Md. Mehadi Hasan

Department of Computer Science & Engineering

Begum Rokeya University, Rangpur

Supervisor

Dr. Md. Ileas Pramanik

In partial fulfillment of the requirements for the degree of

B.Sc(Engg.) in Computer Science & Engineering

February 27, 2021

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor **Assistant Professor Dr. Md. Ileas Pramanik** for the continuous support of my study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

My sincere thanks also goes to **Associate Professor Dr. Md. Mizanur Rahoman**, who provided me an opportunity to join his team as intern, and who gave access to the laboratory and research facilities. Without him precious support it would not be possible to conduct this research.

With best regards,
Md. Mehadi Hasan.

Dedicated to all the students of 𝔻epartment of ℂomputer Science & 𝔼ngineering, 𝔹 ℝ 𝕌 ℝ

CERTIFICATE

This is to certify that Thesis Report entitled **Deep Action:A novel action recognition using wavelet Transformation** which is submitted by **Md. Mehadi Hasan, ID: 1209020, Session:2012-2013** in partial fulfillment of the requirement for the award of degree B.Sc. (Engineering) in Com- puter Science and Engineering to Begum Rokeya University, Rangpur is a record of the candidate own work carried out by him under my supervision. The context in this report is original and has not been submitted for the award of any other degree.

.....

Signature of the Supervisor

Abstract

Action recognition is to translate various object activity or action from video data. Convolutional neural network (CNN), trajectories analysis, temporal analysis, two-stream convolution network are the different types of techniques that have been used to solve this problem. Among these approaches, the two-stream convolution network performs better. It combines both CNN and temporal features on a single model. CNN is good at getting spatial features of an image and temporal analysis is very good at motion understanding from the video. However spectral features of the image are very important for image processing tasks which is mostly lost on CNN. Losing spectral features makes action recognition difficult to classify similar types of actions. Taking both spatial and spectral features of the image can be a booster to solve this problem. In this paper we propose a novel three-stream hybrid architecture that combines wavelet CNN, ConvNet and temporal network together. Our intuition is a spectral feature of the image combined with spatial and temporal in one model that will improve the performance of action recognition from video. For our experiment, we will use UCF-101 which is benchmarks datasets for video action recognition task.

Keywords: Action Recognition, Wavelet Transformation, ConvNet, Temporal analysis

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Action recognition in video data becomes a very interesting research topic in recent years. Action recognition is to translate various object activity or action from video data. In action recognition task input is a video clip and output is to predict what action is performing on that video clip. For example, a short video clip is shot on a football player playing football. If we feed this video clip to an action recognizer framework it will predict a label "playing football".

Video is nothing but a collection of still images. In recent years, Convolutional Neural Network for image recognition and feature extraction[1][2] perform very promising result. Though the video is a collection of still images convolution neural network does not perform well enough on video data for action recognition. One of the main drawbacks of CNN's is it only cares about the spatial feature but it skips most of the spectral features of the image. Two stream[3] architecture use image spatial stream ConvNet and temporal stream ConvNet to classify action but they missed the spectral information of image which is important features of images for action recognition because of CNN spatial behavior.

Due to the duel behavior of video that is to understand still image information and motion information of consecutive frames makes the action recognition task very challenging. A particular action is the collection of movement of an object in a video. So we need both individual frame information and movement of the object that appears in the video. This can be achieved by taking spatial, spectral and temporal information of images

We propose a hybrid architecture **Figure 3.1** that combines spatial, spectral and temporal analysis in a single model. The spatial, spectral and temporal information of images are extracted respectively ConvNet, Wavelet[4] Transformation and temporal ConvNet. Temporal analysis performs better to represent the movement of the object in video data. We use both ConvNet and Wavelet because ConvNet is usually good at capturing spatial features, while a spectral analysis is very good at capturing scale-invariant features of the image. We consider both the spatial and spectral information of the image so that it captures both types of features. We will demonstrate that taking both spatial and spectral feature and combine it with temporal in a single model improve the performance of action recognition in video data.

Chapter 2

Background

Summary

In this section, we will introduce differently terminology, theory, and programming language and their frameworks used to implement this system. In recent years, artificial intelligence (AI) has been a hype. Scientist related to AI, practitioners, Cognitive Science community says AI is the third industrial revolution after the invention of electricity in the scientific community. Machine learning, deep learning, and AI come up in countless articles, often outside of technology-minded publications. Consequently, we can see the application of AI in different domains. For example Semantic Web, Automatic recommended system, chatbots, self-driving cars, and google, assistants, weather forecast, automatic fault detection, this list increase day by day. In this section, our discussion topic is deep learning, not AI or machine learning. Our background studies are divided into two major parts. 1^{st} **part** is definitions and little intuition on various terms related to this research and 2^{nd} **part** describe related works in this research field.

2.1 Deep Learning

From **Figure 2.1** we can see that deep learning (DL) is special field of machine learning. In deep learning the world **"Deep"** refers to it use layered architecture. there can be many layers in a deep neural network. This why Deep keyword is used in this type of learning. Deep learning also called layered representations learning

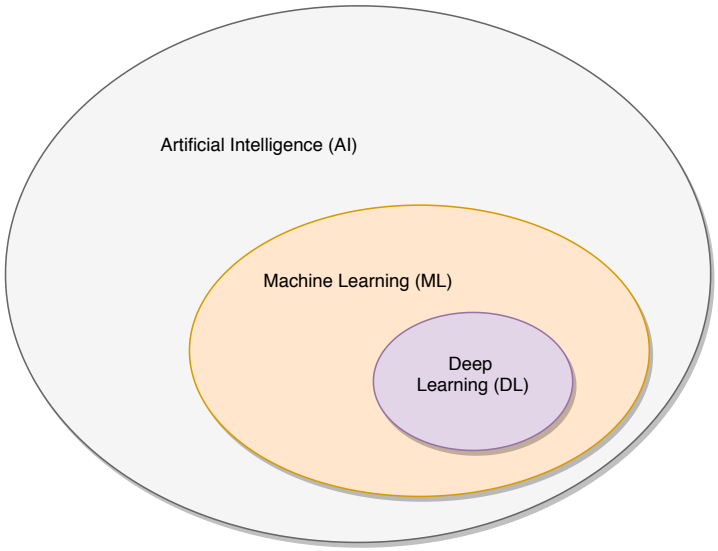

Figure 2.1: Artificial Intelligence, Machine Learning, and Deep Learning

or hierarchical representations learning. The main difference of deep learning from machine learning is that it removes hand craft feature extraction from data. Hand craft feature engineering is very difficult and time consuming task. So automatic feature extraction is main success of deep learning algorithm. And this why deep learning becoming so popular now a days. But the the problem with deep learning algorithm is it requires large number of sample to extract good feature from data. But the good news is over the past few years the scientific community create some large data for deep learning research, for example ImageNet [2] data set for image classification problem. Which contain millions of images.

