C3D: Generic Features for Video Analysis

Du Tran[1,2], Lubomir Bourdev[1], Rob Fergus[1,3], Lorenzo Torresani[2], Manohar Paluri[1]
[1]Facebook AI Research, [2]Dartmouth College, [3]New York University
{dutran,lorenzo}@cs.dartmouth.edu {lubomir,mano}@fb.com fergus@cs.nyu.edu

Abstract

Videos have become ubiquitous due to the ease of capturing and sharing via social platforms like Youtube, Facebook, Instagram, and others. The computer vision community has tried to tackle various video analysis problems independently. As a consequence, even though some really good hand-crafted features have been proposed there is a lack of generic feature for video analysis. On the other hand, the image domain has progressed rapidly by using features from deep convolutional networks. These deep features are proving to be generic and perform well on variety of image tasks. In this work we propose Convolution 3D(C3D) feature, a generic spatio-temporal feature obtained by training a deep 3-dimensional convolutional network on a large annotated video dataset comprising objects, scenes, actions, and other frequently occurring concepts. We show that by using spatio-temporal convolutions the trained features encapsulate appearance and motion cues and perform well on different discriminative video classification tasks. C3D has three main advantages. First, it is generic: achieving state of the art results on object recognition, scene classification, and action similarity labeling in videos. Second, it is compact: obtaining better accuracies than best hand-crafted features and best deep image features with a lower dimensional feature descriptor. Third, it is efficient to compute: 91 *times faster than current hand-crafted features, and two orders of magnitude faster than current deep-learning based video classification methods.*

1. Introduction

Multimedia on Internet is growing rapidly resulting in an exploding number of videos being shared every minute. Websites that were once text dominated have transformed themselves to become photo and video rich. To combat the information explosion it is essential to understand and analyze data for various purposes like search, recommendation, ranking etc. The computer vision community has been working on video analysis for decades and tackled different problems such as action recognition [31, 40, 2], abnormal event detection [3, 7, 55], and activity understanding [29]. Considerable progress has been made in these individual problems by employing various solutions. However, we lack a general way of representing videos and specifically lack a universal descriptor for a video. Such a general way of representing videos would be very helpful for solving various large-scale video tasks in a homogeneous way.

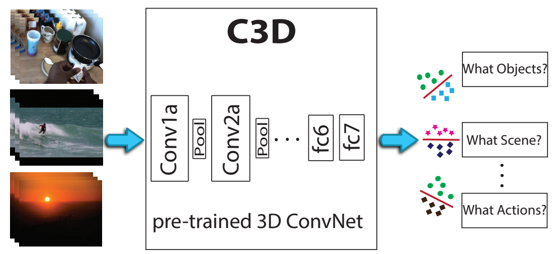

Figure 1. **C3D–generic video features**. C3D is trained using a deep spatio-temporal convolutional network and can be used as a feature extractor for new tasks. It performs well on various video classification problems such as object recognition, scene classification, action similarity labeling, and action recognition.

A universal descriptor should have at least three important properties to be useful at Internet scale. First, this descriptor needs to be **generic**, so that it can represent different types of videos well. For example, Internet videos can be of landscapes, natural scenes, sports, TV shows, movies, or even pets, food and so on. Second, the desired descriptor needs to be **compact**, as we are scaling up to millions of videos, being compact is very helpful for processing, storage, and retrieval. Third, it needs to be **efficient** to compute the feature, as many videos are expected to be processed every day, hour, minute or even seconds in real world systems.

We take motivation from the deep learning breakthroughs in the image domain [30] where rapid progress has been made in the past two years in feature learning. Inspired by the success of deep networks various pre-trained convolutional network(ConvNet) models [21] are commonly used for extracting image features. These features are normally the activations of the network's last few fully connected layers which are showed to be useful for transfer learning tasks [54, 22, 56]. However, the video domain is lacking such generic features due to two main reasons: first, there is no large-scale supervised video dataset that encompasses diverse and generic concepts and second, there is no efficient way to learn compact spatio-temporal features that encapsulate both appearance and motion.

This paper addresses both of the above-mentioned problems, making it possible to learn generic video features. We first build a large manually annotated dataset that contains videos of various diverse concepts. We use this dataset to train a deep 3D ConvNet for learning video features. While most of the recent ConvNet based video classification approaches use 2D convolutions [23, 43], we argue that 3D convolutions(applied both spatially and temporally) can better model spatio-temporal information, thus leading to better features. It is worth noting that 3D ConvNets were proposed before specifically for action recognition setting [20], while we use them for feature learning.

We train the deepest 3D ConvNet model that has ever been proposed and show that it outperforms 2D ConvNet models and hand-crafted features by a good margin both qualitatively and quantitatively. A typical use case of C3D is highlighted in figure 1. We test our pre-trained model on the benchmark datasets and show its transfer learning capability. Our features have the desirable properties that a universal descriptor should have: it is generic, compact, and efficient. To summarize, our contributions in this paper are:

- We propose an approach for generic spatio-temporal feature learning based on the right choice of dataset design and an appropriate learning model using 3D ConvNet.
- We show that with our generic features a simple linear model can either achieve or approach state-of-the-art performance on different video classification benchmarks.
- We show that our features are compact, discriminative and orders of magnitude faster to compute compared with current best hand crafted features [52] and current best deep learning model [43].

