Human Activity Recognition in Temporally Untrimmed Videos

A report submitted to the
Indian Institute of Technology, Kharagpur
in partial fulfillment of the requirements for the degree

of

Bachelor of Technology
(Hons.)

by

Sudeep Raja Putta
(12CS10038)

under the supervision
of

Prof. Partha Pratim Das

Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

May 2016

Declaration

This thesis is a presentation of my original research work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions. The work was done under the guidance of **Prof. Partha Pratim Das** , at Indian Institute of Technology, Kharagpur.

Sudeep Raja Putta
Department of Computer Science and Engineering,
Indian Institute of Technology, Kharagpur.

Certification

This is to certify that the project entitled **Human Activity Recognition in Temporally Untrimmed Videos** being submitted by Sudeep Raja Putta is a bonafide work done by him in the Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur under my supervision and guidance for Bachelor's Thesis Project.

Prof. Partha Pratim Das

Department of Computer Science and Engineering,

Indian Institute of Technology, Kharagpur.

Acknowledgements

I extend my sincere gratitude and indebtedness to my Supervisor, Prof. Partha Pratim Das, Department of Computer Science And Engineering, IIT Kharagpur, for continuously supporting me during the course of the project and providing me with valuable inputs and feedbacks which helped me immensely in carrying out the work.

Abstract

Activity recognition, which is a straightforward task for humans is difficult for computer systems as it involves processing large volumes of information and recognizing patterns from spatial and temporal domains. Deep Learning models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have achieved significant accuracies on activity recognition datasets which consist of short videos consisting of a single activity. We address the task of recognizing activities in longer temporally untrimmed videos by splitting the task into two subtasks. The first subtask uses a Deep Neural network for predicting the temporal extents of the activities present in long videos. The second subtask again uses a Deep Neural network for recognizing activity occurring in the predicted extents. We analyse our models on a subset of the UCF-101 and Thumos-15 datasets. As future work, we propose a few deep architectures for the same task and also discuss about possible new datasets.

Contents

Chapter 1

Introduction

1.1 Background

Activity recognition is the task of recognizing ongoing activities in videos. Human activities are composed of complex actions varying in their spatial and temporal dynamics and often involve interaction with other humans and objects as well. Humans have a highly evolved visual cortex which can detect and recognize activities with ease and requires no conscious supervision. Computer vision systems which aim to solve the same task depend heavily on hand crafted features and expert supervision.

Deep Learning is a branch of machine learning which has regained attention in recent years. Instead of expert designed features, deep learning systems can be trained to learn discriminative features from the raw data itself. The fundamental tool of deep learning is an artificial neuron. These neurons can combined and stacked with other neurons to create a neural network. These neural networks act as potent function approximators. Neural networks can be trained using gradient descent algorithms for tasks such as regression or classification. Specific neural networks exist for specialized tasks. Convolutional neural networks(CNN) are very effective at tasks involving processing of spatial data such as image recognition. Recurrent neural networks(RNN), specifically Long Short Term Memory(LSTM) neural networks are very effective at task involving processing of temporal sequences such as speech recognition. Since videos consist of spatio-temporal information, a network consisting of both CNNs and RNNs is suitable for processing videos.

1.2 Motivation

Human activity recognition is an active area of research in the field of computer vision. The ability of recognizing complex human activities from videos has several important applications. Some applications include: surveillance systems for automatically recognizing suspicious activities in public areas, monitoring of patient activity in hospitals, controlling devices based on gestures.

Video cameras have become ubiquitous in devices such as smart-phones, drones and autonomous vehicles. Because of the presence of cameras in so many devices, there is also an abundance of video data available. Systems which can process video streams and analyse continuous human activities are necessary. This data could be leveraged to enable the construction of several intelligent applications based on activity or gesture recognition.

Since video data is highly voluminous, these systems must be able to process video frames very quickly. Due to availability of specialized General Purpose GPUs which provide high throughput to easily parallelizable operations, we are able to develop complex video processing systems which achieve high data processing rates and can work in real time.

Many state-of-the-art computer vision systems have been developed which make use of deep neural networks. These deep networks use Convolutional neural networks and can be trained on GPUs for classification tasks if labelled data is available. We also make use of deep learning to build our system.

1.3 Scope

Activity recognition systems developed so far which use deep learning have concentrated on short videos. In our work we develop a system to recognize human activities in temporally untrimmed videos. Such a system could be used to look for activities of interest in a direct video stream. It could be used to partition videos based on activity occurring in them. Applications for such a system includes efficient storing, searching and retrieval of videos based on the activity, systems which can describe videos automatically and perform automatic video summarization and video question answering.

Chapter 2

Neural Networks

2.1 Artificial Neural Networks (ANN)

ANNs are models inspired by biological neural networks and are capable of estimating or approximating functions. A neuron has m inputs which is the input feature vector x_1 through x_m and weights w_1 through w_m. The neuron computes the weighted sum of the inputs and applies a non linear transform function σ usually ReLU or sigmoid on the sum. A bias term is usually added. The output of the neuron is:

