NC STATE UNIVERSITY

Video Classification and Action Recognition

Ravindersingh Rajpal (rkrajpal@ncsu.edu)
FNU Vivek (vvivek@ncsu.edu)

Content

- Introduction
- Dataset
- C3D
- I3D
- TSN
- Results
- Demo
- Limitations
- Future Work

Introduction

- Action Recognition on UCF-101 dataset.
- Create 3 DNN architectures for video classification.
  - 3D ConvNet (C3D) - 2014
  - Two-Stream Inflated 3D ConvNet (I3D) - 2017
  - Temporal Segment Networks (TSN)- 2016
- Pre-trained weights are used to initialize the models
- Training and Validation loss analysis
- Metric used: Accuracy

Dataset

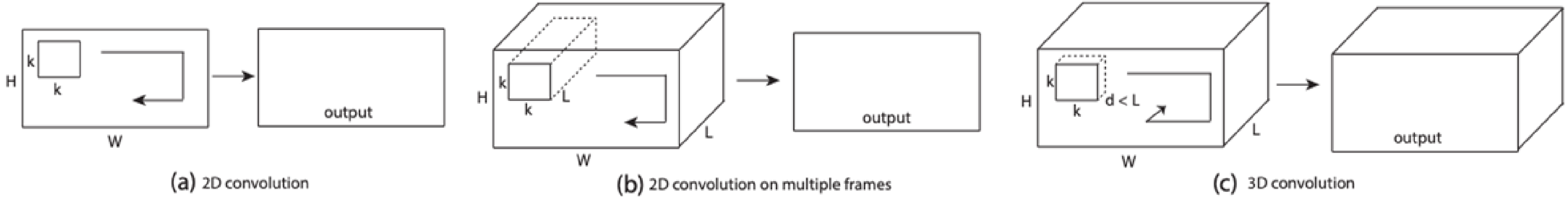
- UCF-101 dataset
  - Action recognition data set of realistic action videos from YouTube
  - 101 action categories, 13320 videos
- 25 video groups (4-7 videos of an action)
- 5 action categories
  - Human-Object Interaction
  - Body-Motion Only
  - Human-Human Interaction
  - Playing Musical Instruments
  - Sports

https://www.crcv.ucf.edu/data/UCF101.php

3D ConvNet

- “Learning Spatiotemporal Features with 3D Convolutional Networks”, Du Tran et al., Dec 2014.
- Repurposing 3D convolutional networks as feature extractors.
- Extensive search for best 3D convolutional kernel and architecture.
- Using deconvolutional layers to interpret model decision.
- Pre-trained weights are used to initialize the model and whole model is re-trained with a dense layer (softmax).

C3D - Architecture



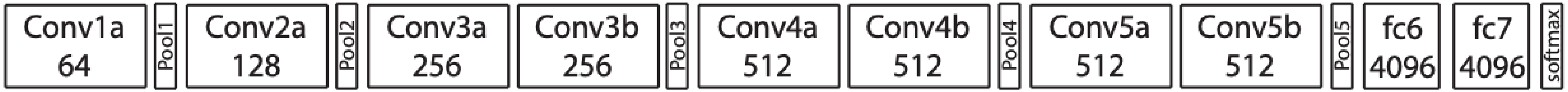
(a) 2D convolution

(b) 2D convolution on multiple frames

(c) 3D convolution

Difference between 2D and 3D covolution

Source:https://arxiv.org/pdf/1412.0767.pdf



Conv1a 64 | Pool1 | Conv2a 128 | Pool2 | Conv3a 256 | Conv3b 256 | Pool3 | Conv4a 512 | Conv4b 512 | Pool4 | Conv5a 512 | Conv5b 512 | Pool5 | fc6 4096 | fc7 4096 | softmax