2. Related Work

Videos have been studied by computer vision community for decades. Over the years various problems like action recognition [31, 40, 2], anomaly detection [24], video retrieval [1] and many more have been proposed. Many of these works are about video representations. Laptev and Lindeberg [31] proposed spatio-temporal interest points (STIPs) by extending Harris corner detectors to 3D. Histogram of Oriented Gradients (HOG) [8] and Histogram of Optical Flows (HOF) [9] are normally extracted at these STIPs and used as video features. SIFT and HOG are also extended into 3D into SIFT-3D [41] and HOG3D [25] for action recognition. Dollar *et al.* proposed Cuboids features for behavior recognition [11]. Recently, Wang *et al.* proposed dense trajectories [50, 51] and later proposed an improved version [52] which is currently the state-of-the-art hand-crafted feature. The improved dense trajectory work is an interesting example showing that temporal signals could be handled differently from that of spatial signal. Instead of extending Harris corner detector into 3D, it starts with 2D Harris to detect corners in video frames and use optical flow to track them. Different hand-crafted features are extracted along the trajectory. Despite its good performance, this method is computationally intensive and becomes intractable on large-scale datasets.

With recent availability of powerful parallel machines (GPUs, CPU clusters), together with large amounts of training data, convolutional neural networks [33] have come back providing breakthroughs on many AI problems, related to text [6, 34], speech recognition [36, 35] and image based problems [30, 15]. ConvNets have been applied to the problem of human pose estimation in both images [17] and videos [18]. More interestingly they are used for image feature learning [12]. Similarly, Zhou *et al.* proposed to learn scene-specific features using deep ConvNets [56]. Deep learning has also been applied to video feature learning in an unsuperivsed setting [32, 5]. In Le *et al.* [32], the authors use stacked ISA to learn spatio-temporal features for videos. Although this method showed good results on action recognition, it is still computationally intensive at training and hard to scale up for testing on large datasets. 3D ConvNets were proposed for human action recognition [20, 46] and for medical image segmentation [48, 19]. Recently, Karpathy *et al.* [23] trained deep networks on a large video dataset for video classification. Simonyan and Zisserman [43] used two stream networks to achieve best results on action recognition.

Among these approaches, the 3D ConvNets approach in [20] is most closely related to us. However, their work is designed for action classification which is very task-specific. In fact, this method employed human detector and head tracking to track the human subjects. The tracked human subjects are segmented out and given as input to the 3D ConvNet for human action classification. In contrary, our method takes full video frames as inputs and does not rely on any preprocessing, making it easily applicable to larger scale and more generic video analysis tasks. In our method we use 3D ConvNet as a method for feature learning and later use the trained model as a feature extractor. Moreover, our network is much deeper, we use 8 convolution layers compared to their 3. Our proposed method also shares some similarities with Karpathy *et al.* [23] and Simonyan and Zisserman [43] in terms of using full frames for training the ConvNet. However, their works are built on using only 2D convolution and 2D pooling operations, which we feel is not the ideal way to handle temporal signals. Figure 3 contrasts the difference between 2D convolution on a single frame, 2D convolution on multiple frames and 3D convolution on multiple frames. 3D convolution preserves temporal information and passes it to the next layer whereas 2D convolution completely collapses it. Our model performs 3D convolutions and 3D pooling propagating temporal information throughout the network and learning temporal filters in all convolution layers.

3. Learning Spatiotemporal Features

We present how to train deep convolutional networks (ConvNets) for feature learning purposes by designing a good video dataset and suitable networks.

3.1. Dataset Design

Our goal is to learn a generic feature descriptor for videos. At a higher level, the feature needs to encode appearance and motion in a combined fashion. At a more finer level, it should encode information of objects occurring, actions taking place, or the scene within the representing video. Using a dataset with a diverse distribution of concepts, we can learn a more generic feature descriptor for videos. Our dataset comprises of more than $380,000$ videos about 382 concepts. Each video is typically 5-15 seconds long. We start with a list of frequently used video hashtags from a popular social network as our set of initial concepts. We manually prune all non-visual tags(ex: love, beautiful etc.) and for each remaining tag we download 5000 videos. As expected the hashtags are very noisy, we use human annotators to label each video. The annotators play the entire video and mark it positive if the respective concept appears in the video. At the end of the annotation process we have a single class label for each video. We further prune all concepts with fewer than 1000 positive annotations. The whole process ensures that our concepts approximately capture the distribution of internet videos(as we start from social network videos and prune infrequent and non visual concepts). We show the distribution of the final concepts in our dataset in figure 2. The most popular concepts are objects, actions and scene covering 72% of the dataset followed by food and animal.

3.2. Learning Spatiotemporal Features with 3D-Convolutional Neural Networks

We propose to use 3D convolutional network [20] for learning spatiotemporal features. More specific, we train a deep 3D convolutional neural network to classify actions, objects, scenes, and other concepts using our dataset described in section 3.1. The trained network is then used as a feature extractor for other video analysis tasks. As our

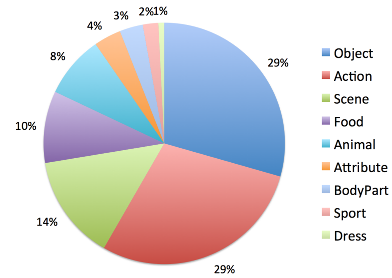

Figure 2. **Our video dataset concept distribution**. Major categories are actions, objects, and scenes covering 72% of the dataset categories. The second column of table 2 gives the exact number of concepts for each category.

dataset covers a large and diverse set of concepts, the trained network is forced to learn generic features which are very useful for various video classification tasks.