$$y = \sigma(\sum_{i=0}^{m} w_j x_j)$$

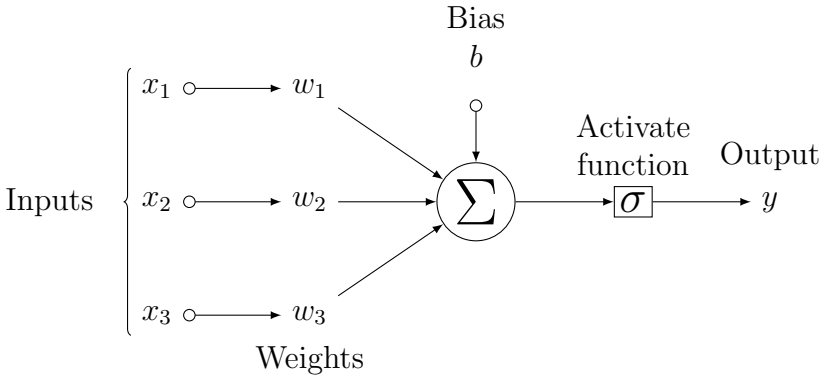


Figure 2.1: Artificial Neuron

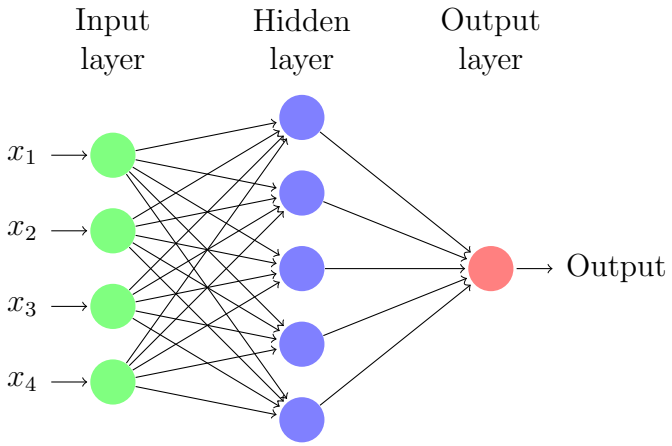


Figure 2.2: FeedForward Neural Network

Many such neurons can be used simultaneously, each with its own set of weights but with the same input vector. This makes a layer of neurons. If the input feature vector is x and W is the matrix of weights then the output of a layer would be the vector $y = \sigma(W \cdot x)$. Several such layers can be stacked together to make a feedforward neural network(FNN), so that the output of previous layer is input to the next layer. The activation of the last layer may be sigmoid or softmax depending on the task the neural network is being used for. The neural network may be thought of as a function whose input is the feature vector and all the weights are learnable parameters that can be tuned to minimize a loss function. Neural networks are typically trained using Stochastic Gradient Descent(SGD) and the backpropagation algorithm.

2.2 Convolotional Neural Networks (CNN)

CNNs are a type of FNN in which the connectivity of neurons is inspired by the organization of the animal visual cortex. CNNs are invariant to translation, rotation and shifting and are ideal for spatial inputs such as images. CNNs consist of Convolution layer, Pooling layer and Fully connected layer. The input to CNN is an input volume of size $height \times width \times depth$. For an image this would correspond to $height \times width \times channels$.

Convolution Layer: This consists of several learnable kernels or feature maps which have a small receptive field but extend through the full depth of the input volume. The

kernels perform a convolution operation on the input volume across the width and height and applies a non linear transform to output volume. The outputs of all the kernels are stacked to create the output volume. Let x be the input volume and W be the kernel matrices. The equation for this layer's operation is as follows:

$$y = \sigma(W * x)$$

Pooling Layer: After each convolution layer, there may be a pooling layer. The pooling layer takes small rectangular blocks from the convolution layer and sub-samples it to produce a single output from that block. Popular pooling methods are using the maximum or average. Its function is to progressively reduce the spatial size of the volume and to reduce the amount of parameters and in the network.

Fully Connected Layer: This layer is a conventional layer of a FNN. These are present at the top of all other layers and are used to perform the classification or regression task.

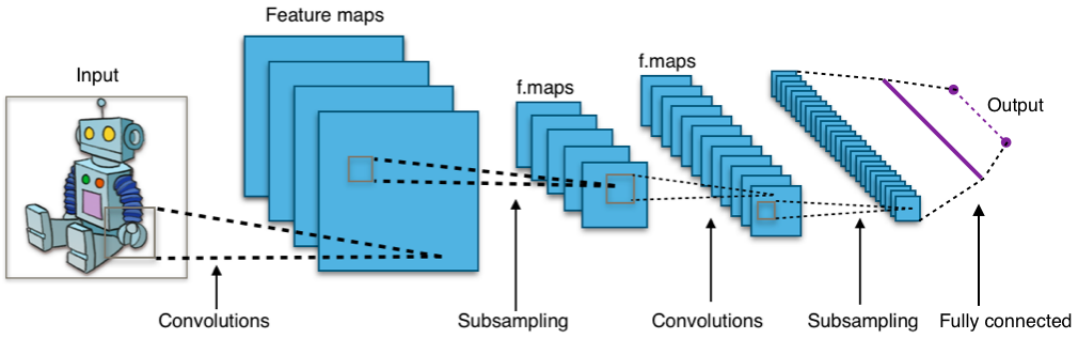


Figure 2.3: Convolotional Neural Network

It is found that the kernels which the CNN learns correspond to filters which are active when a specific object or shape is present in the image. In practise training CNNs from scratch is difficult it is relatively rare to have a dataset of sufficient size. In this case, transfer learning, either by using CNN as fixed feature extractor or Fine-tuning few layers of CNN can be performed.We could also initialize the lower layers of our CNN with the kernels of pretrained CNNs.

2.3 Recurrent Neural Networks (RNN)

RNNs are inspired by the cyclical connectivity of neurons in the brain. These use iterative function loops to store information. RNNs are used to process sequences. At each step, neurons in the recurrent layer take as input the current input vector and the outputs of the neurons of the same layer from the previous step. A FNN can only map from input to output vectors, whereas an RNN can in principle map from the entire history of previous inputs to each output. The recurrent connections allow a 'memory' of previous inputs to persist in the networks internal state, and thereby influence the network output. Let x_j be the input feature vector and y_j be the output vector of the RNN at time j. Let W_i and W_r be the weights matrix of the input and recurrent edges. The output of the RNN layer will be a sequence of vectors given by:

$$y_j = \sigma(W_i \cdot x_j + W_r \cdot y_{j-1})$$

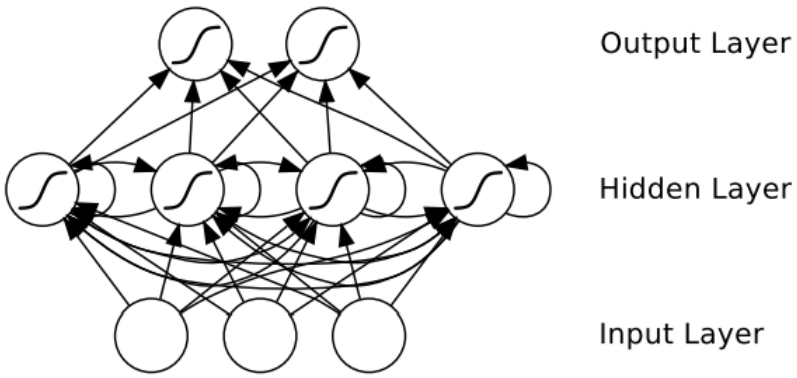


Figure 2.4: Recurrent Neural Network [1]

RNNs can be trained by SGD and Backpropagation Through Time (BPTT). However in practice RNNs face the vanishing gradient problem, where the network output either blows up or decays exponentially as it cycles around the networks recurrent connections. Several types of RNNs have been proposed to remedy the problems of simple RNNs, including Gated Recurrent Units(GRU) and Long Short Term Memory(LSTM).