C3D architecture

Source:https://arxiv.org/pdf/1412.0767.pdf

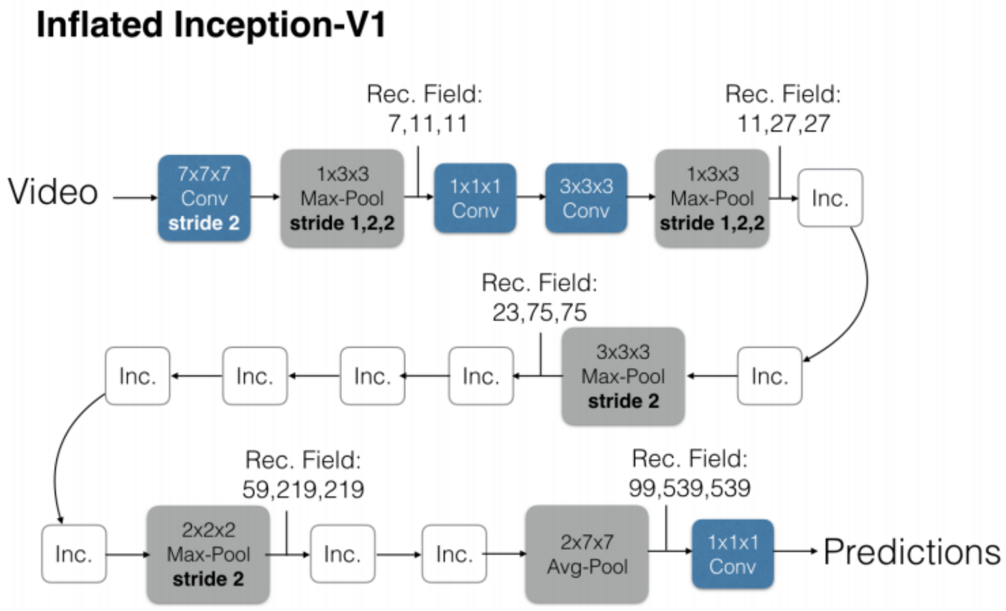
C3D - Hyperparameters

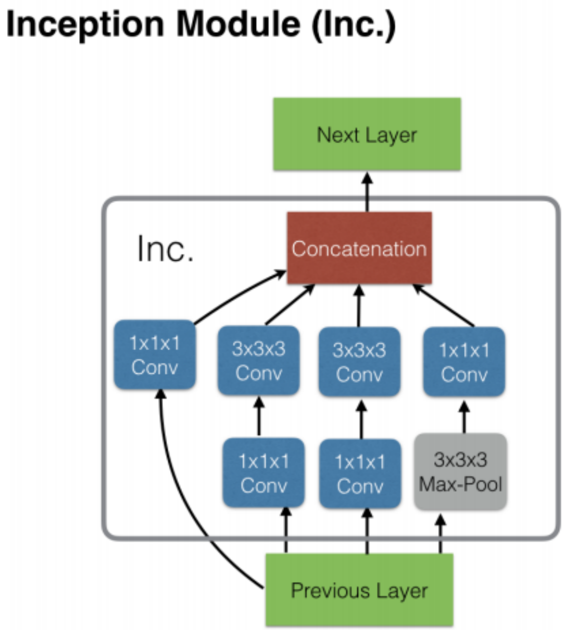
Hyperparameter	Value
Learning Rate	0.001
Dropout	0.5
Batch size	16
Number of frames per video	16
Optimization function	SGD(momentum=0.9)

Two-Stream Inflated 3D ConvNet (I3D)

- “Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset”, Carreira et al., May 2017.
- Combining 3D based models into two stream architecture leveraging pre-training.
- Spatial stream input had frames stacked in time dimension.
- Pre-trained weights are used to initialize the model and whole model is re-trained with a dense layer (softmax).