2.2 Artificial Neuron

Artificial neuron is the core component of any neural network. The Neuron holds problem relative information which is very important for the prediction of the right answer to the problem. A basic artificial neuron has the following structure

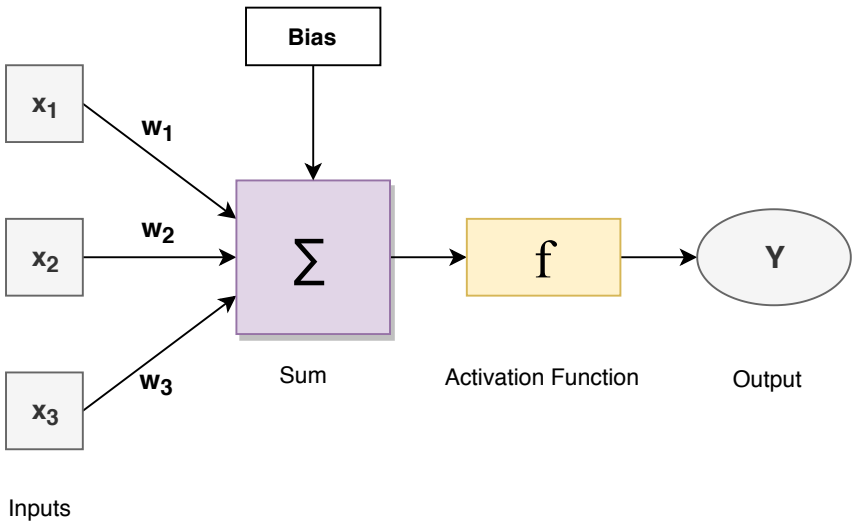

Figure 2.2: A Basic Artificial Neuron

like this one. From **Figure 2.2** we can see that it has three inputs x_1, x_2, x_3 and corresponding three weights w_1, w_2, w_3 and a bias weighted sum is calculated and bias is added to it and this output is go thought a activation function (f). Y is the output of activation function. The activation function apply the non-linearity to it's input value

Here weights and bias is the learnable parameter in the neuron. Initially weights and bias of the neuron is set randomly, but on training time it update weights and bias value to a optimal point. Neuron update it's weights and bias value by a algorithm called backpropagation.

2.3 Neural Network

Neural networks are the networks, which is very much simplified version human brain. Human brain is the collection of neurons, and the neurons can transfer information from one neuron to another neuron. The artificial neuron shown in 2.2 works similar to brain to transfer information from one neuron to another neuron. The core component of any Neural Network is it's artificial neuron.

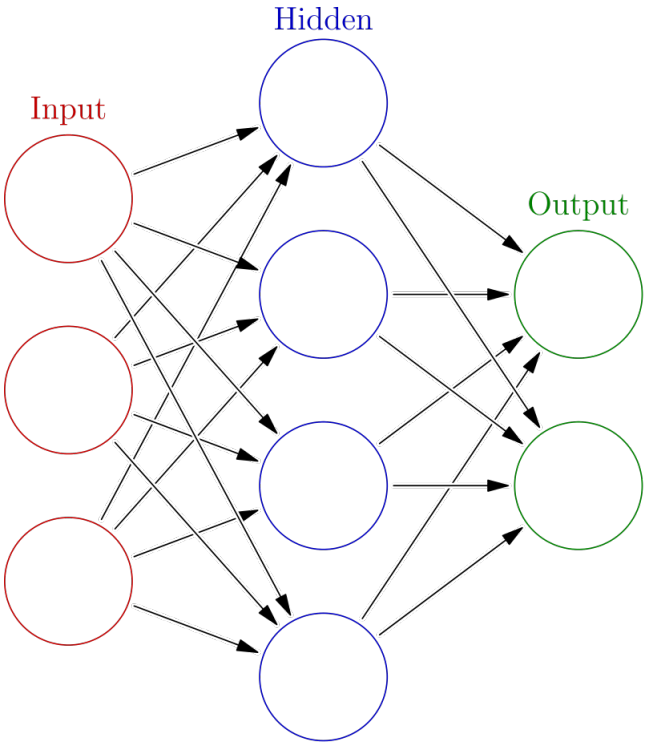

Figure 2.3: A Simplified Neural Network

Figure 2.3 [1] shows a simplified neural network. Here we can see it's collection of artificial neurons and it has three layers,

- **Input Layer:** Input layer has three neurons. It is the input to the neural network.

[1] **Image source:** `https://en.wikipedia.org/wiki/Artificial_neural_network`

- **Hidden Layer:** Hidden layers are the layers that present between input and output layer.This hidden layer has four neurons. A neural network can have multiple hidden layers.
- **Output Layer:** It is the output the neural network. this has two output neuron.

A neural network can have any number of neuron and any number of layers

2.4 Convolution Neural Network (CNN)

There are various types of neural networks for example **Perceptron (P), Feed Forward (FF), Radial Basis Network (RBF)** and so on. A Convolution Neural Network (CNN) is a spatial kind of neural network that perform convolutional operation of it's input data. Convolution operation perform better for any kind of 2D data. for example Image, 2D audio signals and so on.

2.4.1 Convolution Layers

In convolution operation a fixed size window is taken this window also known as filter, kernel and this filter moves over the 2D image and extract image feature like edges, lines, curves. A Convolution layer has two important parameters these are,

- **Stride:** Stride controls how the filter convolves around the input volume
- **Padding:** When a filter window hit the boarder of 2D image what it should do is guide by padding operation.

2.4.2 Pooling Layer

The task of pooling layer is to pool the data from input. A pooling layer is another building block of a CNN. Pooling layer function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network. Pooling layer operates on each feature map inde-

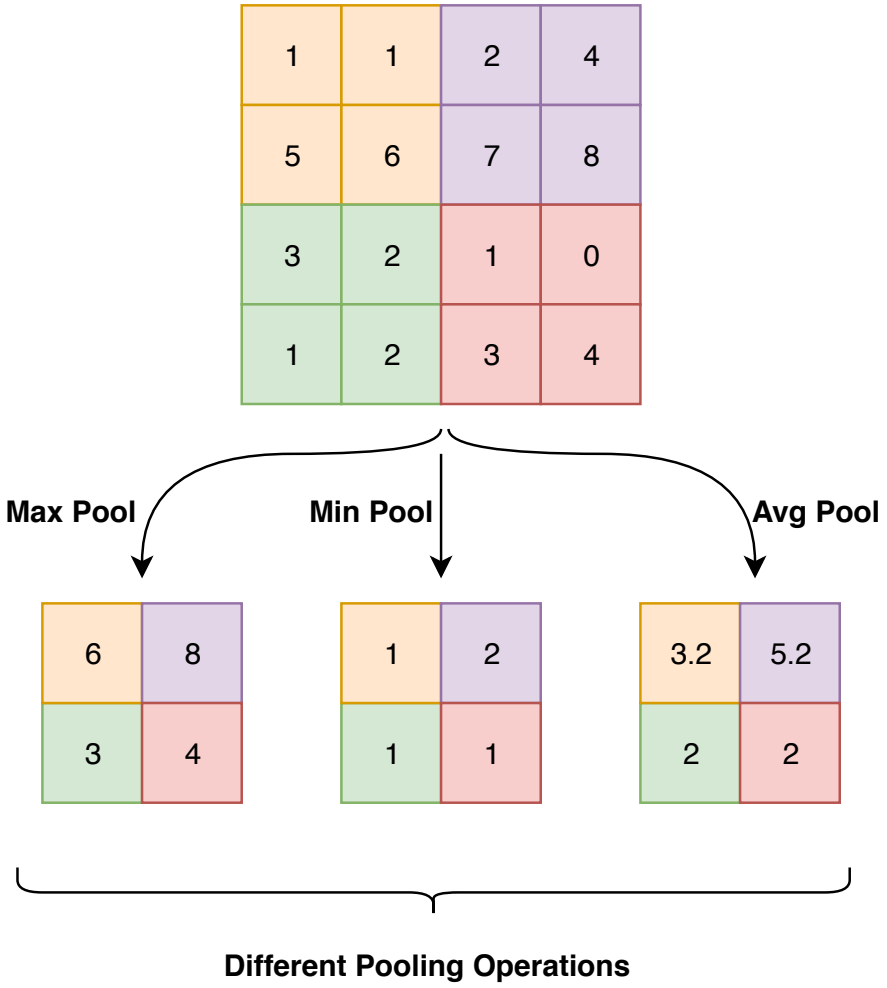

Figure 2.4: Max, Min and Avg pooling

pendently. **Figure 2.4** shows the operation of pooling layer. There are three pooling layer.