Network operations: We argue that 3D ConvNet is well-suited for spatiotemporal feature learning. Compared with 2D ConvNet, 3D ConvNet models temporal information better via 3D convolution and 3D pooling operations. The main difference between 3D ConvNets and 2D ConvNets is that convolutions and pooling operations are perform spatial-temporally while those operations on 2D ConvNets are done only spatially. Figure 3 illustrates the difference between the two convolutions, 2D convolution applied on an image will output an image, 2D convolution on multiple images(treating them as different channels [43]) also results in an output image. Only 3D convolution preserves the temporal information of the input signals. The same phenomena is applicable for 2D and 3D polling. This difference makes 2D ConvNets lose temporal information of the input signal right after every convolution operation. In [43], although the temporal stream network takes multiple-frame inputs, because of the 2D convolutions, after the first convolution layer, temporal information is collapsed completely. Similarly, fusion models in [23] used 2D convolutions, most of the networks lose their input's temporal signal just right after the first convolution layer. The *Slow Fusion* model in [23] can handle temporal information better as they gradually group the signals, it still loses all temporal information after the third convolution layer because of using 2D operations. We believe this is the key reason *Slow Fusion* performs best among all networks studied in [23].

Network's architecture: We choose the best known network architectures for 2D ConvNets for images and extend that to 3D for videos. We believe that a better network architecture can further improve the performance of concept classification (training task) and also improve the output learned features (transferring task), however finding the best architecture is beyond the scope of this work. Inspired by the success of the very deep networks with small receptive fields [44], we design our 3D ConvNet similar to the

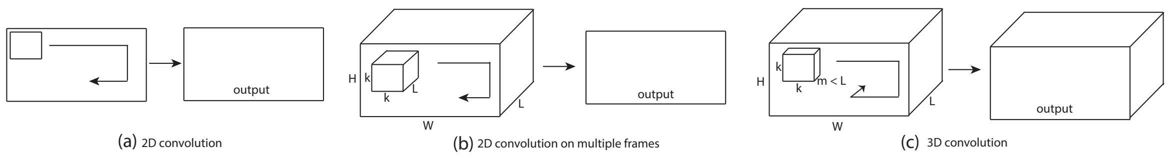

Figure 3. **Convolution operations on images and videos**. a) Applying 2D convolution on an image results in an image. b) Applying 2D convolution on multiple frames (as multiple channels) also results in an image. c) Applying 3D convolution on a video volume results in another volume, preserving temporal information of the input signal.

Net A architecture in [44] by replacing all 2D convolution and pooling operations by 3D operations. The architecture of our 3D ConvNet is presented in figure 4. All of our 3D convolution filters are $3 \times 3 \times 3$ (reading as length $\times$ height $\times$ width) with stride (1 in both space and time). All 3D pooling layers are $2 \times 2 \times 2$ (except for `pool1`) with stride 1. Only `pool1` is of $1 \times 2 \times 2$ with intention of preserving the temporal information at early phase as allowing pooling can blur early motion signals.

Training: Our dataset is randomly split into train, validation, and test sets with proportions of 70%, 10%, and 20%, respectively. We train our $3D$ ConvNet(C3D) with inputs of size 16-frames. The clips are densely sampled from training video with temporal stride of 32 frames. The frames are down-scaled to 128×128 making the input of the network $3 \times 16 \times 128 \times 128$(3 color channels, 16 frames in time and 128×128 frames each). All convolution layers are randomly initialized using normal distribution with standard deviation of 0.01. Similar to [44], we do not use contrast normalization layers. We use mini-batch size of 30. The initial learning rate is 0.003 and is divided by 10 after every 200K iterations. Training is stopped after 600K iterations. On a single Nvidia K40 GPU it takes 4 weeks to train our model. It is worth noting that, although we have multiple sub-categories (e.g. actions, objects, scenes), we train our network with all concepts in a single softmax layer (see figure 4).

We also train two 2D ConvNets on our dataset from scratch and use them as baselines along with the publicly available imagenet model [21] . The two 2D ConvNets are Krizhevsky's architecture [30] and Net A from [44] referred to as KNet and VGGA respectively from now on. Our intention in training these baselines is to understand the impacts of 3D convolutions over 2D convolutions, network architecture, and generalizing capability of features trained on image versus video datasets.

Concept training results: We evaluate C3D features and the other two baseline features(KNet, VGGA) on the test split and report the results in table 1. We report both clip and video level accuracy at top 1 and top 5. For video level accuracy, we do a simple averaging of the clip level decisions to get the video level prediction. At the clip level, our C3D outperforms KNet and VGGA by 4.46% and 1.61% respectively.

Model	Clip@1	Video@1	Video@5
KNet	16.2	21.5	47.6
VGGA	19.1	25.9	54.2
Our C3D	**20.7**	**27.1**	**55.1**

Table 1. **Video classification result on our dataset**. C3D performs 4.46% and 1.61% better than KNet and VGGA respectively. Please note that random chance is of 0.26%.