I3D - Architecture





Source: https://arxiv.org/pdf/1705.07750.pdf

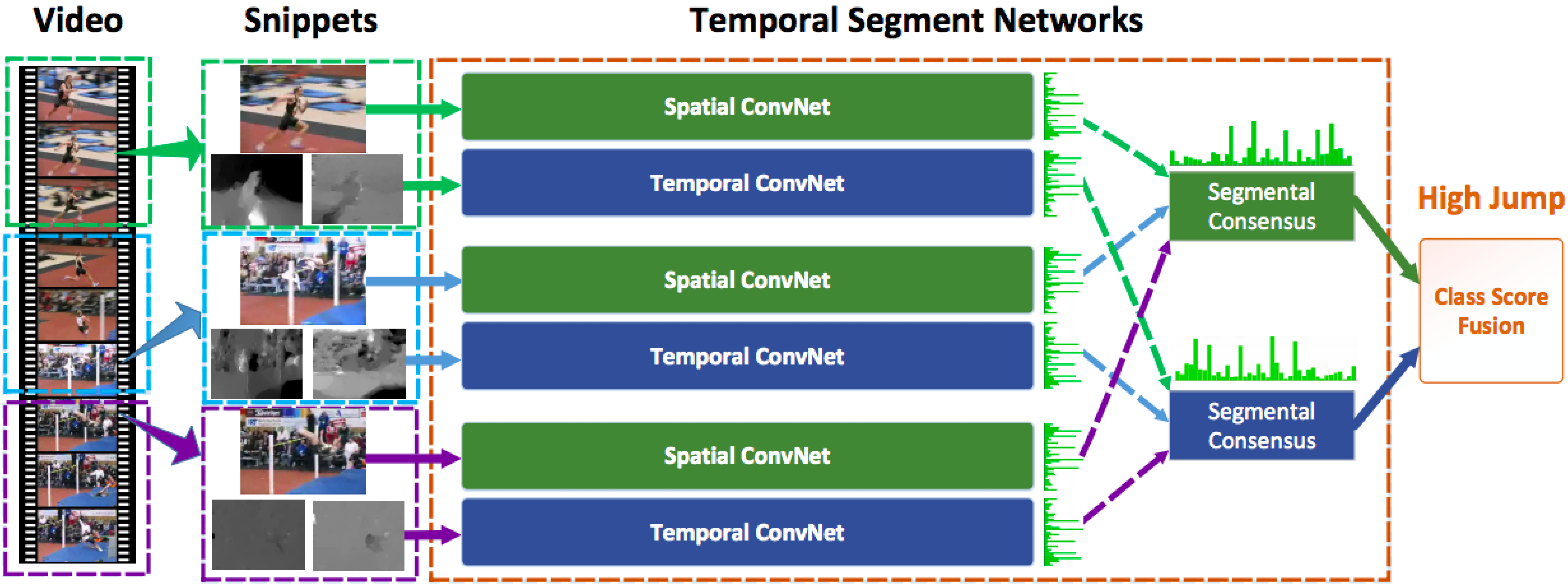
I3D - Hyperparameters

Hyperparameter	**Value**
Learning Rate	0.001
Dropout	0.5
Batch size	16
Number of frames per video for FRAME architecture	16
Number of flows per video for FLOW architecture	16
Optimization function	SGD(momentum=0.9)

Temporal Segment Networks

- "Temporal Segment Networks: Towards Good Practices for Deep Action Recognition", Wang et al., Aug 2016.
- Sample clips (segments) sparsely across video aimed at long range temporal modeling.
- Combine scores of temporal and spatial streams by averaging across snippets.
- Fuse score of final spatial and temporal scores and average across all classes.
- Use pre-trained weights for Inception modules (we used Keras implementation).