Long Short Term Memory (LSTM)[10]: LSTMs were proposed to circumvent the vanishing gradient problem. The LSTM architecture consists of a set of recurrently connected subnets, known as memory blocks. Each block contains one self-connected memory cells and three multiplicative units the input, output and forget gates that provide write, read and reset operations for the cells. These additional cells enable the LSTM to learn extremely complex and long-term temporal dynamics the RNN is not capable of learning. The equations for a LSTM layer are:

$$i_t = f(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci} \circ c_{t-1} + b_i)$$

$$f_t = f(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf} \circ c_{t-1} + b_f)$$

$$o_t = f(W_{xo}x_t + W_{ho}h_{t-1} + W_{co} \circ c_{t-1} + b_o)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ g(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$

$$h_t = o_t \circ h(c_t)$$

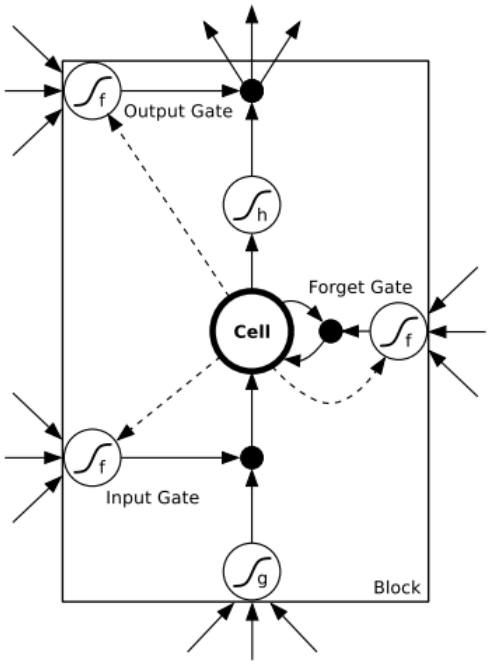


Figure 2.5: LSTM cell [1]

Chapter 3

Deep Learning Architectures for Video Processing

Video processing involves handling two streams of information. First is the spatial information present in each frame. This information can be used to recognize which objects appear in the frame and at which location. Second is the temporal information present across frames. This information is used for understand how the objects move and interact. Activity recognition is a task which requires processing of both the streams of information. For instance, we could extract information about how the limbs and body parts are oriented from the spatial stream. Form the temporal stream, we could extract information about how the limbs move and interact with other body parts. This information could be used for recognizing the activity.

CNNs are capable of processing images and extracting the spatial information from them. RNNs could use the spatial information fed as a sequence and learn the temporal dynamics of the spatial data. Thus an architecture consisting of both CNNs and RNNs is ideal for machine learning tasks involving processing of spatio-temporal data such as videos. In this chapter we introduce a few architectures which could be used for processing videos.

3.1 2-D CNN + RNN

Say we are processing T frames of a video at a time. These frames are fed to a single 2 dimensional CNN which outputs a sequence of T vectors, each representing the spatial information present in the corresponding frame. This sequence of T vectors is fed to the RNN which could be used for classification of the T video frames. The RNN is placed above the CNN and the whole architecture can be trained end to end using Backporpagation and SGD. Instead of sharing a single CNN, we could have used T different CNNs - such an architecture is called Long Recurrent Convolutional Network (LRCN).