Sub-cat	# class	VGGA	C3D	Δ
Object	113	19.22	20.62	**+1.40**
Action	111	19.14	21.66	**+2.52**
Scene	54	19.79	20.97	**+1.18**
Food	37	13.72	13.14	-0.57
Animal	32	22.03	22.93	**+0.90**
Body-part	14	7.74	8.24	**+0.50**
Attribute	12	9.62	9.70	+0.08
Sports	8	35.61	36.37	**+0.76**
Dress	3	15.98	18.94	**+2.96**

Table 2. **Sub-category classification results on our dataset**. C3D outperforms VGGA on most of sub-categories except for food because this category focuses more on texture features.

We provide detailed classification results for both C3D and VGGA for every sub-category in table 2. C3D outperforms VGGA for all the sub-categories except 'food'. This phenomena is probably due to the food category tending to be more texture related which 2D networks favour more. It is worth noting that random chance is of 0.26%.

4. C3D: New Generic Feature for Videos

We show how C3D can be used as generic features for different video classification tasks with state-of-the-art results. An important point to note is we do not fine-tune our model for any dataset or application. We use the model trained on our dataset as is for all of the results below.

4.1. Application 1: Action recognition

Dataset: We evaluate our C3D features on UCF101 dataset [45]. The dataset consists of 13320 videos of 101 human action categories. We use the three split setting provided with this dataset.

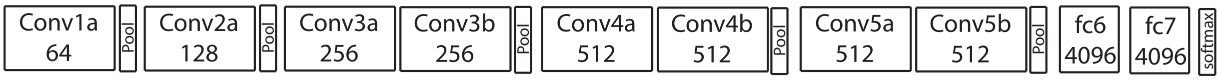

Figure 4. **Our 3D ConvNet**. All 3D convolution layers are $3 \times 3 \times 3$ with stride 1 in both spatial and temporal dimensions. Number of filters are denoted in each box. The 3D pooling layers are denoted from `pool1` to `pool5`. All pooling are $2 \times 2 \times 2$, except for `pool1` is $1 \times 2 \times 2$ with stride $1 \times 1 \times 1$. Fully connected layers `fc6` and `fc7` have 4096 outputs.

Classification model: Because our pre-trained 3D ConvNet takes clip as inputs, we sample the videos densely and extract 16-frame clips with a stride 16, e.g. no overlapping between two consecutive clips. Each clip is passed into our network to extract features. We use activations of the last four layers: `pool5`, `fc6`, `pc7`, and `prob` as features. `pool5` is the output of the last pooling layer, `fc6` and `fc7` are the output activations of two fully connected layers. Finally, `prob` is output activations of the softmax layer. Each of these four is considered a single feature channel. To compute the descriptor for a video, we average its clip features. The averaged feature vector is then L2-normalized and given as input to a multi-class linear SVM for training models.

Baselines: We compare C3D features with a few baselines. The first set of baseline come from the current state-of-the-art hand crafted features, namely improved dense trajectories(iDT) [52]. This method uses optical flows (after motion compensated) to track 2D Harris corners over 15 frames to construct trajectories. Along each trajectory, it extracts HOG [8], HOF [9], motion boundary histories (MBHx, MBHy), and the trajectory displacement vector. For each feature type, we use k-means to build a codebook size 5000. A video is then represented as a histogram of words. The histogram is L2-normalized and passed to a linear SVM for evaluations. The second set of baselines are current deep features. We use Imagenet pre-trained features [12] provided in [21] which is a 2D ConvNet trained on Imagenet ILSVRC12 dataset [39], referred to as Imagenet from now on. We also evaluate two other 2D ConvNets KNet and VGGA which are trained on our dataset(described in section 3.2) as baselines.

Results: Table 3 presents accuracy of action recognition on UCF101 using linear SVM for all single features. Our C3D `pool5` feature is the best single feature achieving 72.26%. The second best feature is also from C3D(`fc6`). Surprisingly, even the C3D `prob` feature with only 382 dimensions achieves 57.91%, this is a very promising compact and discriminative representation for action recognition in videos. Our C3D `pool5` outperforms improved dense trajectories MBHy by 11.91%. Compared to other 2D deep features, C3D outperforms 4-10% owing to the better temporal modeling in training which leads to better features.

iDT [52]					
	Traj	HOG	HOF	MBHx	MBHy
Dim	5000	5000	5000	5000	5000
Acc	52.74	50.94	59.34	59.67	60.69
Imagenet [12]					
		`prob`	`fc7`	`fc6`	`pool5`
Dim		1000	4096	4096	9216
Acc		43.78	66.61	68.78	66.58
KNet					
		`prob`	`fc7`	`fc6`	`pool5`
Dim		382	4096	4096	9216
Acc		43.10	58.15	59.89	61.97
VGGA					
		`prob`	`fc7`	`fc6`	`pool5`
Dim		382	4096	4096	25088
Acc		53.96	67.33	68.68	68.41
C3D					
		`prob`	`fc7`	`fc6`	`pool5`
Dim		382	4096	4096	8192
Acc		57.91	69.23	71.31	**72.26**

Table 3. **Action classification with single features on UCF101**. Our C3D features outperform all other features by a large margin. C3D outperforms iDT [52] by 11.91%, and Imagenet, KNet, and VGGA features by 4-10%.

