Action Recognition in Videos

Presenter: Haochen Li

Overview

- Large-scale Video Classification with Convolutional Neural Networks
- Two-Stream Convolutional Networks for Action Recognition in Videos

Large-scale Video Classification with Convolutional Neural Networks

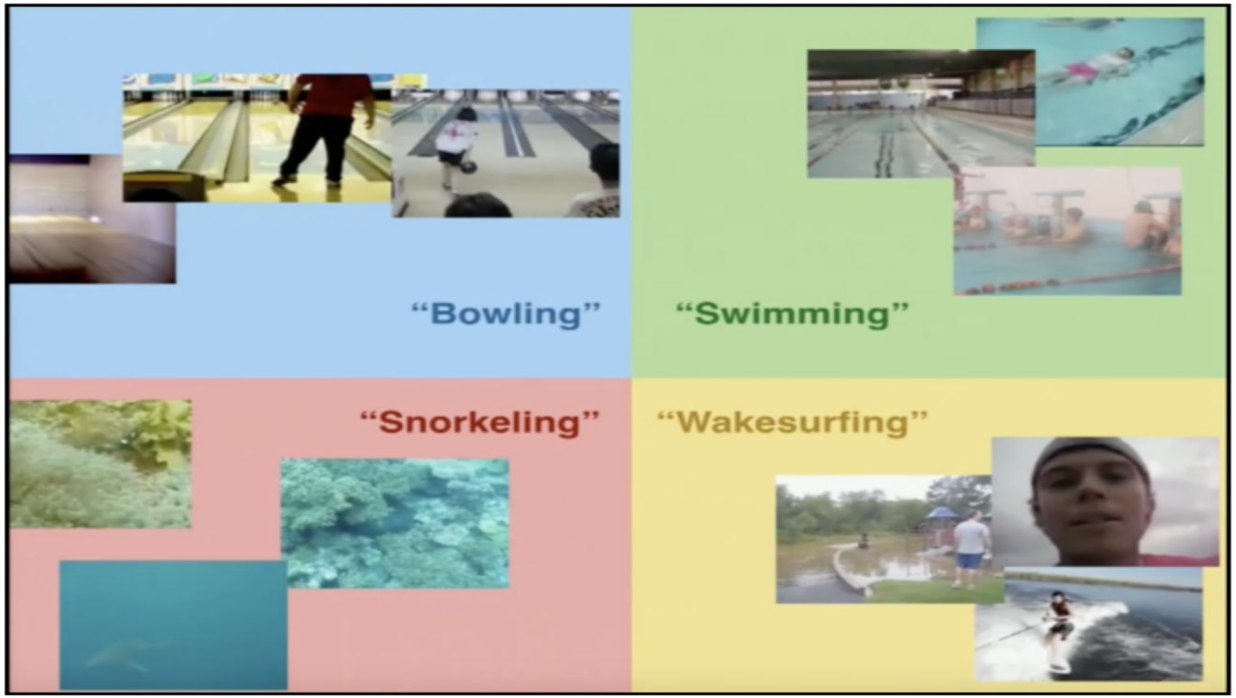
Andrej Karpathy, George Toderici, Sanketh Shetty,

Thomas Leung, Rahul Sukthankar, Li Fei-Fei

Slide borrowed from Esha Uboweja

Problem

Classification of videos in sports datasets



Standard approach to video classification

Bag of Words (BoW) approach:

1. Extraction of local visual features (dense/sparse)
2. Visual word encoding of features
3. Training a classifier (e.g. SVM)

Convolutional Neural Networks (CNNs) emulate all these stages in a single neural network

Motivations for using CNNs for video classification

1. CNNs outperform other approaches in image classification tasks (e.g. ImageNet challenge)

2. Features learned in CNNs transfer well to other datasets (e.g. fine-tuning top layers of a network trained using ImageNet for food recognition)

Dataset

Current video datasets lack variety and number of videos to train a CNN:

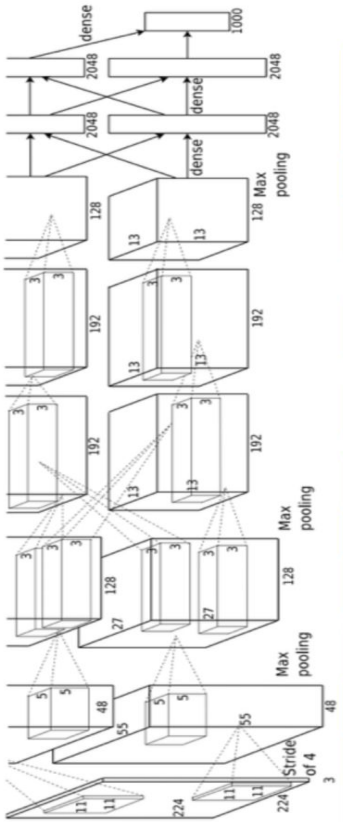
UCF 101 dataset : 13,320 videos, 101 classes

KTH (human action) : 2391 videos, 6 classes

Sports-1M dataset : 1.1 million videos, 487 classes (new!)