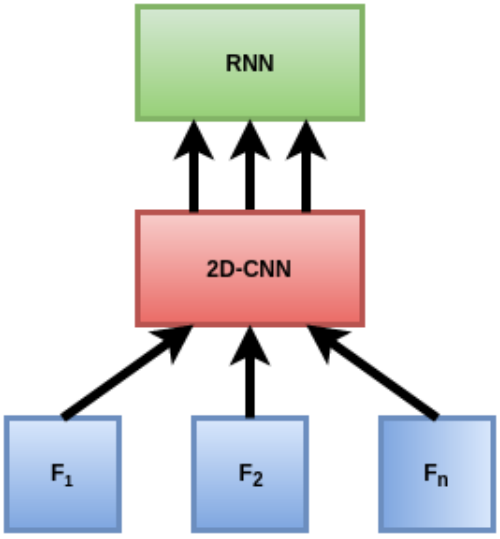


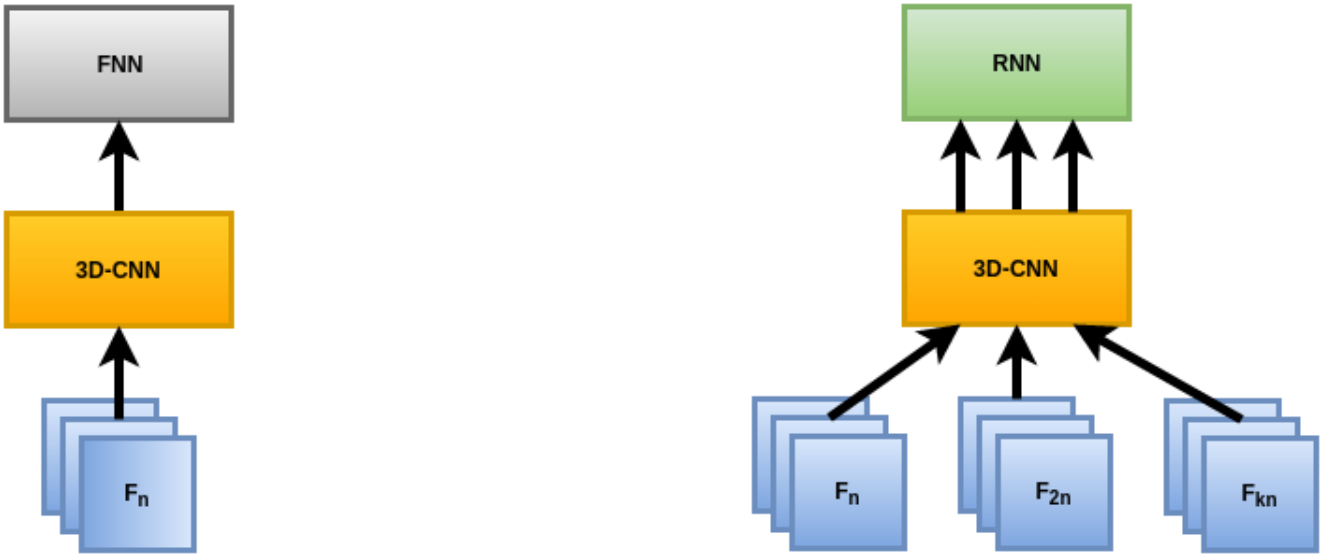
Figure 3.1: 2-D CNN + RNN

3.2 3-D CNN

3 dimensional CNNs are capable of processing spatial and temporal information within a specified receptive field size because of 3D convolution and 3D pooling operations. These 3-D CNNs could be used directly for processing and classification of T frames of a video.

3D-CNNs could also be used to replace 2-D CNNs in the architecture discussed in the previous section. The T frames would broken into multiple frame sequences of size t. These t frame sequences are stacked to create an input volume which is fed to the 3-D CNN. The output would be a vector representing both the spatial and temporal information present in

the frame sequence. A sequence of such vectors are fed into the RNN which classifies the T frames of the videos.



(a) 3D CNN (b) 3D CNN + RNN

Figure 3.2: 3D Convolotional Architectures

3.3 Two stream CNN

In practise, 3-D CNNs are difficult to train and are prone to over fitting. A novel idea is to be able to capture temporal information using solely 2-D CNNs. This can be done by using dense optical flow images.Consider two frames at t and $t + \Delta t$ seconds. A pixel at (x, y) at t and at $t + \Delta t$, it would be at $(x + \Delta x, y + \Delta y)$. Dense optical flow estimates Δx and Δy for each pixel.

Δx and Δy can be seen as image channels suited for recognition using a CNN. To represent the motion across a sequence of frames, we stack the flow channels of T consecutive frames. The 2 stream model has two CNNs. The first one, called the Spatial CNN takes as input the first frame of the video. The second one, called the Temporal CNN takes as input the the stacked flow channels. The output vectors of the two CNNs can be concatenated and used for classification of the T frames.

An alternative to flow stacking is flow images. A Flow image is created by centring x and y flow values around 128 and multiplying by a scalar such that flow values fall between 0 and

255. A third channel for the flow image is created by calculating the flow magnitude. A two stream CNN in which the spatial CNN takes a frame as input and the temporal CNN takes a flow image as input can produce a representation vector which encodes both spatial and temporal information. Such a 2 steam CNN can replace 2-D CNNs in the first architecture.

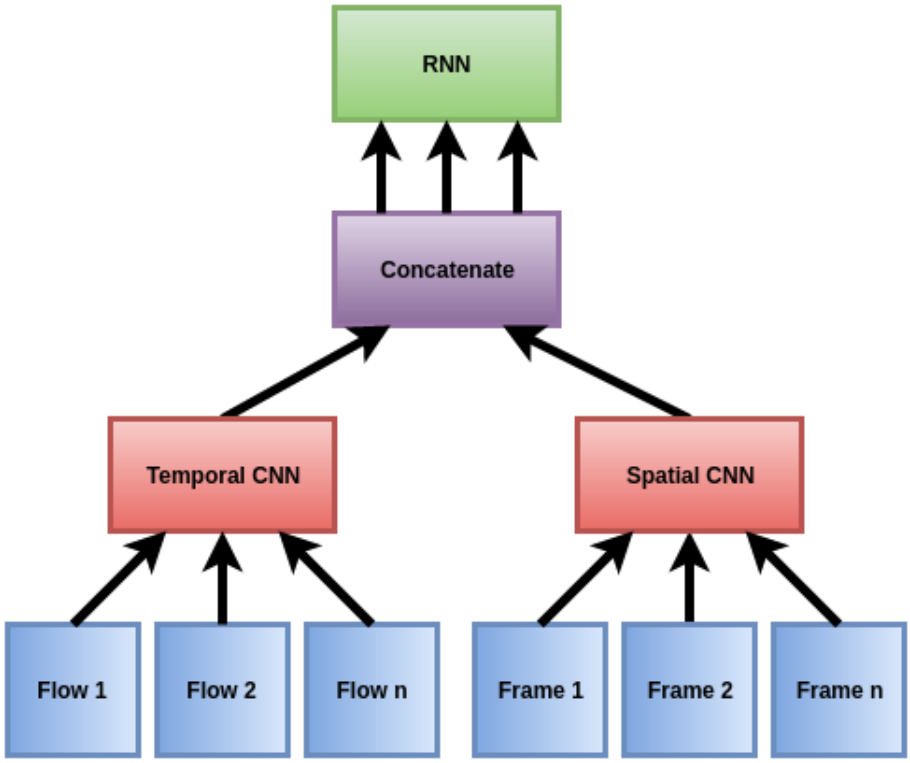


Figure 3.3: 3D CNN + RNN

3.4 ConvRNN

The architectures we have discussed until now process the spatial information first and the temporal information later. In the video however these two are inherently coupled. An architecture which processes both spatial and temporal information simultaneously is necessary for effectively processing videos.