- **Max Pooling:** Takes the maximum value from the specified window.
- **Min Pooling:** Takes the minimum value from the specified window.
- **Average Pooling:** Takes the average value from the specified window.

The most common approach used in pooling is max pooling.

2.4.3 Activation Functions

Activation function apply the non-linearity to it's input. Convolution and Pooling layer is example of linear transformation. But linear transformation is not enough for solving complex problem. This why non-learn function is need to map complex problem. So there are six different activation function available for neu-

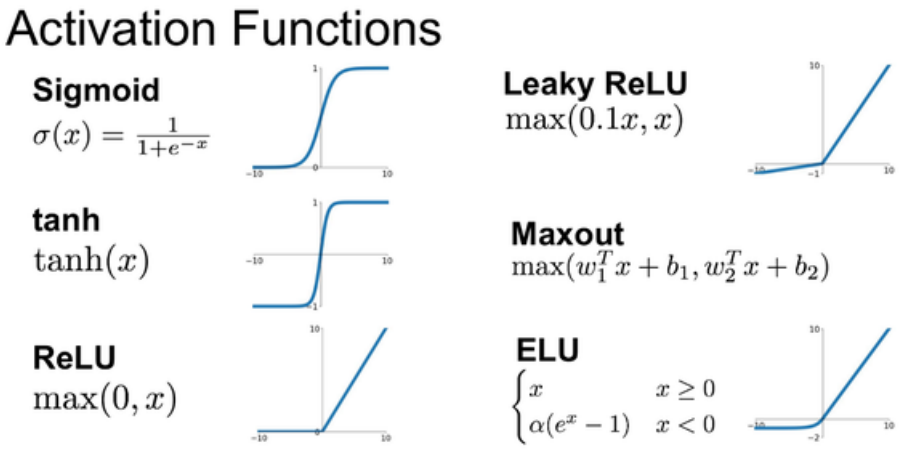

Figure 2.5: Various activation functions and their graphical representation

ral networks.**Figure 2.5**[1] show all the six activation function and it's graphical representation. Activation functions and their range,

- **Sigmoid:** Range is 0 to 1

[1]**Image source:** `https://images.app.goo.gl/YCLKRbWCk8ThwCXS9`

- **tanh:** Range is -1 to 1
- **Rectified Linear Unit (ReLU):** Range is $max(0, x)$
- **Leaky ReLU:** Range is $max(0.1 \times x, x)$
- **Maxout:** Takes $max({w_1}^T + b_1, {w_2}^T + b_2, ...{w_n}^T + b_n,)$
- **Exponential Linear Unit (ELU):** Same as ReLU except it takes negative value $>= -1$

2.4.4 Dense Layers

Dense layers is another building blocks of neural networks. All the neurons in this layer is densely connected to each other. **Figure 2.6** shows a basic densely

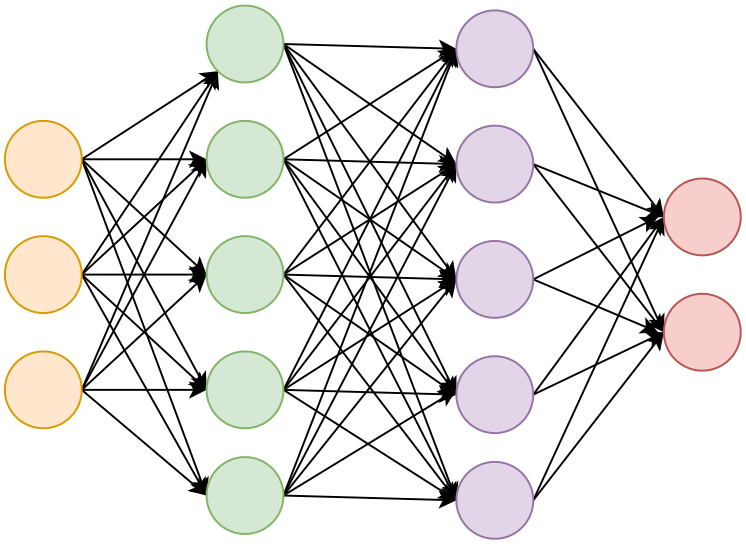

Figure 2.6: A Densely Connected Neural Network

connected neural network. Dense layer also called **fully connected layer**.

2.4.5 Dropout

Dropout is a technique by which we can solve **overfitting** problem with our neural network. Overfitting is a problem where network memorize all it's input to corresponding outputs. On other words it does not generalize to it's inputs. So when new data comes network fail to predict correctly. **Figure 2.7** shows

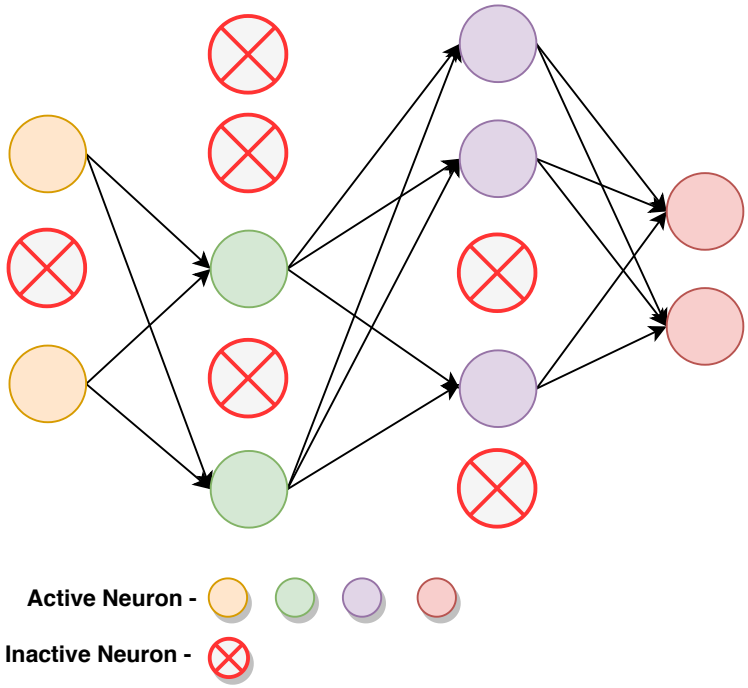

Figure 2.7: Dropout scenario. Read neurons are disconnected. It does not impact to total calculation of NN in particular iteration

dropout scenario of densely connected neural network shown in 2.6. The main mechanism of dropout technique is to deactivate some percentage of neuron in each training **iteration**. Consequently network can not memorize it's inputs to outputs rather it try to find a generalize solution of the problem.