Method	Dim	Accuracy
improved Dense Traj (iDT)	25,000	76.2
Imagenet(`fc6`)+C3D(`fc6`)	8192	76.4
iDT+Imagenet(`fc6`)+C3D(`fc6`)	33192	86.7
Deep networks [23]	4096	65.4
Appearance stream network [43]	2048	72.6
Two-stream networks [43]	4096	87.6
Fisher vector [37]	102400	**87.9**

Table 4. **Action recognition results on UCF101**. Comparison with current methods . Upper table presents classification results using different combinations of our features with a linear SVM. Lower table presents results of current methods.

In table 4, we present classification results of different feature combinations using linear SVM. The lower part of the table presents the results of current best methods.

In feature combination, we concatenate the descriptor and pass them to a linear SVM for evaluations. Combining features from the same network normally yield only 1-1.5% indicate that the features from the same network are not much complementary. However, the features from improved dense trajectories (iDT) [52] are highly complementary to each other. Combining all 5 features of iDT (making an 25000 dimensions) boost performance to 76.2%. We observe similar complementary nature with the deep features, C3D is quite complementary to the other baseline 2D ConvNet features. We did not study all combinations, but rather try to combine C3D and each 2D deep feature within the same channel (e.g. `fc6`). The best combination is C3D `fc6` with Imagenet `fc6` which achieves 76.4% with only 8192 dimensions. Interestingly, if we concatenate these 8192 dimensional vector with the 25000-dim of iDT, we can approach 86.7% which is only 1.2% below state of the art method [37] which has 102400 of Fisher vector encoding of iDT. We note that iDT is computationally expensive compared to our C3D (see section 5).

Compared with deep learning based approach, the authors of [23] trained a deep network on millions of sport videos with fine-tuning on UCF101, they achieve 65.4%, while our single C3D `pool5` feature with linear SVM gets 72.26%, and our combined deep features from both Imagenet and C3D obtains 76.4% (11% improvement over [23]) without any fine-tuning. This improvement can be attributed to two factors: first, our network can model better temporal signals and second our dataset captures more generic concepts. The method in [43] used two separate networks, one is trained on images(called appearance stream) and the other is trained on optical flows(called temporal stream). Their appearance stream gets 72.6% which is comparable to our C3D `fc6` features, and 3.6% lower than C3D fc6 with Imagenet fc6. Their two-stream networks achieves 87.6%, 0.9% better than our C3D combined with Imagenet and iDT. It is worth noting that our model is very simple and our features are also compact, fast to compute (see section 5) so there is still a considerable advantage to C3D features when operating on large scale datasets.

4.2. Application 2: Action similarity labeling

Dataset. The ASLAN dataset consists of 3631 videos from 432 action classes. The task is to predict if a given pair of videos belong to the same or different action. We use the prescribed 10-fold cross validation with the splits provided with the dataset. This problem is different from action recognition, as the task focuses on predicting action similarity not the actual action label. The task is quite challenging because the test set contains videos of "*never-seen-before*" actions.

Features. We densely sample videos into 16-frame clips with a stride of 8 (overlap of 8 frames). We extract C3D features: `prob`, `fc7`, `fc6`, `pool5` for each clip. The features for videos are computed by averaging the clip features separately for each type of feature, followed by an L2 normalization.

Method	Features	Model	Acc.	AUC
[27]	STIP	linear	60.9	65.3
[28]	STIP	metric	64.3	69.1
[26]	MIP	metric	65.5	71.9
[16]	MIP+STIP+MBH	metric	66.1	73.2
[53]	iDT+FV	metric	68.7	75.4
Ours	C3D	linear	**72.9**	**79.8**

Table 5. **Action similarity labeling result**. C3D improves state-of-the-art method by 4.2% in accuracy and by 4.4% in area under ROC curve.

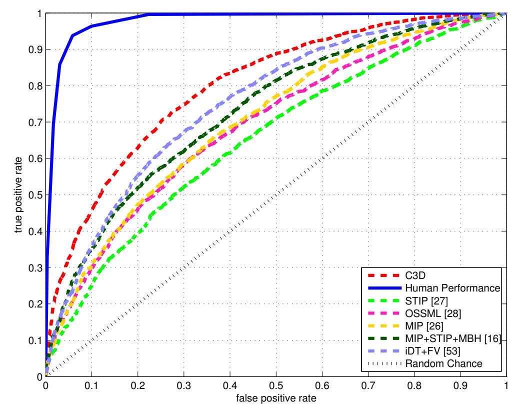

Figure 5. **Action similarity labeling result**. ROC curve of C3D evaluated on ASLAN. C3D achieves 79.8% on AUC and outperforms current state-of-the-art by 4.4%.

Classification model. We follow the same setup used in [27]. Given a pair of videos, we compute the 12 different distances provided in [27]. With 4 types of features, we obtain 48-dimensional ($12 \times 4 = 48$) feature vector for each video pair. As these 48 distances are not comparable to each other, we normalize them independently such that each dimension has zero mean and unit variance. Finally, a linear SVM is trained to classify video pairs into same or different on these 48-dim feature vectors.