ConvRNNs are convolutional networks in which there are recurrent connections. The equations for activations would be the same a that of an RNN but with matrix multiplications replaced by convolutions. An example of such an architecture is the ConvLSTM[12] whose

equation for each layer would be:

$$i_t = f(W_{xi} * x_t + W_{hi} * h_{t-1} + W_{ci} \circ c_{t-1} + b_i)$$

$$f_t = f(W_{xf} * x_t + W_{hf} * h_{t-1} + W_{cf} \circ c_{t-1} + b_f)$$

$$o_t = f(W_{xo} * x_t + W_{ho} * h_{t-1} + W_{co} \circ c_{t-1} + b_o)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ g(W_{xc} * x_t + W_{hc} * h_{t-1} + b_c)$$

$$h_t = o_t \circ h(c_t)$$

Observer that these equations are very similar to the equations of LSTM, but with matrix multiplication replaced with convolutions. A variant of ConvRNN might include extra connections between neighbouring neurons of a layer so as to give each neuron more spatial context.

Chapter 4

Related Work

4.1 Hand Crafted Features

Before the recent popularization of deep learning, expert designed hand crafted features were the state of the art for activity recognition. This involves extracting features from the video, using an encoding scheme to encode these features and using a discriminative classifier such as SVM for classification.

Hand crafted features include Space Time Interest Points, Cuboids, Dense Trajectories and Improved Trajectories. Among these features, improved trajectories with rich descriptors of Histogram of Oriented Gradient (HOG), Histogram of Optical Flow (HOF), and Motion Boundary Histograms (MBH) have been shown to be successful on a number of challenging datasets. The extracted trajectories are located at regions with high motion salience, which contain discriminative information for action recognition. The feature descriptors are computed along the trajectories of feature points to capture shape, appearance, and motion information. The features are then encoded into a Bag of Features representation using a learned k-means dictionary. A classifier such as an SVM is trained on the quantized features to distinguish the video classes. Improved Dense Trajectories (IDT)[7] is the state of the art method which uses only hand crafted features and achieves high accuracy.

4.2 Deep Learning Approach

Convolutional Neural Networks are a biologically inspired class of deep learning models that replace all three stages with a single neural network that is trained end to end from raw pixel values to classifier outputs. Thus, these architectures effectively shift the required engineering from feature design and accumulation strategies to design of the network connectivity structure and hyperparameter choices.

3D convolutional networks [3] are good feature learning machines that model appearance and motion simultaneously. The features from these 3D CNNs encapsulate information related to objects, scenes and actions in a video, making them useful for various tasks without requiring to fine tune the model for each task. 3D CNNs have also been used in conjunction with LSTM for activity recognition[8]. [9]

The Slow Fusion [11] model slowly fuses temporal information throughout the network such that higher layers get access to progressively more global information in both spatial and temporal dimensions. This is implemented by extending the connectivity of all convolutional layers in time and carrying out temporal convolutions in addition to spatial convolutions to compute activations. Slow Fusion model uses 3D convolutions and average pooling in its first 3 convolution layers however it loses temporal information after the third convolution layer because they use 2D convolutions in the higher layers.

The Long-term Recurrent Convolutional Networks (LRCN)[13] model combines CNNs and RNNs. It is a powerful and general model applicable to visual time-series modelling. It has been used to solve three vision problems , activity recognition, image description and video description. For the task of activity recognition, T individual frames are inputs into T CNNs which are then connected to a single-layer LSTM.

Another approach is to use Two Stream CNN[5]. One stream trained on video frames to capture spatial features and the other on stacked dense Optical Flow vector channels to capture the temporal features. Two steam CNNs achieve state of the art accuracy for activity recognition when using only deep learned features.

4.3 Hybrid Approach

The state of the art method for activity recognition combines both hand crafted features and deep learned features. Trajectory-pooled Deep-Convolutional Descriptor (TDD)[14] integrates the key factors from two successful video representation, IDT and 2 Stream CNNs. The 2 stream CNN is used as a feature extractor. A set of point trajectories are detected with the method of improved trajectories.Based on CNN features and improved trajectories, the local CNN responses over the spatio-temporal tubes centered at the trajectories are pooled. The resulting descriptor is called TDD.Fisher vector representation for aggregation is applied to these local TDDs over the whole video to form a global super vector, and use linear SVM as the classifier to perform action recognition.

Chapter 5

Problem Formulation

Human activity recognition in untrimmed videos requires continuous recognition of activities, detecting starting and ending times of all occurring activities from an input video. The general human activity recognition problem can be stated as a sequence labelling problem.

Given a video X which is a sequence of T frames $(x_1, x_2, ..., x_T)$, find the corresponding sequence of actions $(y_1, y_2, ..., y_{T'})$ where $y_i \equiv (c_i, s_i, e_i)$. Here $c_i \in C \cup \{\epsilon\}$. C is the set of action classes, ϵ is when no activity occurs. s_i is the frame number when the activity begins and e_i is when it ends.

Training a single neural network for this task is time consuming and since each frame of the video has to be propagated through the network every time, it would be slow as well. We propose two neural networks which work together to achieve this task. The first one we call Activity Detection Network (ADN) and the second one is Activity Recognition Network (ARN).

ADN would scan through the video to predict the temporal extents of an activity of interest. This is a sequence labelling problem. Given the frames $(x_1, x_2, ..., x_T)$, label each frame as either ϵ meaning no activity of interest occurs or μ meaning that there is an activity of interest.

ARN would only take as input the frames labelled μ by ADN and predict the activity occurring in them. This is a sequence classification problem. Given the frames $(\mu_1, \mu_2, ..., \mu_{T'})$, find the corresponding activity $c_i \in C$.

Chapter 6

Approach

Our approach to the task of recognizing activities in untrimmed videos is to split it into two independent tasks, each performed by a separate neural network. We describe these neural networks in detail below:

6.1 Activity Detection Network (ADN)

The job of the ADN is to predict the frames in which an activity of interest occurs from the input video. It does not recognize the activity itself, it only labels the frame as containing an important activity μ or as not containing an any interesting activity ϵ. Since our system should be able to process long videos, the ADN has to be able to process each frame fairly quickly. For this we design a neural neural network which contains fewer number of convolution layers than conventional CNNs architectures. We use a simple 2D-CNN + LSTM architecture with the CNN being shared by the frames in the same sequence.