2.5 Softmax Activation Function

Softmax is an activation function. Other activation functions include **ReLU** and **Sigmoid**. It is frequently used in classifications. Softmax output is large if the score (input called logit) is large. Its output is small if the score is small. It turns scores also known as logits into probabilities. In other words its outputs a probability distribution over the classes.

2.6 Optimizers

Optimizers update the weight parameters to minimize the total loss of neural network. Loss function acts as guides to the NN by telling optimizer if it is moving in the right direction to reach the bottom of the valley, the global minimum. There are several optimizer use in training a neural networks,

- **SGD** Stochastic Gradient Descent. It calculate gradient and using this grading it update weights values.
- **Adagrad** Adaptive Gradient Algorithm
- **Adadelta** Adadelta is an extension of Adagrad and it also tries to reduce Adagrads aggressive, monotonically reducing the learning rate
- **RMSProp** is Root Mean Square Propagation. It was devised by Geoffrey Hinton.
- **Adam** It is Adaptive Moment Estimation. It calculates the individual adaptive learning rate for each parameter from estimates of first and second moments of the gradients.

Adam optimizer is one of the most popular gradient descent optimization algorithms. In out experiment we use Adam as our optimizer.

2.7 Loss Functions

Loss function act as guide to the neural network. Its a method of evaluating how well specific algorithm models the given input data. If predictions deviates too much from actual results, loss function would cough up a very large number. And if predictions is close to the actual results, loss function would five a very small number. Bellow we will depict several loss functions and their mathematical representation,

Mean Square Error (MSE):

$$MSE = \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{n}$$

Mean Bias Error (MBE):

$$MBE = \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)}{n}$$

SVMLoss:

$$SVMLoss = \sum_{j \neq y_i} max(0, s_j - s_{y_i} + 1)$$

Cross Entropy Loss:

$$CrossEntropyLoss = -(y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i))$$

2.8 Forward Propagation

As the name suggests, the input data is fed in the forward direction through the network. Each hidden layer accepts the input data, processes it as per the activation function and passes to the successive layer. In order to generate some

output, the input data should be fed in the forward direction only. The data should not flow in reverse direction during output generation otherwise it would form a cycle and the output could never be generated. Such network configurations are known as feed-forward network. The feed-forward network helps in forward propagation.

2.9 Backward Propagation

The basic backpropagation algorithm is based on minimizing the error of the network using the derivatives of the error function.Calculation of the derivatives flows backwards through the network, hence the name, backpropagation. These derivatives point in the direction of the maximum increase of the error function A small step (**learning rate**) in the opposite direction will result in the maximum decrease of the (local) error function,

$$w_{new} = w_{old} - \alpha \frac{\partial E}{\partial W_{old}} \tag{2.1}$$

where,

$$\begin{aligned} w_{old} &= \text{Old weight} \\ w_{new} &= \text{New weight} \\ \alpha &= \text{Learning rate} \\ E &= \text{Error function} \\ \partial &= \text{Partial derivatives} \end{aligned}$$

The learning rate is important. If learning rate is too small it convergence extremely slow. On the other hand if learning rate is high it may not converge to the solution.

2.10 Metrics

Metrics are use for evaluating the performance of neural network models. There are several metrics used for measuring performance of deep learning model. Most common performance metrics are,

- Accuracy
- Confusion metrics
- Precision
- Recall
- Specificity
- F1 Score

Which performance metrics we should use is depend on problem domain. In our proposed system we use **Accuracy** for measuring performance of our model. Accuracy is a good measure when the target variable classes in the data are nearly balanced. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition:

$$Accuracy = \frac{Numberofcorrectpredictions}{Totalnumberofpredictions}$$

2.11 Multiresolution Analysis

Multiresolution analysis refers to any image processing technique where the source image is processed at multiple resolutions. For example: **Image** $\Rightarrow \frac{1}{2}$ **Image** $\Rightarrow \frac{1}{4}$ **Image** $\Rightarrow \frac{1}{8}$ **Image** $\Rightarrow \frac{1}{16}$ **Image** and so on. Now we can apply and combine filtering at all these resolution.

2.12 Related Works

In recent years numerous methods have been study for video action recognition. Many of them try to solve this problem by taking hand crafted features[5; 6; 7] and some use deep learning approach[8; 9; 10; 11]. Hand crafted local features for video recognition become popular and useful representation for video action recognition because of it has less video noise, lighting variation on image, background disorder. Thus network does not need to play attention on these challenge on video action recognition.

Cuboid [5] detector based on temporal Gabor filters for extraction of informative portion image image. Space time interest points [6] introduce Harris3D detector. A scale invariant spatio temporal interest point detector is proposed by [12]. It is base on Hessian saliency measurement. In the meantime few local descriptors have been proposed to constitute the 3D volumes extracted using interest points. Some of example of techniques are Histogram of Oriented Gradient (HOG), Histogram of Optical Flow (HOF) [13], 3D Histogram of Gradient(HOG3D) [14]. In [15]

Limitation of these local features is that absence of semantics and biased capacity. To overcome this lack, some mid-level and high-level video representations have been proposed for example: Actions [16], Action Bank [17], Motionlets [18], Mining motion atoms and phrases for complex action recognition [19], and Video action detection with relational dynamic-poselets [20]

Their technique is to adopt some heuristic mining methods to select biased visual elements as feature units. Instead, this paper takes a different view of this problem and replace these local hand-crafted descriptors with deep-learned rep-

resentations of video. Our solution automatically can learn high level informative video representation direct from data.

In recent years deep learning algorithm have achieved impressive success in image classification [21; 22; 23] and there have been a number of attempts to develop deep architectures for video action recognition [8; 9; 10].These deep models achieved lower performance compared to hand crafted feature extraction. There could be two reason for it, one is there is not enough data for deep learning, two: learning complex movement patters is more challenging.

A 3D ConvNets approach has been proposed in [24]. In more recent paper [12] show a scale invariant spatio-temporal architectures. In [3] A two-stream ConvNets architecture is design. It has spatial and temporal net. They use ImageNet weight for features extraction from data. And explicitly calculate optical flow for capturing motion information. Finally the use fusion technique to take advantage of both spatial and temporal net.

These deep models consider spatial and temporal features they does not consider spectral features of image. Which could be contributory for video classification problem.

Chapter 3

System's Architecture

Overview

We proposed a novel approach that combines Wavelet, ConvNet, and Temporal analysis. It can be achieved by training WaveletCNN, ConvNet and Temporal Network separately and finally calculate the softmax class score using the fusion method. In the following subsection, we explain our proposed architecture and it's different parts.