Results. Table 5 reports the result of our C3D compared with state-of-the-art methods. While most current methods use multiple hand-crafted features, strong encoding methods (VLAD, Fisher Vector), and complex learning models, our method uses a simple averaging of our C3D features over the video and a simple *linear* SVM. C3D outperforms state-of-the-art method by 4.2% on accuracy and 4.4% on area under ROC curve(AUC). Figure 5 plots the ROC curves of C3D compared with current methods and human performance [27].

Dataset	[10]	[47]	[13]	[14]	**C3D**
Maryland	43.08	74.6	67.69	**77.69**	**77.69**
YUPENN	80.71	85.0	85.95	96.19	**96.67**

Table 6. **Scene recognition compared with state-of-the-art methods**. C3D achieves state-of-the-art results on Maryland and YUPENN datasets with a simple linear SVM.

4.3. Application 3: Dynamic scene recognition

Datasets. We evaluate C3D features on two scene recognition benchmarks: YUPENN [10] and Maryland [42]. YUPENN consists of 420 videos of 14 scene categories and Maryland has 130 videos of 13 scene categories.

Classification model. We use the same setup of feature extraction and linear SVM for classification as described in the other applications. We follow the same leave-one-out evaluation protocol as described by the authors of the datasets and compare with current methods.

Results. Table 6 reports our C3D results using `fc6` feature and compares them with the current best methods. C3D achieves state-of-the-art performance on both datasets using only a *linear* SVM with simple averaging of clip features while the second best method [14] uses kernel SVM with different *complex* feature encoding (FV, LLC).

4.4. Application 4: Object recognition in videos

Dataset: We evaluate our C3D on egocentric object recognition dataset [38]. This dataset consists of 10 long video sequences capturing 42 types of everyday objects such as milk bottles, lunch boxes, staplers, etc. The total number of frames is $105,627$. Each frame is manually annotated with an object label. A point to note, this dataset is egocentric and all videos are recorded in a first person view which have quite different appearance and motion characteristics than any of the videos we have in the training dataset.

Evaluation setup: Although this is a video set, the standard evaluation used in this dataset is based on frames. On the other hand, our features are applied on only video chunks (e.g. 16 frame chunk). We slide a window of 16 frames over all videos to extract our C3D features. We choose the ground truth label for each chunk to be the most frequently occurring label of the chunk. If the most frequent label in a chunk occurs fewer than 8(50% of the time), we consider it as negative chunk with no object and disregard it in both training and testing. We train and test our C3D features using linear SVM and report the object recognition accuracy. We follow the same split provided in [38].

Results: Table 7 presents classification accuracy of our C3D features compared with current state of the art method [38] to our knowledge. Most of our C3D features outperform this method and our combination of `fc6` and `pool5` outperforms it by 3.3% with only linear model where the comparing method used RBF-kernel on strong SIFT-RANSAC feature matching. We note that the authors in [38] show that further enforce temporal smoothness over the long videos can improve the performance of object recognition. However, investigating in good models for egocentric object recognition is beyond the scope of this study as we are more interested in studying the general applicability of C3D features.

Method	[38]	**Our C3D**			
Feature	SIFT	`fc7`	`fc6`	`pool5`	`fc6+pool5`
Model	kernel	linear	linear	linear	linear
Acc.	12.0	12.9	13.8	15.0	**15.3**

Table 7. **Object recognition on Egocentric Dataset**. Most of our C3D features outperform [38]. Our combination of `fc6` and `pool5` from C3D outperforms [38] by 3.3%.

5. Compactness and Efficiency

C3D is compact: In order to evaluate the compactness of C3D features we use PCA to project the features into lower dimensions and report the classification accuracy of the projected features on UCF101 [45]. We apply the same process with the current best hand-crafted features [52] as well as the current deep features for images [12] and compare the results in Figure 6. At the extreme setting with only 10 dimensions, C3D accuracy is 45.4% which is 10% better than accuracy of Imagenet, and 19% better than iDT. At 50 dim, C3D is also about 5-10% better than Imagenet and approximately 20% better than iDT. Finally, with $100-200$ dimensions, our C3D is able to gets 70% accuracy indicating our features are both compact and discriminative. This is very helpful for large-scale retrieval applications where low storage cost and fast retrieval are crucial.

C3D is efficient. We do a runtime analysis of our proposed C3D and compare it with current best hand-crafted features [52] and current best deep learning model for action recognition [43]. For iDT, we use the code kindly provided by the authors [52]. For [43], there is no public pretrained model or wrapper which we can evaluate. However, this method uses Brox's optical flows [4] as low level input signals. We manage to evaluate runtime of Brox's method using two different versions: CPU implementation provided by the authors [4] and the GPU implementation provided in OpenCV. We note that, the runtime of Simonyan and Zisserman [43] is greater than that of [4].

We extract features using iDT and our C3D features as well as compute Brox's optical flows for the whole UCF101 dataset. We report runtime of different methods in table 8 evaluated using a single CPU (Brox's CPU implementation and iDT) or a single K40 Tesla GPU (Brox's GPU imple-

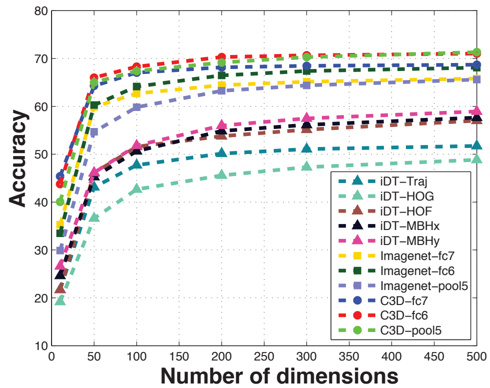

Figure 6. **Comparison of feature accuracy following PCA dimensionality reduction on UCF101**. C3D achieves 70.3% accuracy using only 200 dimensions. This outperforms Imagenet and iDT (on the same dimension) by 5-10% and 15-20% respectively.