Training phase of the ADN: This network is trained on the entire video. Given a video of total number of frames L and the temporal extents of the activities in it, break it into frame sequences of length T. These may be generated by a moving window of size T with a specified stride S. There would be $\left\lfloor \frac{L-T}{S} \right\rfloor + 1$ such frame sequences generated for a video of L frames. The ground truth for a frame would be 1 if it belongs to the temporal extent of an activity otherwise it is 0. The ground truth is also similarly broken into sequences. This would be the ground truth for the corresponding input frame sequence. Since this is a

one-to-one sequence binary labelling task, the binary cross-entropy error is used.

Prediction phase of ADN: Given a video of total number of frames L, break it into frame sequences of length T. Predict the labels of a sequence using the ADN. If a frame is predicted as 1, it is a μ frame implying it belongs to an activity. As a video could consist of different activities, the μ frames which occur contiguously are stacked together and constitute the input to the ARN.

6.2 Activity Recognition Network (ARN)

The ARN predicts the activity occurring in contiguous μ frames. It uses the flow image and RGB image stacked together to create an input volume of 6 channels. This incorporates the characteristics of a two-stream CNN but by using only a single 2D-CNN, called the spatio-temporal CNN. We use a spatio-temporal CNN + LSTM to predict the class of the activity occurring in those frames. The ARN is significantly deeper than the ADN in terms of the depth of the spatio-temporal CNN used. This is required as the task of activity recognition involves significantly more number of CNN layers to achieve good results. Since activities of interest in videos only make up a part of the video's length, the ARN is invoked on mostly short sequences. The ARN and ADN can be pipelined so that the entire system can process videos quickly.

Training phase of ARN: This network is trained only on the frames which fall within the temporal extent of an activity. Similar to the training phase of ADN, the input frames are broken into frame sequences of length T. The ground truth consists of only the activity class occurring in the temporal extent of the particular frame broken into sequences. This is a sequence classification task and uses the categorical cross entropy error function.

Prediction phase of ARN: The contiguous μ frames labelled by ADN are assumed to constitute a single activity. These are broken into frame sequences of length T and fed into ARN. The predictions of ARN are fused across the frame sequences to predict the class of the activity.

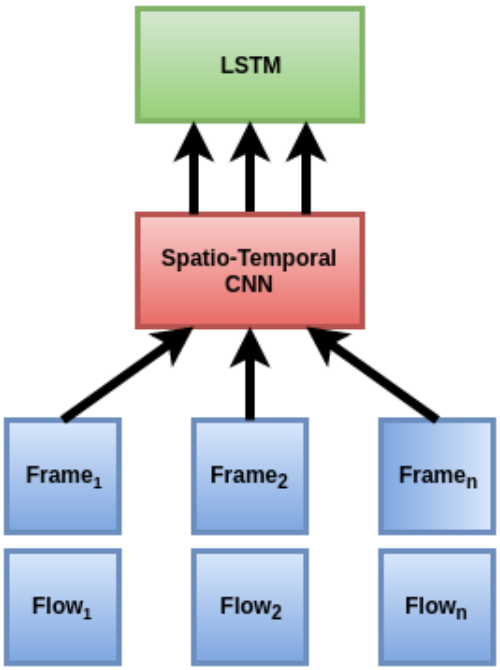


Figure 6.1: 3D CNN + RNN

6.3 Data and Pre-Processing

We train ADN on 2 types of inputs. We use raw RGB frames as a baseline to compare accuracies for ADN. Frames which have been back-ground subtracted and thresholded are used as this will significantly reduce the problem space and may be trained on smaller networks , which is ideal for ADN. The ARN is also trained on two types of inputs. Raw RGB frames to form a baseline and the six channel volume formed by stacking RGB and Flow images to contain both spatial and temporal information in the same frame input.

Chapter 7

Dataset

Understanding the intricacies of the dataset is a crucial part of any learning task. Several Human Activity recognition datasets are available but they typically too small for CNNs to give significant results. Few large video datasets are available, but consist of short videos having a single activity. UCF-101 [6] is a moderate size activity dataset having 101 different activity classes and 13320 videos. An associated dataset is the validation dataset of THUMOS'15[4] which contains temporally untrimmed videos. For 20 classes which are common to both UCF-101 and THUMOS'15 validation, temporal annotations are provided. We use a dataset consisting of videos from these 20 classes taken from both UCF-101 and THUMOS'15. For all the tables below, we sample the videos as 10 frames per second.

The details of the dataset are as follows:

	Number of Videos	Number of Frames	Number of Frames having Activity
THUMOS'15	412	877976	275733
UCF-101	2765	138659	138659
Total	3177	1016635	414392

Table 7.1: Outline of dataset

Since there are only 412 videos with temporal annotations, we augment this dataset with videos from the UCF 101 dataset belonging to the 20 activity classes making the assumption that The class wise details of activities in our dataset are as follows: We report the median length here because it a more robust estimate of the class center than mean.

Video Class	Number of Activity Instances	Median Activity Length(Frames)
BaseballPitch	242	32.5
BasketballDunk	916	18.0
Billiards	340	52.0
CleanAndJerk	312	81.5
CliffDiving	507	32.0
CricketBowling	518	24.0
CricketShot	611	22.0
Diving	1027	30.0
FrisbeeCatch	282	33.5
GolfSwing	228	47.5
HammerThrow	568	71.5
HighJump	543	46.0
JavelinThrow	503	56.0
LongJump	448	62.0
PoleVault	663	64.0
Shotput	371	38.0
SoccerPenalty	249	35.0
TennisSwing	379	27.0
ThrowDiscus	321	43.0
VolleyballSpiking	384	23.0

Table 7.2: Outline of dataset

From the videos, we sample frames at a rate of 10 frames per second. We create sequences consisting of 10 consecutive frames. The dataset consists of 110,000 sequences. We say a sequence Contains Activity if all the frames in it belong to some activitys temporal extant, No Activity if none of the frames belong to an activitys temporal extant and Transition if an activity begins or ends during the frame sequence.