TSN - Architecture

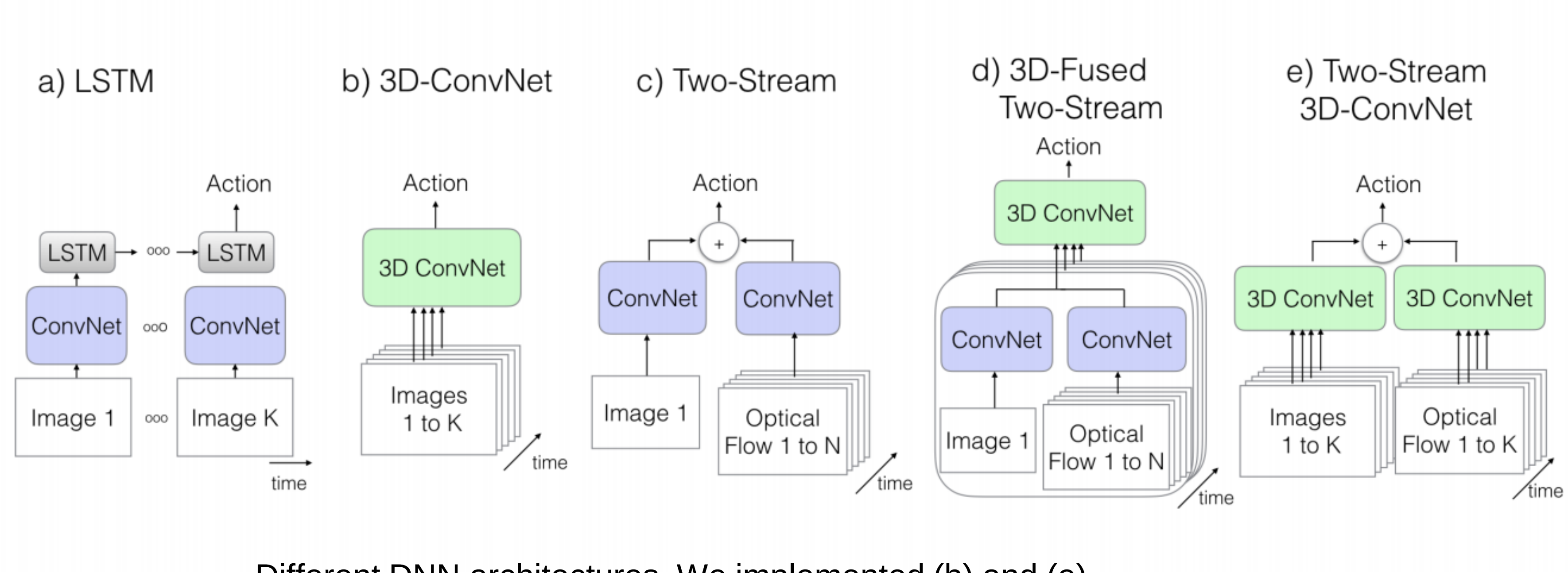


Source: https://arxiv.org/pdf/1608.00859.pdf

TSN - Hyperparameters

Hyperparameter	**Value**
Learning Rate	0-44 epochs: 1e-3, 45-200: 2e-3
Dropout	0.5
Batch size	16
Number of frames per video	1
Number of flows per video	1
Optimization function	Adam