Method Usage	iDT CPU	Brox's GPU	Brox's CPU	C3D GPU
RT(in hours)	202.2	607.8	2513.9	2.2
x Slower	91.4	274.6	1135.9	1

Table 8. **Runtime evaluations on UCF101**. Our C3D is 91x faster than improved dense trajectories [52] and 274x time faster than Brox's optical flow methods, thus relatively more than two orders of magnitude faster than [43].

mentation and our C3D). It's worth nothing that it is not a fair comparison for iDT as they only use CPU. We cannot find any GPU implementation of this method and it is not trivial to implement a parallel verison of this algorithm on GPU as it involves complex modules like tracking, motion compensation, and different feature encoding methods which might not be conducive to GPU optimizations.

Comparing our feature extraction method with Brox's optical flow using the same K40 GPU, it takes 2 hours and 12 minutes to extract our features for the whole UCF101 dataset, while it take up to 607 hours to compute Brox's flows for the whole frames of UCF101. The method in [43] used Brox's flows as input signals, thus it will take longer for them to classify videos or extract video features. As we mentioned, there is no GPU implementation of iDT [52], here we give a rough comparison with this method. Extracting iDT features for the whole UCF101 dataset on a CPU will take 202 hours and 10 minutes which is roughly 91 times slower than our feature extraction.

6. Qualitative Visualization

We qualitatively evaluate our learned C3D features to verify if it is a good generic feature for video by visualizing the learned feature embedding on another dataset. We randomly select 100K clips from UCF101, then extract `pool5` features for those clips using for features from Imagenet and C3D. These features are then projected to 2-dimensional space using t-SNE [49]. Figure 7 visualizes the feature embedding of the features from Imagenet and our C3D on UCF101. It is worth noting that, we did not do any fine-tuning as we wanted to verify if the features show good generalization capability across datasets. We quantitatively observe that C3D is better than Imagenet.

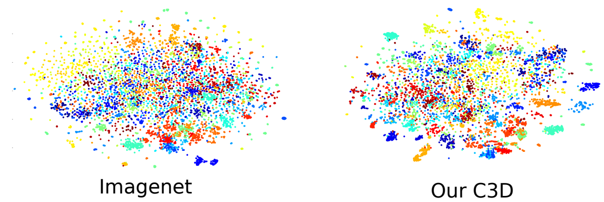

Figure 7. **Feature embedding**. Feature embedding visualizations of Imagenet and C3D on UCF101 dataset using t-SNE [49]. C3D's features are semantically separable compared to the Imagenet feature embedding suggesting that it is a good generic feature for videos. Best view in color.

7. Conclusions

Feature learning plays a key role in solving many machine leaning problems across different domains, e.g text, speech, images and videos. With discriminative, compact, and fast-to-compute features, one can aspire to solve various problems using a simple linear model. The video domain has been lagging behind in this aspect and in this work we try to address the problem of learning generic features for videos. Our findings in this study suggest that the combination of the right dataset design, strong network architecture, and good temporal modeling is crucial to learn generic features for videos. These findings are consistent with the discussions in [56] where they found appropriate choices of training dataset gives better features for scene classification in still images. Our approach for learning generic video features uses deep 3D convolutional networks trained on a large-scale manually annotated video dataset. We show that the learned features are not only discriminative but also compact and efficient to compute. Our work has dual impacts on the video domain. On one hand, the discriminative power of the features makes them a good alternative to existing features for video problems. On the other hand, it opens up opportunities for large-scale video analysis as C3D is compact and efficient to compute. We demonstrated that our features can achieve or approach state-of-the-art results on different video tasks with a simple linear model in a cheap and low-dimensional feature space.

We will make our code and the pre-trained model available upon publication acceptance.

References

[1] M. Bendersky, L. Garcia-Pueyo, J. Harmsen, V. Josifovski, and D. Lepikhin. Up next: retrieval methods for large scale related video suggestion. In *ACM SIGKDD*, pages 1769–1778, 2014. 2

[2] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, pages 1395–1402, 2005. 1, 2

[3] O. Boiman and M. Irani. Detecting irregularities in images and in video. *IJCV*, 2007. 1

[4] T. Brox and J. Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE TPAMI*, 33(3):500–513, 2011. 7

[5] C. F. Cadieu and B. A. Olshausen. Learning intermediate-level representations of form and motion from natural movies. *Neural computation*, 24(4):827–866, 2012. 2

[6] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, 2008. 2

[7] Y. Cong, J. Yuan, and J. Liu. Sparse reconstruction cost for abnormal event detection. In *CVPR*, 2011. 1

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

[9] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, 2006. 2, 5

[10] K. Derpanis, M. Lecce, K. Daniilidis, and R. Wildes. Dynamic scene understanding: The role of orientation features in space and time in scene classification. In *CVPR*, 2012. 7