Dataset	Total Sequences	Contain Activity	No Activity	Transition
Train	100,000	39,145	55,951	4,904
Test	10,000	4,102	5,753	145
Total	110,000	43,247	61,704	5,049

Table 7.3: Sequence Profile

Chapter 8

Results and Discussion

We evaluate the accuracy and performance of ADN and ARN separately and then evaluate the overall system's accuracy at the end. We train and test our models on Nvidia K20 GPU using Keras library. Keras is a high level and modular neural network library which can work on both Theano and Tensorflow. We use Open CV for processing our images and numpy for processing the dataset. The dataset is stored using HDF5 using h5py library.

8.1 Activity Detection Network

Since ADN solves a one to one sequence labelling problem, the ground truth y and prediction y' are both binary strings of length equal to the sequence length T. The accuracy of the prediction is defined as:

$$SampleAccuracy = \frac{\sum_{i=1}^{T} xnor(y_i, y'_i)}{T}$$

Where $xnor$ is the bitwise negation of exclusive OR function. the If there are n samples in the dataset, the overall accuracy is:

$$Accuracy = \frac{\sum_{j=1}^{n} \sum_{i=1}^{T} xnor(y_{ni}, y'_{ni})}{nT}$$

A video sequence is correctly labelled if all the frames in it are classified correctly. We

train on 100,000 sequences and test on 10,000 sequences, each of length 10. The training time per epoch of the dataset is about 6 hours. Due to limited time, we train for only 10 epochs.

The accuracies(a) of the ADN are as follows:

Dataset	$a = 1$	$0.5 \leq a < 1$	$0 < a < 0.5$	$a = 0$
Train	67,031	19,591	12,175	1,203
Test	6,152	1,742	1,682	424

Table 8.1: Accuracies of ADN

Accuracy class	Total	Contain Activity	No Activity	Transition
$a = 1$	6,152	1,603	4,516	33
$0.5 \leq a < 1$	1,742	1,434	253	55
$0 < a < 0.5$	1,682	1,008	623	51
$a = 0$	424	57	361	6
Total	10,000	4,102	5,753	145

Table 8.2: Accuracies of ADN

Our model labels 61.52% of the sequences correctly, 17.42% of the labels with an accuracy of over 0.5 and 16.82% with accuracy less than half. Most of the sequences which have no activity have been correctly labelled. Sequences which do contain activity have been labelled correctly with accuracy atleast half. Sequences which contain transitions were mostly recognized with accuracy more than half and also with an accuracy of one.

8.2 Activity Recognition Network

The ARN solves a sequence classification task. The accuracy of this classifier would simply be the number of sequences which have been classified correctly divided by total number of sequences n. The ARN is trained and tested on the 20 class subset of UCF-101 videos. The training time per epoch is about 24 hours. We train our model for 20 epochs

The accuracies for certain classes which are easily discernible from the others are much higher. For instance Billiards is much different from all other activities and so achieves high accuracy. Shotput, ThrowDiscus and HammerThrow have similar actions and hence have

lower accuracy. These three activities have similar motion and also similar background. Subtracting the background might be helpful in this case, but it has been found that the background also plays an important in activity recognition.

Class	Number of videos	Accuracy (%)
BaseballPitch	150	82.5
BasketballDunk	131	95
Billiards	150	97.2
CleanAndJerk	112	83.3
CliffDiving	138	85.7
CricketBowling	139	87.1
CricketShot	167	85.8
Diving	150	86.8
FrisbeeCatch	126	84.9
GolfSwing	139	88.5
HammerThrow	150	76.2
HighJump	123	72.6
JavelinThrow	117	94.2
LongJump	131	95.9
PoleVault	149	87.8
Shotput	144	71.4
SoccerPenalty	137	94.3
TennisSwing	166	87.8
ThrowDiscus	130	71.1
VolleyballSpiking	116	92.9
Total	2765	86.01