Different DNN Architectures

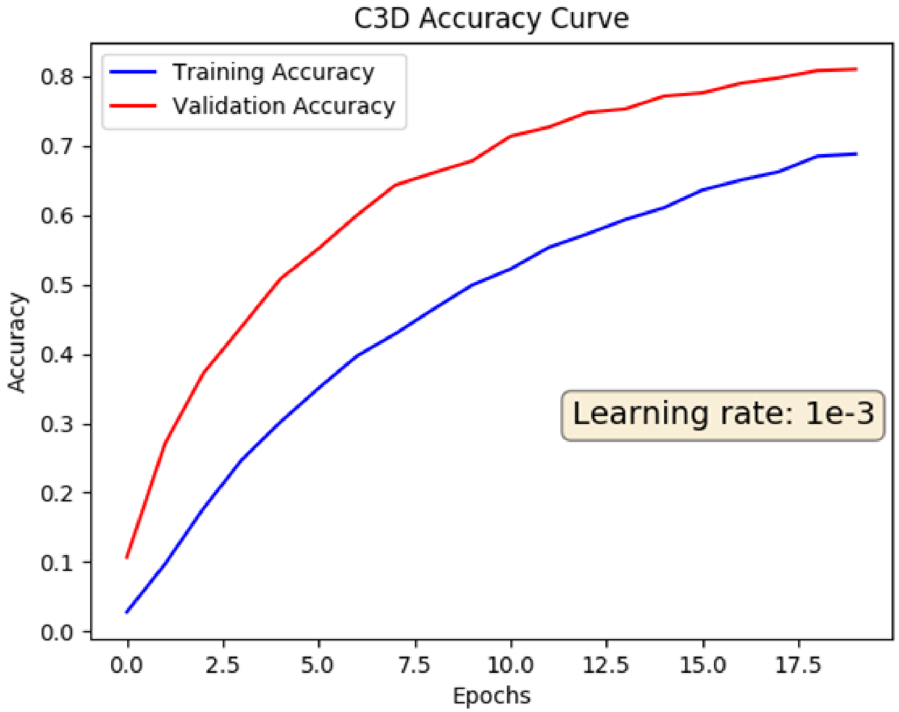


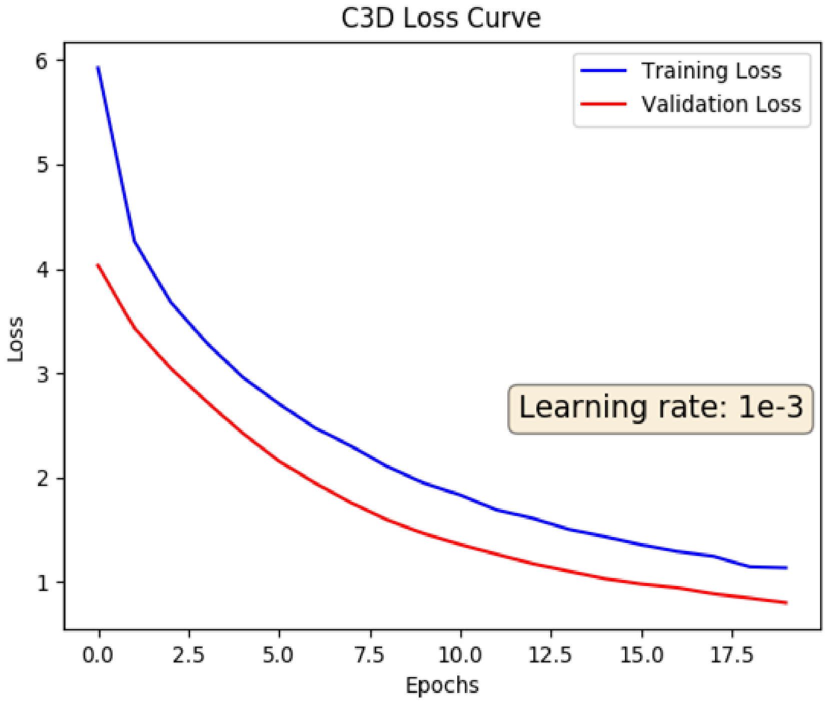
Different DNN architectures. We implemented (b) and (e)