[11] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Proc. ICCV VS-PETS*, pages 65–72, 2005. 2

[12] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2013. 2, 5, 7

[13] C. Feichtenhofer, A. Pinz, and R. P. Wildes. Space-time forests with complementary features for dynamic scene recognition. In *BMVC*, 2013. 7

[14] C. Feichtenhofer, A. Pinz, and R. P. Wildes. Bags of space-time energies for dynamic scene recognition. In *CVPR*, 2014. 7

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv preprint arXiv:1311.2524*, 2013. 2

[16] Y. Hanani, N. Levy, and L. Wolf. Evaluating new variants of motion interchange patterns. In *CVPR workshop on action similarity in unconstrained video*, 2013. 6

[17] A. Jain, J. Tompson, M. Andriluka, G. W. Taylor, and C. Bregler. Learning human pose estimation features with convolutional networks. In *ICLR*, 2014. 2

[18] A. Jain, J. Tompson, Y. LeCun, and C. Bregler. Modeep: A deep learning framework using motion features for human pose estimation. In *ACCV*, 2014. 2

[19] V. Jain, B. Bollmann, M. Richardson, D. R. Berger, M. N. Helmstaedter, K. L. Briggman, W. Denk, J. B. Bowden, J. M. Mendenhall, W. C. Abraham, et al. Boundary learning by optimization with topological constraints. In *CVPR*, pages 2488–2495. IEEE, 2010. 2

[20] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE TPAMI*, 35(1):221–231, 2013. 2, 3

[21] 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. 1, 4, 5

[22] S. Karayev, M. Trentacostea, H. Han, A. Agarwala, T. Darrell, A. Hertzmann, and H. Winnemoeller. Recognizing image style. In *BMVC*, 2014. 2

[23] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 2, 3, 5, 6

[24] J. Kim and K. Grauman. Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates. In *CVPR*, pages 2921–2928, 2009. 2

[25] A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008. 2

[26] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf. Motion interchange patterns for action recognition in unconstrained videos. In *ECCV*, 2012. 6

[27] O. Kliper-Gross, T. Hassner, and L. Wolf. The action similarity labeling challenge. *TPAMI*, 2012. 6

[28] O. Kliper-Grossa, T. Hassner, and L. Wolf. The one shot similarity metric learning for action recognition. In *International Workshop on Similarity-Based Pattern Analysis and Recognition*, 2011. 6

[29] D. B. Kris M. Kitani, Brian D. Ziebart and M. Hebert. Activity forecasting. In *ECCV*, 2012. 1

[30] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 2, 4

[31] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, 2003. 1, 2

[32] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, 2011. 2

[33] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time-series. *The Handbook of Brain Theory and Neural Networks*, 1995. 2

[34] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 2

[35] A. Mohamed, T. Sainath, G. Dahl, B. Ramabhadran, G. Hinton, and M. Picheny. Deep belief networks using discriminative features for phone recognition. In *ICASSP*, 2011. 2

[36] A. Mohamed, D. Yu, and L. Deng. Investigation of full-sequence training of deep belief networks for speech recognition. In *INTERSPEECH*, 2010. 2

[37] X. Peng, L. Wang, X. Wang, and Y. Qiao. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *CoRR*, abs/1405.4506, 2014. 5, 6

[38] X. Ren and M. Philipose. Egocentric recognition of handled objects: Benchmark and analysis. In *the First Workshop on Egocentric Vision*, 2009. 7

[39] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge, 2012. 5

[40] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *ICPR*, 2004. 1, 2

[41] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proc. ACM Multimedia*, pages 357–360, 2007. 2

[42] N. Shroff, P. K. Turaga, and R. Chellappa. Moving vistas: Exploiting motion for describing scenes. In *CVPR*, 2010. 7

[43] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 2, 3, 5, 6, 7, 8

[44] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 3, 4

[45] 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, 7

[46] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *ECCV*, pages 140–153. Springer, 2010. 2

[47] C. Theriault, N. Thome, and M. Cord. Dynamic scene classification: Learning motion descriptors with slow features analysis. In *CVPR*, 2013. 7

[48] S. C. Turaga, J. F. Murray, V. Jain, F. Roth, M. Helmstaedter, K. Briggman, W. Denk, and H. S. Seung. Convolutional networks can learn to generate affinity graphs for image segmentation. *Neural Computation*, 22(2):511–538, 2010. 2

[49] L. van der Maaten and G. Hinton. Visualizing data using t-sne. *JMLR*, 9(2579-2605):85, 2008. 8

[50] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action Recognition by Dense Trajectories. In *CVPR*, pages 3169–3176, June 2011. 2

[51] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *IJCV*, 103(1):60–79, 2013. 2

[52] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. 2, 5, 6, 7, 8

[53] Q. P. X. Peng, Y. Qiao and Q. Wang. Large margin dimensionality reduction for action similarity labeling. *IEEE Signal Processing Letter*, 2014. 6

[54] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev. Panda: Pose aligned networks for deep attribute modeling. In *CVPR*, 2014. 2

[55] B. Zhao, L. Fei-Fei, , and E. Xing. Online detection of unusual events in videos via dynamic sparse coding. In *CVPR*, 2011. 1

[56] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014. 2, 8