Table 8.3: Accuracies of ARN

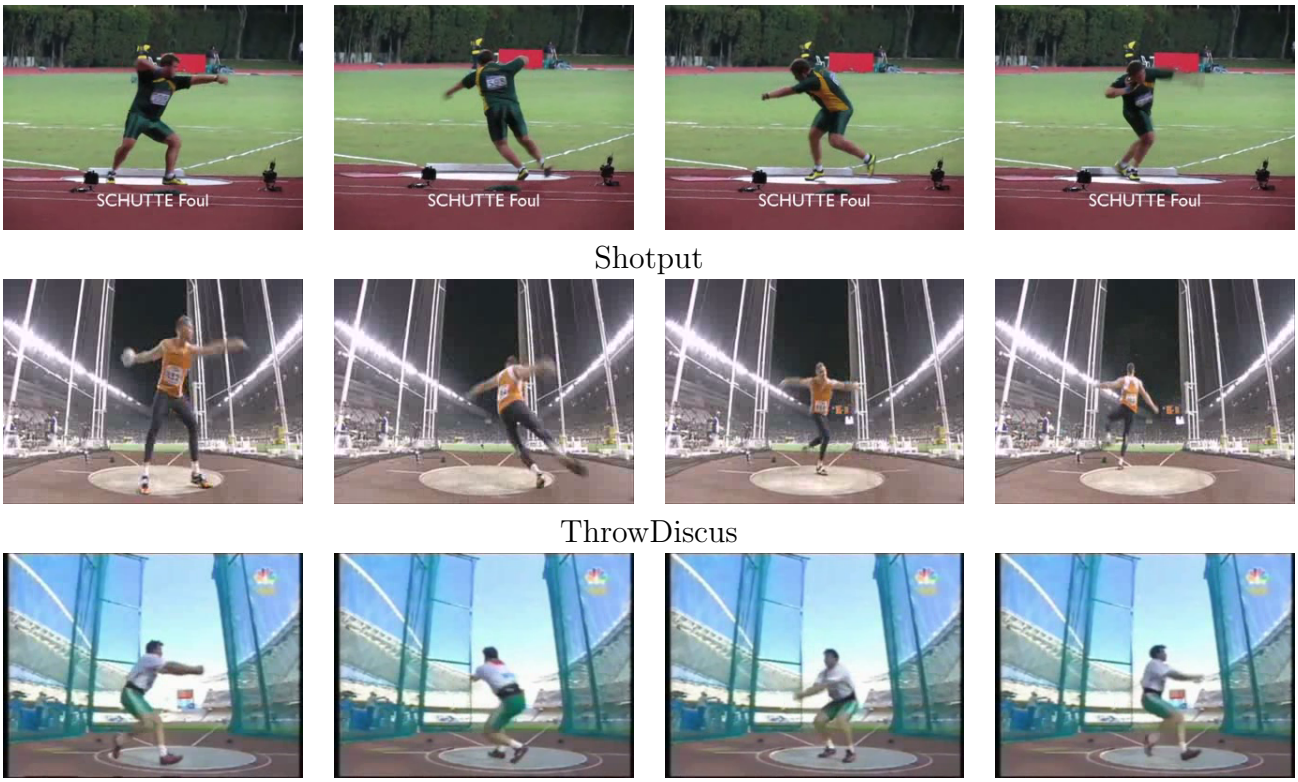


Figure 8.1: Actions having similar motion

8.3 ADN - ARN System

The frames belonging to temporal extents predicted by ADN are used as inputs to the ARN. The processing time for a video of length 1 minute which contains a 10 second activity is about 120 seconds by the ADN and 30 seconds by the ARN when running using on a CPU. When running on GPU, it is much faster, making the system capable of processing streaming videos.

We test our system on 100 long videos which contain multiple instances of the same activity, The ADN - ARN system accuracy on untrimmed videos is as follows:

We observer that for only in 14% of the videos, all the activity instances are completely recognized. For 64% of videos, atleast one instance is recognized and for 34% of videos no instance is being recognized. This can be attributed to the fact that the ARN is sensitive to the output of ARN. If the ADN returns a false positive or false negative frame sequences,

Total	All instances recognized	At least one but not all instances recognized	No instance recognized
100	14	52	34

Table 8.4: Accuracies of ADN - ARN system

the ARN fails to recognize it. The 100 videos consist of a total of 295 instances of activities. Of these 146 instances are correctly recognized, giving our system an accuracy of 49.49% for our system.

Chapter 9

Future Work

9.1 Dataset Improvement

Deep neural networks work well when there is a large amount of data. The 20 action dataset we used from UCF-101 and THUMOS'15 validation could be considered a small to medium sized dataset at best. The UCF 101 subset of the data contains only short videos which have been assumed to contain the activity only. The THUSMOS'15 validation subset contains long videos with temporal extent annotations for activities in the video. The original datasets were specifically tailored for the task of activity recognition and not for activity detection and activity recognition.

ActivityNet[15] is a new dataset which contains 10,024 untrimmed videos having 200 activity classes with over 15,410 instances of activities. This dataset is ideal for a more rigorous evaluation of our system. The ActivitNet Challenge aims become the benchmark dataset for new algorithms and techniques in human activity understanding. This challenge is tailored to 200 activity categories in two different tasks.

- Untrimmed Classification Challenge: Given a long video, predict the labels of the activities present in the video
- Detection Challenge: Given a long video, predict the labels and temporal extents of the activities present in the video.

Our system is theoretically capable of solving both problems simultaneously.

9.2 Model Improvement and Further Experimentation

Promising new deep models which are suitable for processing complex spatio temporal information are being developed by the computer vision community. One such model is the ConvLSTM which has been used for precipitation nowcasting and is an ideal candidate for experimentation with video activity recognition datasets.

Several of the pre-existing models could also be combined to create models which can process videos. Some examples of possible models we could try are 3D-CNN + LSTM, several variants of 2 stream architectures are possible like using 3D-CNNs, using only a single CNN for both image and flow etc, most of which could be combined with a LSTM layer as well.

Chapter 10

Conclusions

We develop a system which is capable of detecting activities and also recognizing them in untrimmed videos. We discuss about several deep neural network architectures which could be used to process spatio-temporal information such as videos. We develop two deep neural networks based on these architectures. One is trained for activity detection and the other for recognition. A special dataset created by considering 20 classes from UCF-101 and THUMOS'15 validation is used for training and testing our system. The neural networks function independently and achieve good accuracies for their tasks. Since the second neural network is sensitive to the outputs of the first, the overall accuracy of the system is moderate. As future work, we would like to explore better models like ConvLSTM and new dataset like ActivityNet.

Bibliography

[1] Graves, Alex. Supervised sequence labelling. Springer Berlin Heidelberg, 2012.

[2] J. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. ACM Comput. Surv., 43(3):16, 2011.

[3] S. Ji, W. Xu, M. Yang, and K. Yu. 3D convolutional neural networks for human action recognition. TPAMI, 35(1), 2013.

[4] Y.-G. Jiang, J. Liu, A. Roshan Zamir, I. Laptev, M. Piccardi, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes, 2013.

[5] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. NIPS, 2014.

[6] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. CoRR, abs/1212.0402, 2012.

[7] H. Wang and C. Schmid. Action recognition with improved trajectories. In ICCV, 2013

[8] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Action classification in soccer videos with long short-term memory recurrent neural networks. In Proc. ICANN, pages 154159, Thessaloniki, Greece, 2010. 2

[9] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential Deep Learning for Human Action Recognition. In 2nd International Workshop on Human Be- havior Understanding (HBU), pages 2939, Nov. 2011.

[10] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Computing, 9(8):17351780, Nov. 1997.

[11] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convo- lutional neural networks. In CVPR, 2014.

[12] Shi, Xingjian, et al. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting." arXiv preprint arXiv:1506.04214 (2015).

[13] Donahue, Jeffrey, et al. "Long-term recurrent convolutional networks for visual recognition and description." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[14] Wang, Limin, Yu Qiao, and Xiaoou Tang. "Action recognition with trajectory-pooled deep-convolutional descriptors." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[15] Caba Heilbron, Fabian, et al. "Activitynet: A large-scale video benchmark for human activity understanding." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.