preprint arXiv:1511.06070 (2015)
23. Ma, S., Bargal, S.A., Zhang, J., Sigal, L., Sclaroff, S.: Do Less and Achieve More: Training CNNs for Action Recognition Utilizing Action Images from the Web. arXiv preprint arXiv:1512.07155 (2015)

24. Ng, J.Y.H., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici, G.: Beyond Short Snippets: Deep Networks for Video Classification. In: CVPR (2015)
25. Oneata, D., Verbeek, J., Schmid, C.: Action and Event Recognition with Fisher Vectors on a Compact Feature Set. In: ICCV (2013)
26. Oneata, D., Verbeek, J., Schmid, C.: The LEAR submission at THUMOS 2014 (2014)
27. Oreifej, O., Liu, Z.: HON4D: Histogram of Oriented 4D Normals for Activity Recognition from Depth Sequences. In: CVPR (2013)
28. Peng, X., Zou, C., Qiao, Y., Peng, Q.: Action Recognition with Stacked Fisher Vectors. In: ECCV (2014)
29. Raza, S.H., Javed, O., Das, A., Sawhney, H., Cheng, H., Essa, I.: Depth Extraction from Videos using Geometric Context and Occlusion Boundaries. In: BMVC (2014)
30. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. IJCV (2015)
31. Simonyan, K., Zisserman, A.: Two-Stream Convolutional Networks for Action Recognition in Videos. NIPS (2014)
32. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. In: ICLR (2015)
33. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A Dataset of 101 Human Action Classes From Videos in The Wild. In: CRCV-TR-12-01 (2012)
34. Sourimant, G.: A Simple and Efficient Way to Compute Depth Maps for Multi-View Videos. In: 3DTV-Conference (2010)
35. Srivastava, N., Mansimov, E., Salakhutdinov, R.: Unsupervised Learning of Video Representations using LSTMs. In: ICML (2015)
36. Sun, L., Jia, K., Yeung, D.Y., Shi, B.E.: Human Action Recognition using Factorized Spatio-Temporal Convolutional Networks. In: ICCV (2015)
37. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning Spatiotemporal Features with 3D Convolutional Networks. In: ICCV (2015)
38. Vieira, A.W., Nascimento, E.R., Oliveiraa, G.L., Liuc, Z., Campos, M.F.: On the Improvement of Human Action Recognition from Depth Map Sequences using SpaceTime Occupancy Patterns. Pattern Recognition Letters (2014)
39. Wang, H., Schmid, C.: Action Recognition with Improved Trajectories. In: ICCV (2013)
40. Wang, J., Liu, Z., Wu, Y.: Human Action Recognition with Depth Cameras. Springer International Publishing (2014)
41. Wang, J., Liu, Z., Chorowski, J., Chen, Z., Wu, Y.: Robust 3D Action Recognition with Random Occupancy Patterns. In: ECCV (2012)
42. Wang, J., Liu, Z., Wu, Y., Yuan, J.: Mining Actionlet Ensemble for Action Recognition with Depth Cameras. In: CVPR (2012)
43. Wang, L., Qiao, Y., Tang, X.: Action Recognition with Trajectory-Pooled Deep Convolutional Descriptors. In: CVPR (2015)
44. Wang, L., Xiong, Y., Wang, Z., Qiao, Y.: Towards Good Practices for Very Deep Two-Stream ConvNets. arXiv preprint arXiv:1507.02159 (2015)
45. Wang, P., Li, W., Gao, Z., Tang, C., Zhang, J., Ogunbona, P.: ConvNets-Based Action Recognition from Depth Maps Through Virtual Cameras and Pseudocoloring. In: ACM MM (2015)
46. Wang, X., Farhadi, A., Gupta, A.: Actions Transformations. In: CVPR (2016)

47. Wu, Z., Wang, X., Jiang, Y.G., Ye, H., Xue, X.: Modeling Spatial-Temporal Clues in a Hybrid Deep Learning Framework for Video Classification. In: ACM MM (2015)
48. Xu, Z., Yang, Y., Hauptmann, A.G.: A Discriminative CNN Video Representation for Event Detection. In: CVPR (2015)
49. Yang, X., , Zhang, C., Tian, Y.: Recognizing Actions using Depth Motion Maps-based Histograms of Oriented Gradients. In: ACM MM (2012)
50. Yang, X., Tian, Y.: Super Normal Vector for Activity Recognition using Depth Sequences. In: CVPR (2014)
51. Yu, G., Liu, Z., Yuan, J.: Discriminative Orderlet Mining for Real-time Recognition of Human-Object Interaction. In: ACCV (2014)
52. Zha, S., Luisier, F., Andrews, W., Srivastava, N., Salakhutdinov, R.: Exploiting Image-trained CNN Architectures for Unconstrained Video Classification. In: BMVC (2015)
53. Zhang, G., Jia, J., Wong, T.T., Bao, H.: Consistent Depth Maps Recovery from a Video Sequence. TPAMI (2009)
54. Zhao, S., Liu, Y., Han, Y., Hong, R.: Pooling the Convolutional Layers in Deep ConvNets for Action Recognition. arXiv preprint arXiv:1511.02126 (2015)