3.1 Architecture

We know that video is a collection of frames. We can not feed a video directly to a neural network because the neural network takes fixed-size input. But video data is continuous data. So we extract all the frames from video using frame extractor. The input to the frame extractor is a single video output is all the frame of that video. Now we can use these frames to training neural our networks

Figure 3.1 show the basic architecture of our proposed framework. We can

see the extracted frames from video is used. The single frame is directly used in WaveletCNN and Spatial ConvNet. But For temporal ConvNet, we can not use frames directly. To feed image data to temporal ConvNet we first extract optical flow from frames. we add an optical flow extractor before feed image data to temporal ConvNet. In our Experiment we use [25] for extracting optical

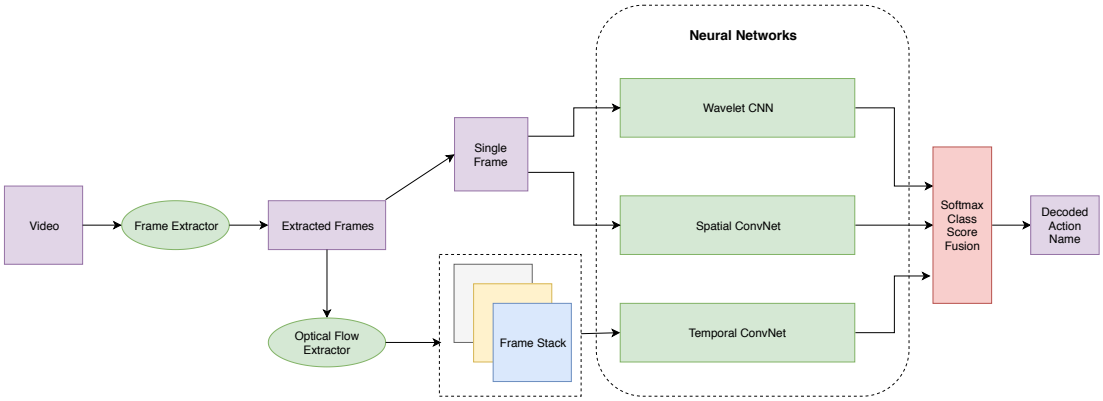

Figure 3.1: Overview of three-stream deep action architecture

flow of video data. We training each neural network individually after that use fusion method to combine the prediction of WaveletCNN, ConvNet and temporal ConvNet. The Basic architecture has four key components,

- Wavelet
- Optical Flow
- Temporal ConvNet
- ConvNet
- Fusion

Now we discuss each of them in the following section.

3.2 Wavelet

Wavelet is an image compression technique. There are different families of wavelet transformation. Here we use Wavelet Haar transformation in WaveletCNN model. The key advantage of this kind of transformation it captures both frequency and location information for data. In **Figure 3.2** we can see that a boy is throwing a

Figure 3.2: Input Image

basketball. If we apply Wavelet transformation is this image It will output four different images shown in **Figure 3.3**. These images have the following feature present in it,

- **Approximated** image from original input image.
- **Horizontal** detailed image, where horizontal feature of image will be visible.
- **Vertical** detailed image, where vertical feature of the input image will be visible.
- **Diagonal** detailed image, where feature of input image will be visible

Please check the **Figure 3.3** which is output of Wavelet Haar Transformation of input image show in **Figure 3.2** We can see that transformation output Approx-

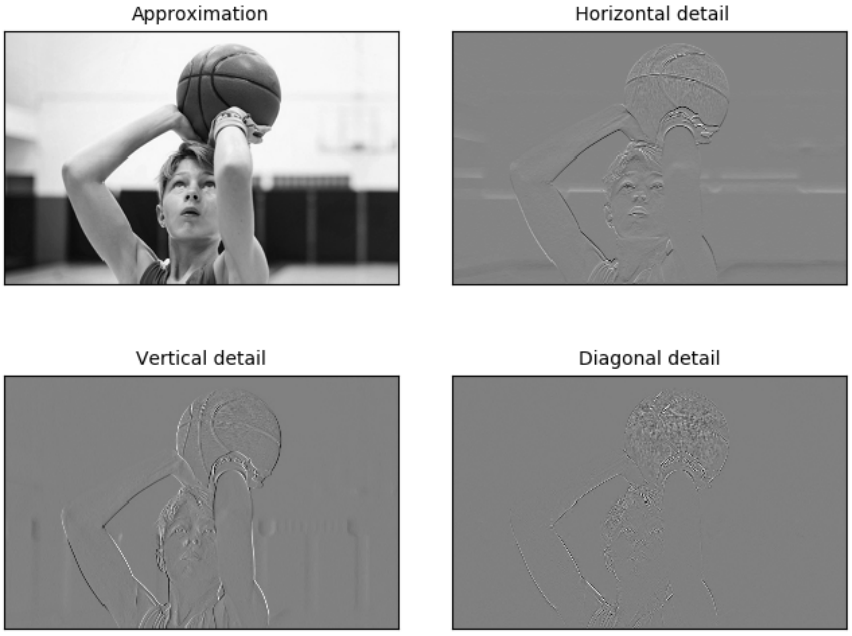

Figure 3.3: Wavelet Haar Transformation

imated image, Horizontal, Vertical, Diagonal detailed image of the input image. In our experiment for Wavelet CNN we use four level decomposition of input images. With each level of decomposition, the size of the input image will decrease by the power of two. For Example, if input image size is 224×224 after the first level of decomposition approximate image size will be 112×112, Second level of decomposition approximate image will be 56×56 and this process goes on according to level to wavelet decomposition .

The N (let's say N = 2) level decomposition of an input image can be view as follows. Approach showed in **Figure 3.4** also known as multiresolution analysis [26] (MRA) or multiscale approximation (MSA), which is the core design method

of wavelet transformation. In **Figure 3.4** shown the most common approach to the multilevel wavelet transform and it involves the further decomposition of only the approximation sub-band at each subsequent level. We can see that trans-

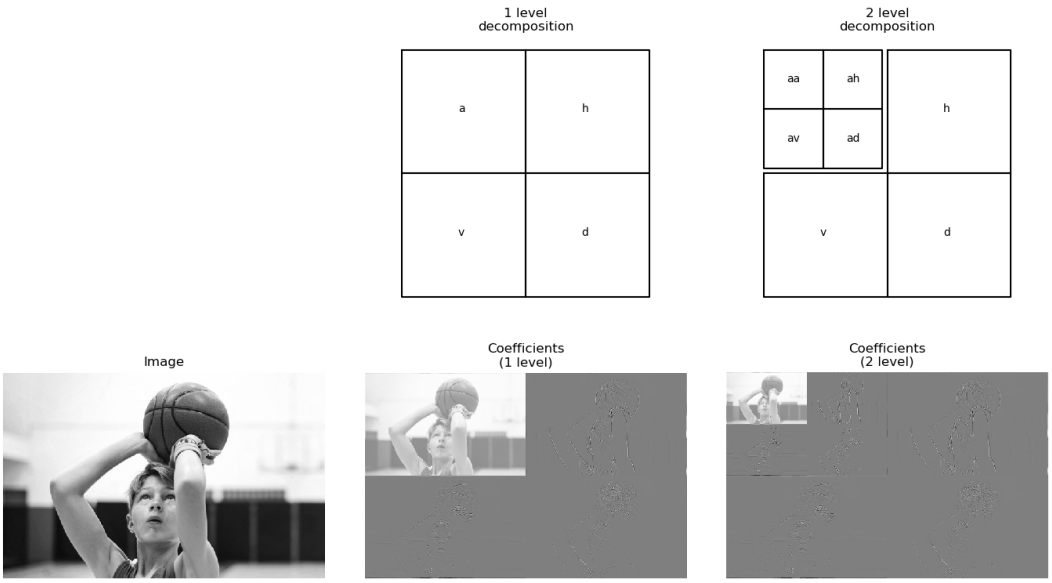

Figure 3.4: Two level Wavelet Decomposition

formation output Approximated image, For subsequent levels of decomposition, only the approximation coefficients for our example sub-band **'a'** will be used as an input for the second level of decomposition. In **Figure 3.4 'a'** indicate approximated image, **'h'** indicate horizontal featured image, **'v'** indicate vertical featured image and **'d'** indicate diagonal featured image.