Models

Baseline CNN



Fully-Connected

Fully-Connected

Max-Pooling

Convolution

Convolution

Convolution

Normalization

Max-Pooling

Convolution

Normalization

Max-Pooling

Convolution

Image

Goal: Extend 2D convolutional layers to 3D to learn spatio-temporal filters

↓

Call this

Single-frame baseline

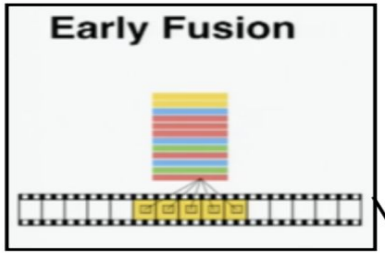
Temporal Fusion in CNNs

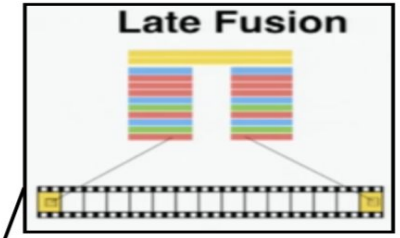
Modify 1st convolutional layer to be of size 11 x 11 x 3 x T pixels

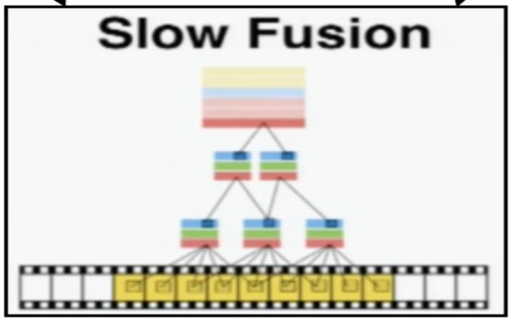
T = # frames (authors use 10)