Source: https://arxiv.org/pdf/1705.07750.pdf

Training and Validation Graphs - C3D

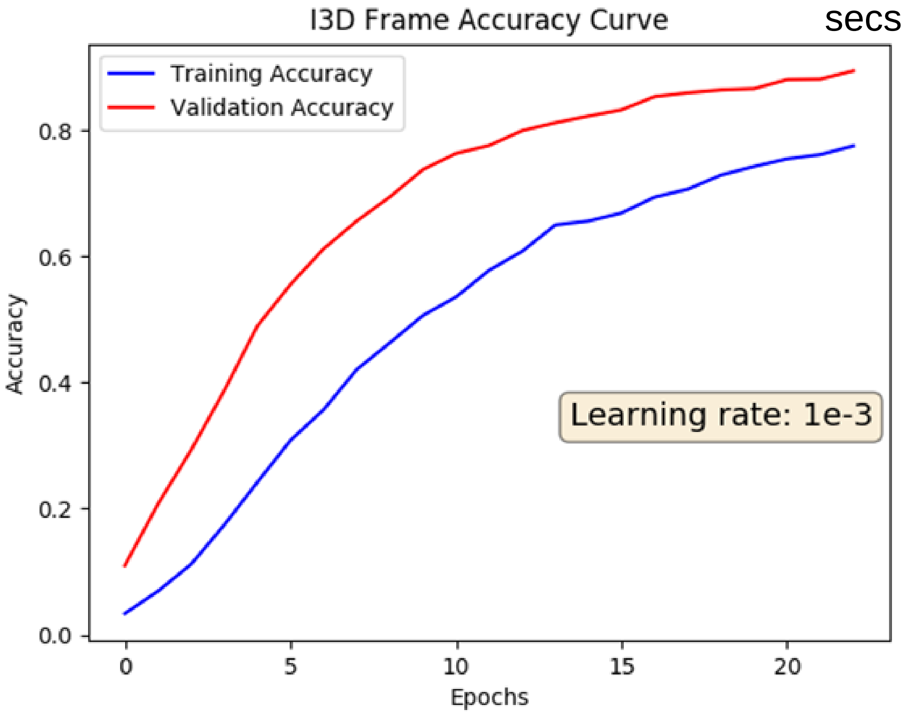
Training time 556 secs/epoch

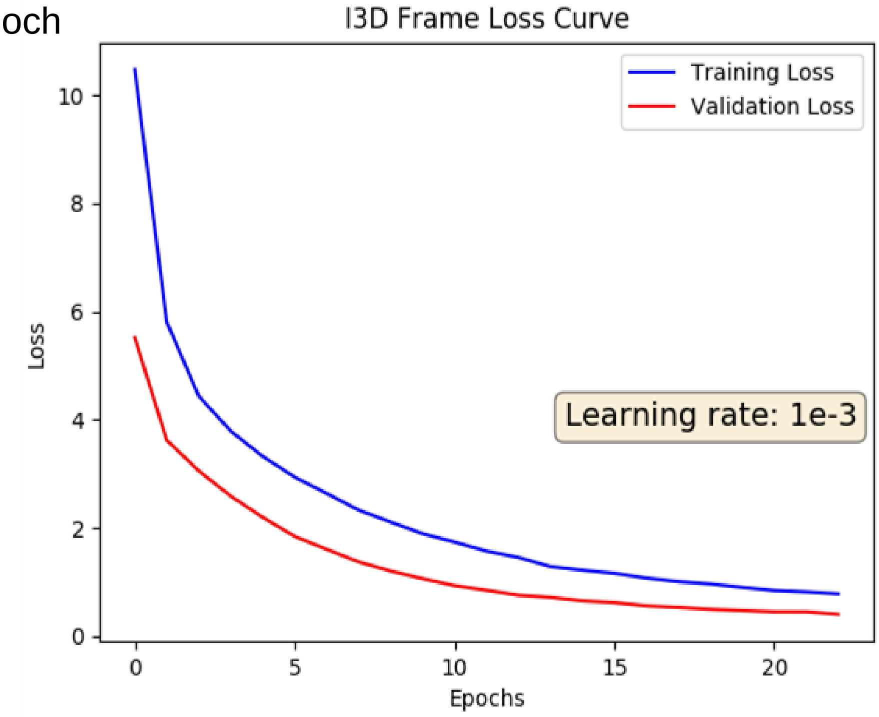




Training and Validation Graphs - I3D

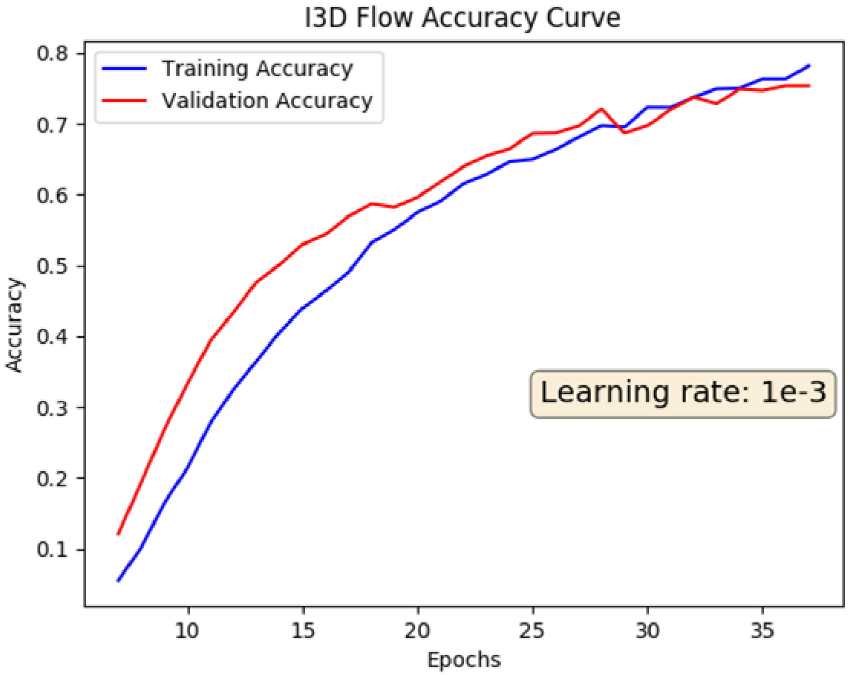
Frames: Training time 628 secs/epoch

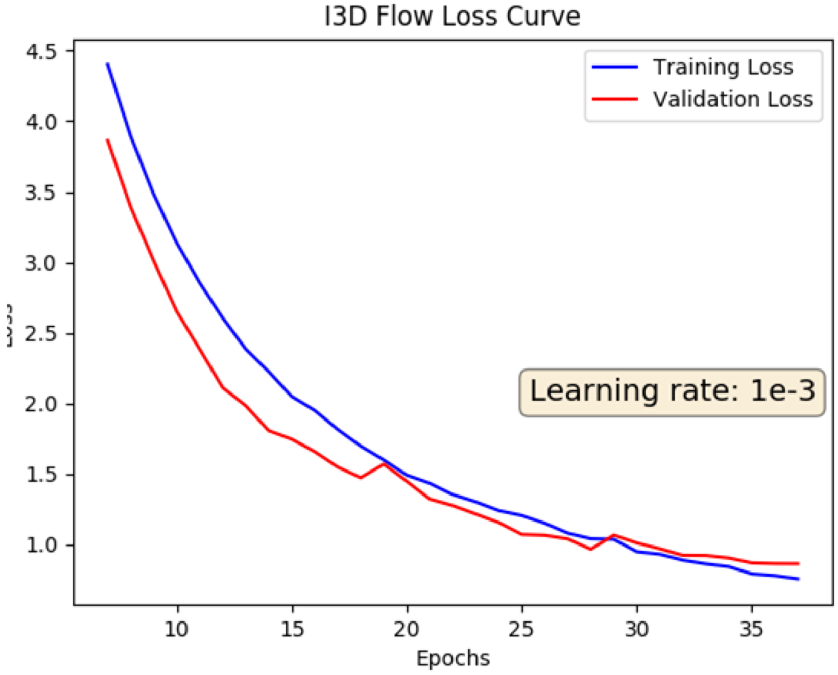




Training and Validation Graphs - I3D

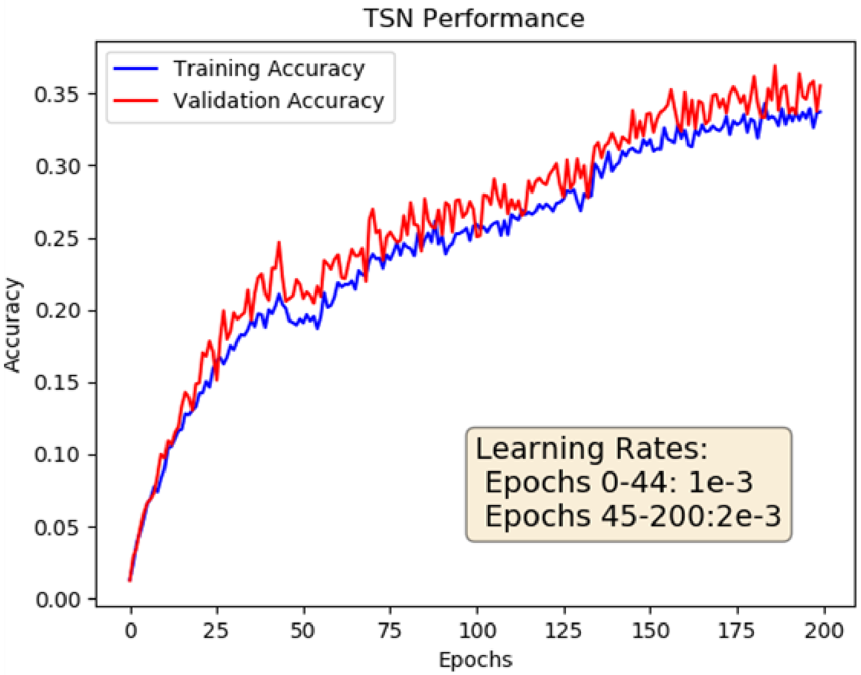
Flows : Training time 2552
secs/epoch

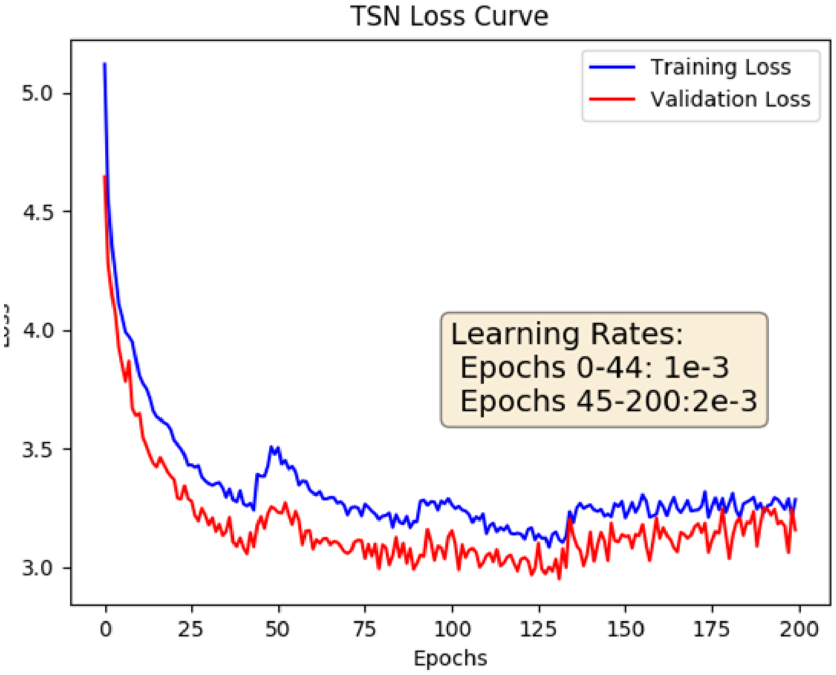




Training and Validation Graphs - TSN

Training time 435 secs/epoch



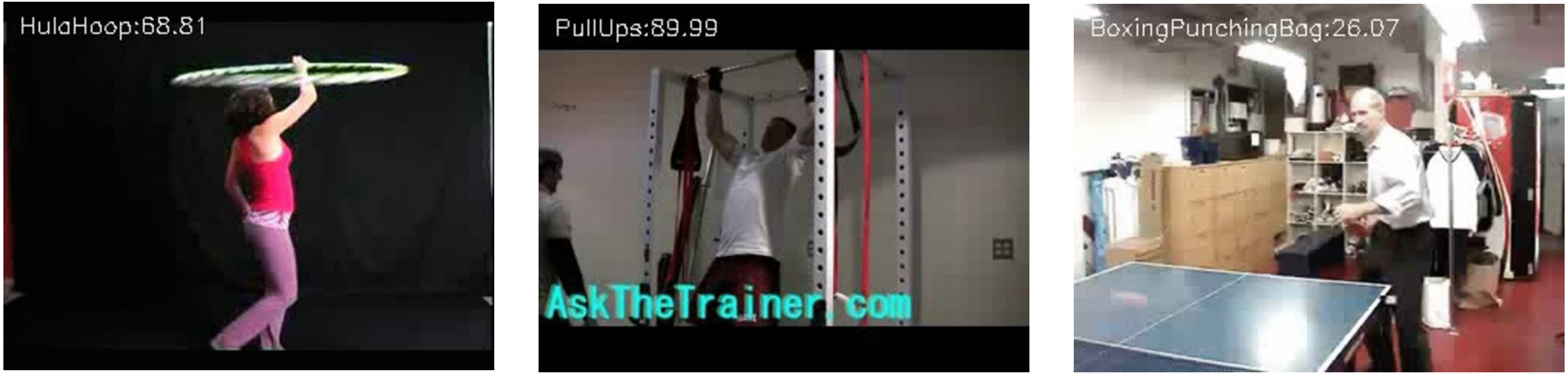


Results

DNN Architecture	Test Accuracy
C3D	83.2%
I3D	74.5%
TSN	38.1%

Note: TSN was still undergoing training, after 200 epochs and 25 hours of training, we still see a rise in validation and training accuracy. Due to computational complexity and time constraints we decided to report the last achieved accuracy.

Demo



C3D model sample predictions

Limitations

- C3D pre-trained architecture input required 112x112
  - 224x224 frames were scaled down with a factor of 2
  - Long range temporal modelling still an issue.
- I3D model required to take average output of Flow and Frame models
  - A simple average after taking argmax was reported
- TSN had more than 300 million trainable params
  - Imagenet pre-trained weights were used
  - Model was trained for 25 hours for 200 epochs
  - Time and resource constraints
  - Increasing learning rate led to under-fitting
  - Keras implementation flaw for BatchNormalization layer

Future Work

- Train C3D and I3D model for more epochs to get even better validation accuracy.
- Implement TSN in Pytorch that has no issue with BatchNormalization.
- Train TSN with better learning rate for more epochs.
- Compare results with publicly available split results.

Thank You!