So we can see in each decomposition level the approximate image is downsampled by the power of two with respect to image wide and height. The main advantage of this type of transformation is that we can go backward from any decomposition level. This is possible in each decomposition level it stores horizontal, vertical and diagonal information of the image. So we can say we can reconstruct the

original image from any level of decomposing image. And this is very helpful for any learning algorithm to understand better from image.

3.3 Optical Flow

Optical flow is the pattern that represent motion of objects, surfaces, and edges in a visual scene caused by the movement of camera or movement of object itself. To understand what activity is going on in a video it is very important to understand the movement of object present in the video. So in this section we describe optical flow base Temporal ConvNet and how we deploy it our main proposed system.

From **Figure 3.5** we can see estimation of motion of two frame. The 3^{rd}

(a) Optical Flow Example-1:Two frames with Optical Flow output

(b) Optical Flow Example-2:Two frames with Optical Flow output in different time frame in same video

Figure 3.5: Optical Flow of **CricketShot** extracted from .avi file of **UCF101** dataset

output image only highlights the five moving objects. In our experiment to extract optical flow from video use [25] Which perform state of the art to extract optical flow from video stream data. The main purpose of estimating only moving portion of frame, so that our temporal ConvNet can focus only to the moving portion of the video. Because other portion of video is less significant in terms of understanding action in video stream.

Figure 3.6 shows architecture of Temporal Convolution Neural Network. From

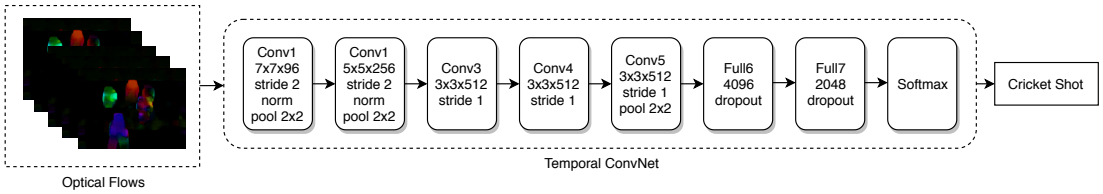

Figure 3.6: Temporal ConvNet Architecture

the figure we can see that optical flow is input to the ConvNet and final output is prediction of action appear in desire video. ConvNet take fix size input to handle this situation we take ten optical flow stack as a fix size and feed this ten stack of optical flow to our Temporal ConvNet. To training Temporal ConvNet we extract all the optical flow of UCF101 dataset and then feed these optical flow to our Temporal ConvNet.

We use total of five convolution layer among them first two layer use stride 2 and max pooling with 2×2 In Conv3, and Conv4 we does not use poling layer and in Conv5 layer we use max pooling with 2×2 and stride is 2. These five convolution layer use for feature extraction from optical flow. After convolution layer we use two dense layer with output neuron 4096 and 2048. After each dense layer we use dropout layer with dropout percentage value 0.3. Finally a softmax layer is used for classification. Softmax layer give a probability distribution over the all class in range of 0 to 1, and maximum probability assign by softmax is

predicted level of the input video.

3.4 Basic Convolution Neural Network (ConvNet)

A Convolutional Neural Network is a Deep Learning algorithm which can take an input image, and able to differentiate one image from the other. Its possible because it assign importance that is weights and biases (Two parameter is learnable) to various perspectives in the image. This why Convolution neural network does not need hand craft feature selection. It can automatically extract

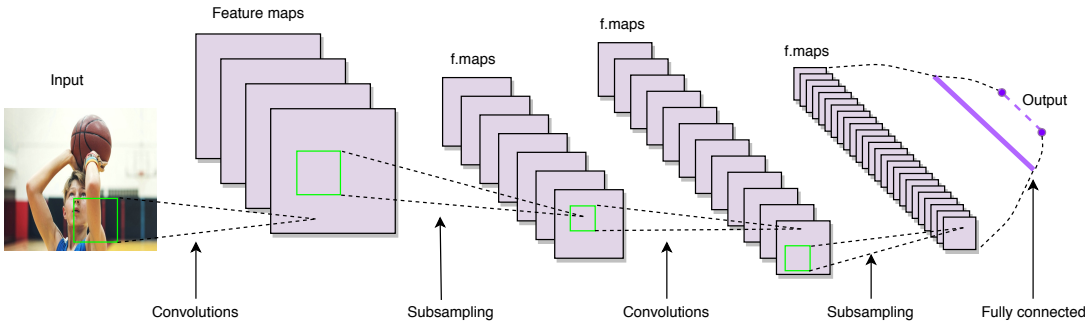

Figure 3.7: A typical ConvNet architecture

features from input images. Consequently therefore a ConvNet required less attention on prepossessing task on image as compared to other machine learning algorithms.The architecture of a ConvNet is similar to Biological Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Like biological neuron artificial neuron activate on specific event. **Figure 3.7** show a general architecture of convolution neural network [**?**]. It use different type of convolution filters to extract spatial features of input images.

A ConvNet is able to successfully capture the Spatial and Temporal information

in an image through the application of relevant filters. The architecture performs a better fitting to the image data set due to the reduction in the number of parameters involved and re-usability of weights. In other words, the network can be trained to understand the image better.

3.5 Spatial ConvNet Used in Our Framework

We use the same network architecture that used in Temporal ConvNet for Spatial ConvNet. The Main difference among these network architecture is the input. In Temporal ConvNet we feed our network stack of Optical Flow but here in Spatial ConvNet we feed single frame to our network.

Figure 3.8 shows the network setup of our Spatial ConvNet. From figure we

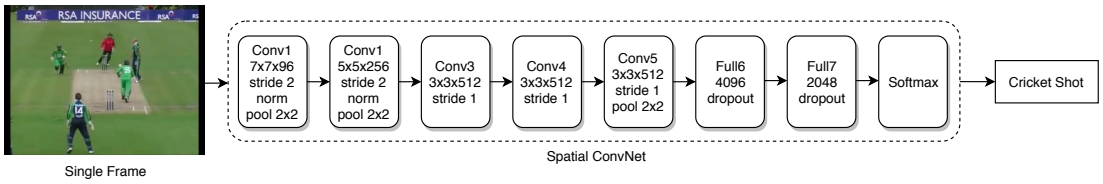

Figure 3.8: Spatial ConvNet Architecture

can see that we feed network single frame. All the network parameter is same as the temporal ConvNet that is, We use total of five convolution layer among them first two layer use stride 2 and max pooling with 2×2 In Conv3, and Conv4 we does not use poling layer and in Conv5 layer we use max pooling with 2×2 and stride is 2. These five convolution layer use for feature extraction from optical flow. After convolution layer we use two dense layer with output neuron 4096 and 2048.