2 single-frame networks 15 frames apart merge in 1st fully connected layer

The fully connected layer can compute global motion characteristics





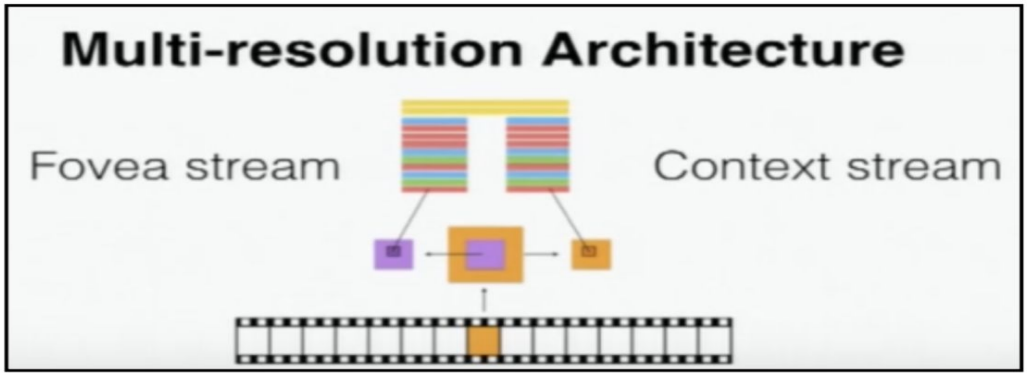


Spatial + temporal convolutions, and higher layers get more global information

Multiresolution CNNs

To improve runtime performance:
Input = 178 x 178 frame video clip

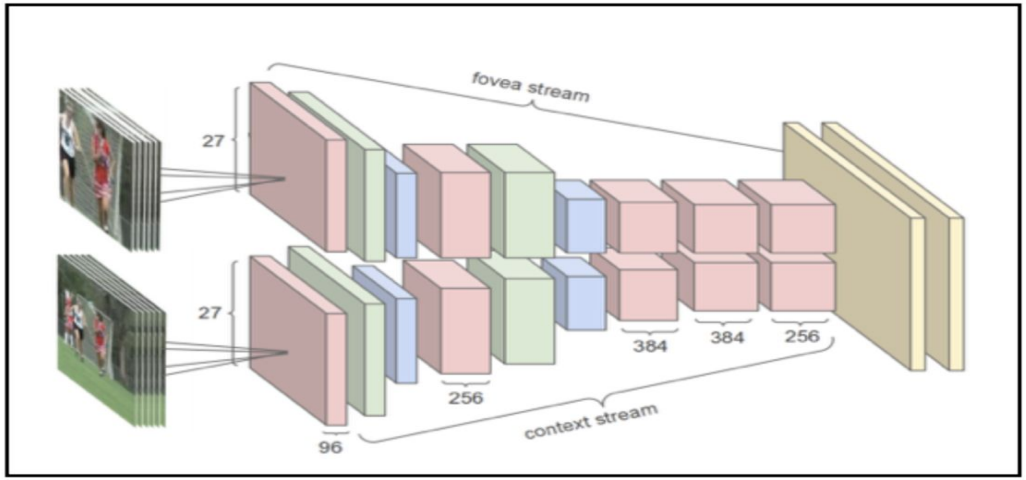
Low-Res Context stream gets down sampled 89 x 89 (entire frame)
High-Res Fovea stream gets cropped center 89 x 89 patch
Both streams merge in 1st fully connected layer



Multiresolution CNNs

To improve runtime performance:
Input = 178 x 178 frame video clip

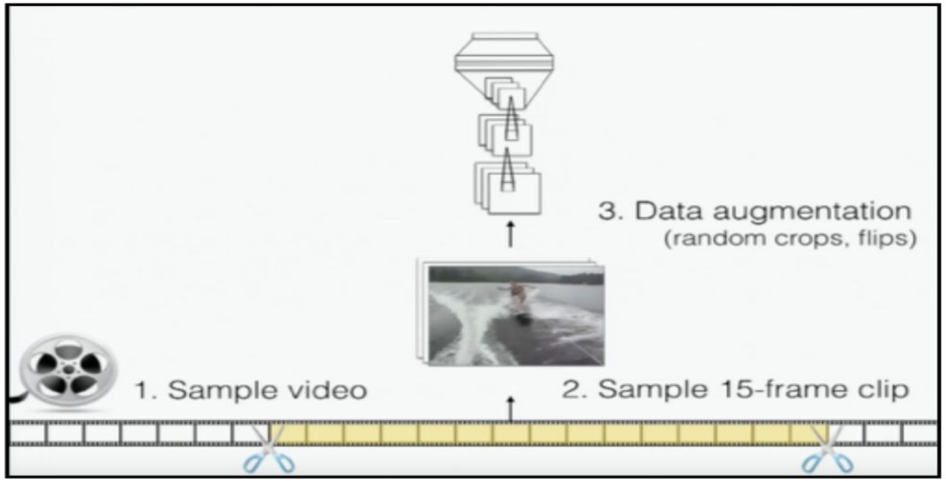
Low-Res Context stream gets down sampled 89 x 89 (entire frame)
High-Res Fovea stream gets cropped center 89 x 89 patch
Both streams merge in 1st fully connected layer



Train Procedure

1. Randomly sample a video
2. Sample a 15 frame (~0.5 secs) clip from (1)
3. Randomly crop, flip frames in clip, subtract mean of all pixels in images (data augmentation + preprocessing)

   Test Procedure is similar



Experiments

Feature Histogram Baseline

1. Extraction of local visual features :
   HoG, Texton, Cuboids, Hue-Saturation, Color moments, #Faces detected

2. Visual word encoding of features:
   Spatial pyramid encoding in histograms after k-means : Finally obtain a 25,000 D feature vector for the entire video

3. Training a classifier:
   Use a 2-hidden layer neural net (worked better than any linear classifier)

Testing Procedure

1. Randomly sample 20 clips for a given test video
2. Present each clip individually to the network (with different crops and flips)
3. Individual clip class predictions are averaged to get a class result for the entire video