After each dense layer we use dropout layer with dropout percentage value 0.3. Finally a softmax layer is used for classification. Softmax layer give a probability

distribution over the all class in range of 0 to 1, and maximum probability assign by softmax is predicted level of the input video. Apart from this we apply data augmentation technique in the input image for example we apply random zoom, horizontal flip, vertical flip, normalization.

3.6 Fusion

The main concept of Fusion technique is "Many brain is better than one brain". We can consider each of three different network shown in **Figure 3.9** as a single brain. In Fusion method it takes advantage of all the three different network.

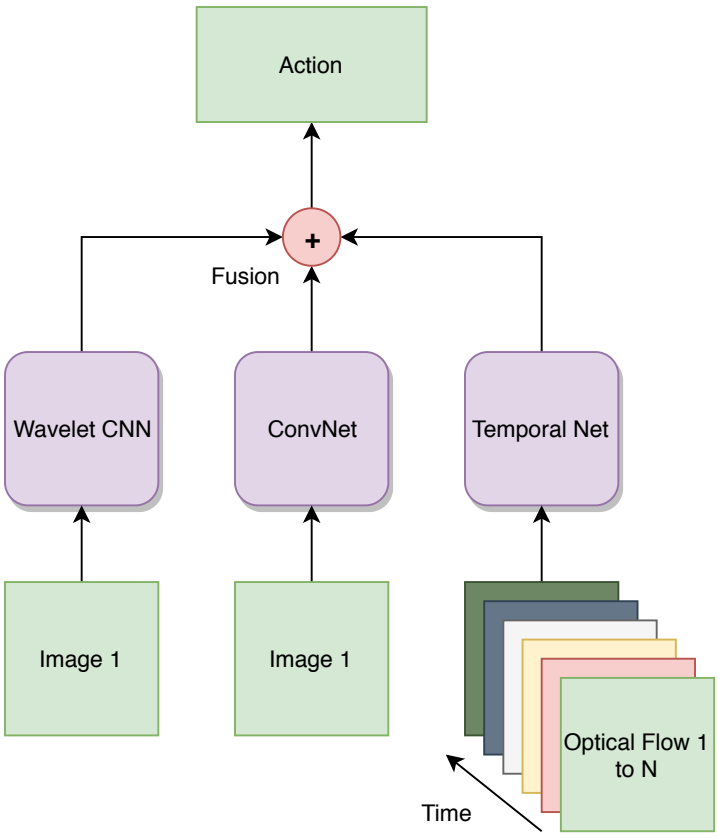

Figure 3.9: Class Score Fusion

Using different network's individual knowledge it will predict the final action.

To implement Fusion with three network we take softmax output from each of network and we average the three network output. One thing keep in mind that the dimension of each network must be save in shape to calculate average. After averaging the final output from three network the average output dimension will be same as individual network output.

Chapter 4

Methodology

Summary

In this section we discuss the methodology used in the study, the stages by which the methodology was implemented, and the research design. We divide this section into three subsections. In first we discuss about dataset, Second section we discuss about instrument used in in this experiment. And the third section cover implementation detail for this experiment to perform.

4.1 Dataset

For our experiment we will use UCF101[27]. It is publicly available dataset for video base action recognition. UCF-101 dataset contain 101 different type of realistic action videos. This video is collected from YouTube. It contains 13320 videos from 101 action categories. The action categories can be divided into five main types:

- Human-Object Interaction
- Body-Motion Only

- Human-Human Interaction
- Playing Musical Instruments
- Sports

The videos in 101 action categories are grouped into 25 groups, where each group can consist of 4-7 videos of an action. The videos from the same group may share some common features, such as similar background, similar viewpoint, etc.

The action categories for UCF101 data set are: Apply Eye Makeup, Apply Lipstick, Archery, Baby Crawling, Balance Beam, Band Marching, Baseball Pitch, Basketball Shooting, Basketball Dunk, Bench Press, Biking, Billiards Shot, Blow Dry Hair, Blowing Candles, Body Weight Squats, Bowling, Boxing Punching Bag, Boxing Speed Bag, Breaststroke, Brushing Teeth, Clean and Jerk, Cliff Diving, Cricket Bowling, Cricket Shot, Cutting In Kitchen, Diving, Drumming, Fencing, Field Hockey Penalty, Floor Gymnastics, Frisbee Catch, Front Crawl, Golf Swing, Haircut, Hammer Throw, Hammering, Handstand Pushups, Handstand Walking, Head Massage, High Jump, Horse Race, Horse Riding, Hula Hoop, Ice Dancing, Javelin Throw, Juggling Balls, Jump Rope, Jumping Jack, Kayaking, Knitting, Long Jump, Lunges, Military Parade, Mixing Batter, Mopping Floor, Nun chucks, Parallel Bars, Pizza Tossing, Playing Guitar, Playing Piano, Playing Tabla, Playing Violin, Playing Cello, Playing Daf, Playing Dhol, Playing Flute, Playing Sitar, Pole Vault, Pommel Horse, Pull Ups, Punch, Push Ups, Rafting, Rock Climbing Indoor, Rope Climbing, Rowing, Salsa Spins, Shaving Beard, Shotput, Skate Boarding, Skiing, Skijet, Sky Diving, Soccer Juggling, Soccer Penalty, Still Rings, Sumo Wrestling, Surfing, Swing, Table Tennis Shot, Tai Chi, Tennis Swing, Throw Discus, Trampoline Jumping, Typing, Uneven Bars, Volleyball Spiking, Walking with a dog, Wall Pushups, Writing On Board, Yo

Yo.

4.2 Instruments

We use following hardware and software setup

- Hardware
  - CPU : AMD Ryzen 7
  - GPU : Nvidia RTX 2080
  - Primary memory: 64 GB
- Software
  - Ubuntu 16.04
  - Python 3.5
  - Tensorflow-gpu 1.8
  - Keras 2.2
  - OpenCV 4.1

4.3 Implementation Details

Spatial ConvNets configuration: The layer configuration of our spatial and temporal ConvNets is schematically shown in 3.8. All hidden weight layers use the rectification (ReLU) activation function. max-pooling is performed over 3 3 spatial windows with stride 2.

Training: For training we split UCF101 into two part. One for train another

is for testing. We randomly separate 20% data for testing and 80% data for training. The network weight is learned using Adam optimizer with batch size 32, each batch data is selected randomly from training data. Original image is resize to 224x224 and data augmentation for example random flip, zoom are apply to the image

Temporal ConvNet configuration: The Layer configuration of our temporal ConvNets is schematically shown in 3.6. It is same as spatial ConvNet configuration. The only difference between spatial and temporal ConvNet configurations is that we removed the second normalisation layer from the latter to reduce memory consumption.

Training: To train the Temporal ConvNet Optical flow is computed using the off-the-shelf GPU implementation from the OpenCV toolbox. Here we use Adam as our optimizer for updating network weights. Other's setup are same as spatial ConvNet.

Wavelet CNN configuration We use 3×3 convolution kernels exclusively and 1×1 padding to ensure the output is the same size as the input. Instead of using pooling layers to reduce the size of the feature maps, we use convolution layers with increased stride. If 1×1 padding is added to the layer with a stride of two, the output becomes half the size of the input layer.

Training: We train our proposed wavelet model with image size 224×224. These images are achieved by first scaling the training image to 256×256 pixels and then conducting random crops to 224×224 pixels and flipping. Random variation helps the model to prevent overfitting. We use Adam optimizer instead of

SGD. Batch normalization technique is use for dealing with overfitting problem.