Results on Sports-1M dataset

Video Results

https://www.youtube.com/watch?v=qrzQ_AB1DZk

Cycling

Basketball

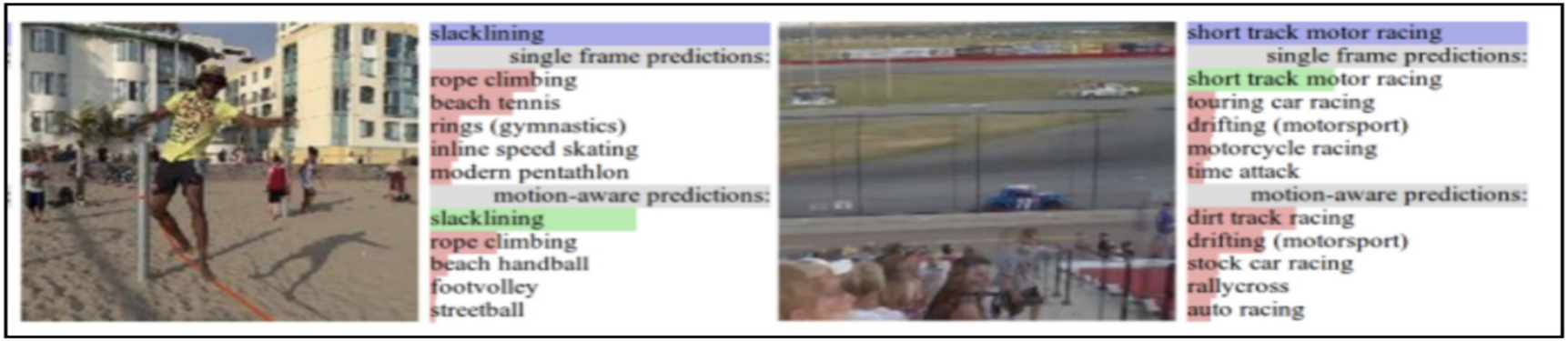
Quantitative Results

Model	Clip Hit@1	Video Hit@1	Video Hit@5
Feature Histograms + Neural Net	-	55.3	-
Single-Frame	41.1	59.3	77.7
Single-Frame + Multires	**42.4**	**60.0**	**78.5**
Single-Frame Fovea Only	30.0	49.9	72.8
Single-Frame Context Only	38.1	56.0	77.2
Early Fusion	38.9	57.7	76.8
Late Fusion	40.7	59.3	78.7
Slow Fusion	**41.9**	**60.9**	**80.2**
CNN Average (Single+Early+Late+Slow)	41.4	63.9	82.4

Qualitative Results

1. The confusion matrix shows that the network doesn't do well on fine-grained classification

2. Slow-fusion networks are sensitive to small motions, hence "motion-aware", but don't work well with presence of camera translation and zoom



Transfer Learning

UCF-101 dataset

5 main categories of data

1. Human Object Interaction
2. Body-Motion only
3. Human-Human interaction
4. Playing Musical Instruments
5. Sports

Soomro et al. ’12

Transfer Learning Performance

Model	3-fold Accuracy
Soomro et al [22]	43.9%
Feature Histograms + Neural Net	59.0%
Train from scratch	41.3%
Fine-tune top layer	64.1%
Fine-tune top 3 layers	**65.4%**
Fine-tune all layers	62.2%

Performance By Categories

Group	mAP from scratch	mAP fine-tune top 3	mAP fine-tune top
Human-Object Interaction	0.26	0.55	0.52
Body-Motion Only	0.32	0.57	0.52
Human-Human Interaction	0.40	0.68	0.65
Playing Musical Instruments	0.42	0.65	0.46
Sports	**0.57**	**0.79**	**0.80**
All groups	0.44	0.68	0.66

Two-Stream Convolutional Networks for Action Recognition in Videos

Authors: Karen Simonyan, Andrew Zisserman

Introduction

Motivation

- At this point, deep-learning approach to action recognition performs a lot worse than best hand-crafted shallow representations.
- This paper aims to come up with a deep convolutional network architecture that performs well in video action recognition tasks

Motivation

- Networks whose inputs are individual frames perform similarly to networks whose inputs are stacks of frames. This suggests that previous attempts did not capture motion in videos well and new approaches needed to capture motion.

Approach

Architecture

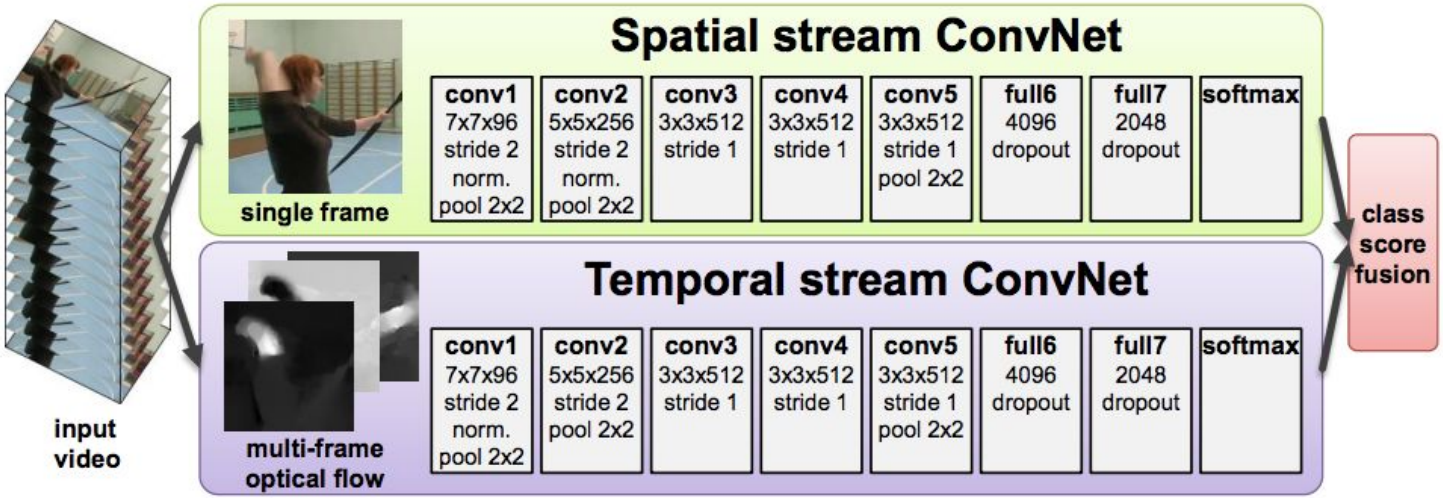


Figure 1: **Two-stream architecture for video classification.**

Temporal Network Input Variations: Optical Flow Stacking vs Trajectory Stacking

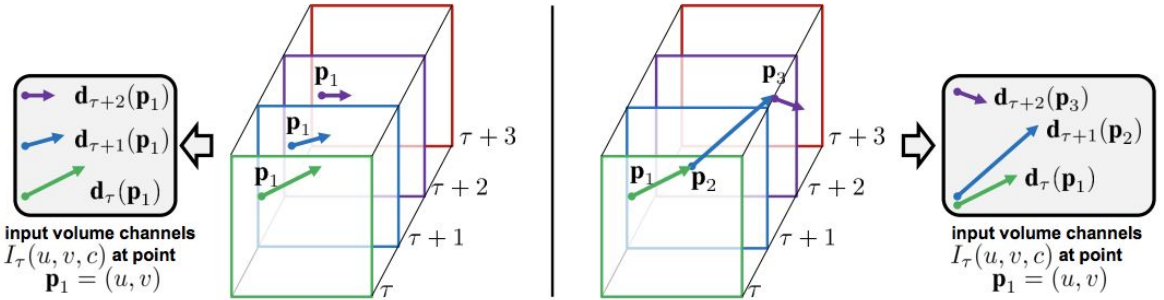
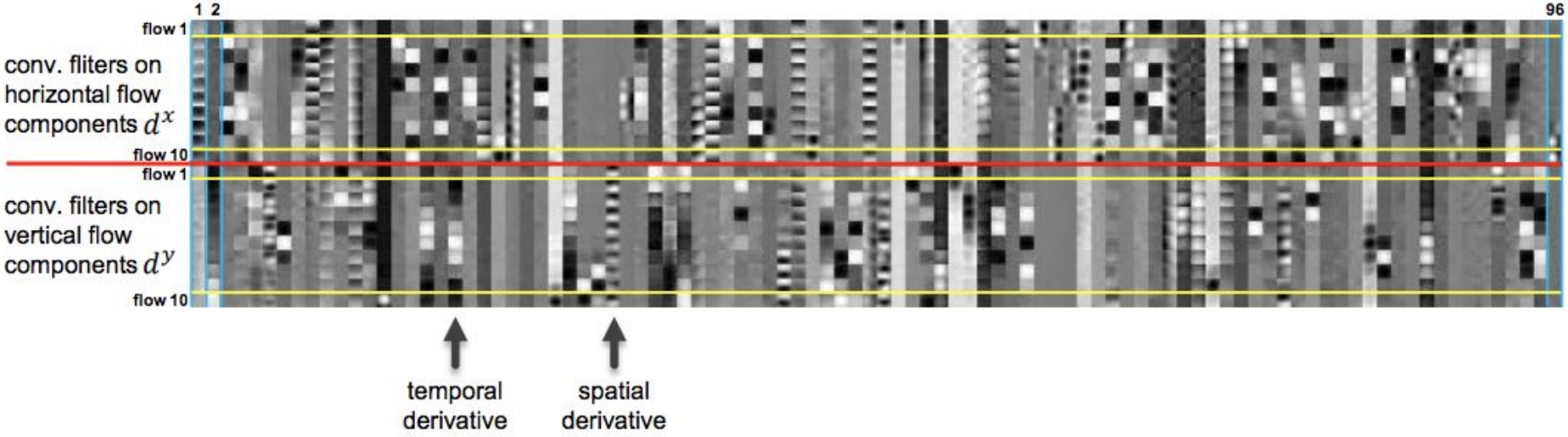


Figure 3: **ConvNet input derivation from the multi-frame optical flow.** *Left:* optical flow stacking (1) samples the displacement vectors **d** at the same location in multiple frames. *Right:* trajectory stacking (2) samples the vectors along the trajectory. The frames and the corresponding displacement vectors are shown with the same colour.

Temporal Network Input Variations: Bidirectional Flows and Mean Flow Subtraction

- Consider both forward optical flows and backward optical flows
- For each displacement field, subtract the mean vector

Visualization of Learnt Convolutional Filters



Multi-task Learning

- The temporal net needs to be trained on videos, but video dataset is small
- Aim to learn a video representation that is applicable not only to the task in question, but to all tasks
- For our network, two softmax layers are added to the top of the last fully connected layers(one to compute HMDB-51 classification score, the other one is to compute UCF-101 score)

Evaluation

Using the Spatial Net or Temporal Net Alone

Table 1: **Individual ConvNets accuracy on UCF-101 (split 1).**

(a) **Spatial ConvNet.**

Training setting	Dropout ratio	
	0.5	0.9
From scratch	42.5%	52.3%
Pre-trained + fine-tuning	70.8%	**72.8%**
Pre-trained + last layer	**72.7%**	59.9%

(b) **Temporal ConvNet.**

Input configuration	Mean subtraction	
	off	on
Single-frame optical flow ($L = 1$)	-	73.9%
Optical flow stacking (1) ($L = 5$)	-	80.4%
Optical flow stacking (1) ($L = 10$)	79.9%	**81.0%**
Trajectory stacking (2)($L = 10$)	79.6%	80.2%
Optical flow stacking (1)($L = 10$), bi-dir.	-	**81.2%**

Temporal Net Performance on HMDB-51

Table 2: **Temporal ConvNet accuracy on HMDB-51 (split 1 with additional training data).**

Training setting	Accuracy
Training on HMDB-51 without additional data	46.6%
Fine-tuning a ConvNet, pre-trained on UCF-101	49.0%
Training on HMDB-51 with classes added from UCF-101	52.8%
Multi-task learning on HMDB-51 and UCF-101	**55.4%**

Two Stream Network Performance

Table 3: **Two-stream ConvNet accuracy on UCF-101 (split 1).**

Spatial ConvNet	Temporal ConvNet	Fusion Method	Accuracy
Pre-trained + last layer	bi-directional	averaging	85.6%
Pre-trained + last layer	uni-directional	averaging	85.9%
Pre-trained + last layer	uni-directional, multi-task	averaging	86.2%
Pre-trained + last layer	uni-directional, multi-task	SVM	**87.0%**

Comparison with the State of Art

Table 4: **Mean accuracy (over three splits) on UCF-101 and HMDB-51.**

Method	UCF-101	HMDB-51
Improved dense trajectories (IDT) [26, 27]	85.9%	57.2%
IDT with higher-dimensional encodings [20]	**87.9%**	61.1%
IDT with stacked Fisher encoding [21] (based on Deep Fisher Net [23])	-	**66.8%**
Spatio-temporal HMAX network [11, 16]	-	22.8%
"Slow fusion" spatio-temporal ConvNet [14]	65.4%	-
Spatial stream ConvNet	73.0%	40.5%
Temporal stream ConvNet	83.7%	54.6%
Two-stream model (fusion by averaging)	86.9%	58.0%
Two-stream model (fusion by SVM)	**88.0%**	**59.4%**

Future Work

- Train on Sports-1M(Huge Volume Presents Additional Challenge)
- Use more sophisticated techniques to correct camera motion