Chapter 5

Experimental Results

Summary

In this section we discuss details all the results of our study. Also contain a full discussion and evaluation of the results. This section is mainly divided into two main part. In the first part individual ConvNet accuracy. And second part contain result of three stream architecture and it's comparison to others action recognition methods.

5.1 Individual ConvNet and Wavelet CNN Accuracy

Classification result for different levels of multiresolution analysis shown in 5.1. It also show four different training settings and corresponding accuracy.From the table we can see train with from scratch + 2-level decomposition has lower accuracy. But with the increase of decomposition level + Using a pre-train model accuracy starts increase.

We get the best result with pre-train model plus 5-level decomposition. The

Training setting	Accuracy
From scratch + level of decomposition = 2	62.3%
Pre-train + level of decomposition = 3	74.6%
Pre-train + level of decomposition = 4	77.2%
Pre-train + level of decomposition = 5	**77.6%**

Table 5.1: Wavelet ConvNet accuracy on UCF-101

result of spatial convolution neural network model is shown in 5.2. Network training from scratch get poor result. But using a per-train InceptionV3 with fine tuning get the improve accuracy. In this setup use two different dropout ratio.And we get best result with pre-train with only fine-tune last layers of the model with dropout rate 50%. The result of temporal convolution neural

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%

Table 5.2: Spatial ConvNet

network is shown in **5.3**. We first train our mode with single frame optical flow with optical flow staking parameter $L = 1$ and it get better accuracy then normal spatial convolution neural network. After that we increase the optical flow

Input configuration	Mean subtraction	
	Off	On
Single-frame optical flow (L=1)	-	73.9%
Optical flow stacking (L=5)	-	80.0%
Optical flow stacking (L=10)	79.9%	81.0%
Optical flow stacking (L=10), bi-directional	-	81.2%

Table 5.3: Temporal ConvNet. L define number of optical flow to stack

staking value from 5 to 10 and we get 80.0% accuracy with $L = 10$. Addition of

mean subtraction to the input optical flow increase the the accuracy. Means subtraction is a technique where first calculate the mean from original input and this mean value is subtracted from original input and this subtracted output is feed to the network. We get the best result with optical flow stacking value $L = 10$ and **bi-directional** stacking of optical flow. In bi-directional stacking we make the optical stack in reverse order.

5.2 Three stream ConvNet accuracy

In this subsection we evaluate the complete three stream model, that combine three recognition streams. We first tried to join last fully connected layer of each of three steam but it cause over-fitting. So, instead of merging three different

Spatial ConvNet	Temporal ConvNet	Wavelet ConvNet	Fusion Metod	Accuracy
Pre-trained + last layer	bi-directional	Decomposition Level = 4	averaging	85.6%
Pre-trained + last layer	uni-directional	Decomposition Level = 4	averaging	85.9%
Pre-trained + last layer	**uni-directional**	**Decomposition Level = 4**	**SVM**	**92.3%**

Table 5.4: Three-stream ConvNet accuracy on UCF-101

layers we take softmax score from each recognition stream and apply **fusion** technique. We use two different fusion approach one is averaging another is linear SVM. From **5.4** we can conclude that linear SVM fusion method perform better than the averaging fusion method with combination of uni-directional optical flow stacking and a pre-train spatial ConvNet model.

Chapter 6

Conclusions

Understanding various task in the video such as action recognition has a very important application such as human behavior understanding, video translation, understanding suspicious behavior of humans on surveillance camera, this will be a revolutionary change in the camera base security system. We applied our model on UCF101 dataset. However, our method can be applied in any kind of video-based activity recognition, for example, can track the activity of different types of animals in the forest where reaching for human is difficulty.

We present a novel three-stream architecture that incorporates spatial, spectral and temporal analysis into a single model using fusion technique. We demonstrated that combining these features into one model improves the accuracy of action recognition for video data.

We also notice that using an optical flow stack instead of using just a single optical flow improves accuracy.. We also notice that for Wavelet CNNs with the increase of decomposition level it improves the performance.

There is still some space to improve this accuracy. We can increase the num-

ber of data. UCF101 data is not adequate. We can also extract pose from the input image then it can feed to ConvNet instead of direct feeding of image.

Appendix A

Installing Keras and other dependencies on Ubuntu

The process of setting up a deep-learning workstation for this research consist the following steps:

- **Step 1:** Install the Python scientific suite Numpy and SciPy and make sure you have a Basic Linear Algebra Subprogram (BLAS) library installed so your models run fast on CPU.
- **Step 2:** Install two extras packages that come in handy when using Keras: HDF5 (for saving large neural-network files) and Graphviz (for visualizing neural-network architectures).
- **Step 3:** Make sure your GPU can run deep-learning code, by installing CUDA drivers and cuDNN.
- **Step 4:** Install a backend for Keras: TensorFlow, CNTK , or Theano. We use **TensorFlow** as backed
- **Step 5:** Install Keras
- **Step 6:** Install OpenCV

Step 1:

To install scientific suite for example Numpy, SciPy, BLAS we need to install **Anaconda** it contain all the popular scientific suite in one package. Go to `https://www.anaconda.com/` and download anaconda 64 bit version. You can install it by a bash command:

$bash install Anaconda*.sh

Step 2:

Install Graphviz and pydot-ng, two packages that will let you visualize Keras models. They arent necessary to run Keras, so you could skip this step and install these packages when you need them. Here are the commands:

$ sudo apt-get install graphviz

$ sudo pip install pydot-ng

Step 3:

Download CUDA For Ubuntu (and other Linux flavors), NVIDIA provides a ready-to-use package that you can download from `https://developer.nvidia.com/cuda-downloads`

Install CUDA. The easiest way to do so is to use Ubuntus apt on this package. This will allow you to easily install updates via apt as they become available:

$ sudo dpkg -i cuda-repo-ubuntu1604_9.0.176-1_amd64.deb

$ sudo apt-key adv –fetch-keys

`http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/7fa2af80.pub`

$ sudo apt-get update

$ sudo apt-get install cuda-9.1

Install cuDNN: a Register for a free NVIDIA developer account (unfortunately, this is necessary in order to gain access to the cu DNN download), and download cu DNN at `https://developer.NVIDIA.com/cudnn` (select the version of cu DNN compati-ble with TensorFlow). Like CUDA , NVIDIA provides packages for different Linux flavorswell use the version for Ubuntu 16.04. Install cuDNN by following command:
$ sudo dpkg -i dpkg -i libcudnn6*.deb

Step 4:
TensorFlow with or without GPU support can be installed from PyPI using Pip. Heres the command with GPU support:
$sudo pip install tensorflow-gpu

Step 5:
You can install Keras from PyPI: $ sudo pip install keras
Alternatively, you can install Keras from GitHub. Doing so will allow you to access the keras/examples folder, which contains many example scripts for you to learn from:
$ git clone https://github.com/fchollet/keras $ cd keras $ sudo python setup.py install

Step 6: you can install OpenCV from PyPI using following command:
$pip install opencv-python

References

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014. 1

[2] I. S. A. Krizhevsky and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," 2012. 1, 4

[3] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," 2014. 1, 17

[4] A. Haar, "Zur theorie der orthogonalen funktionensysteme. mathematische annalen," 1910. 2

[5] G. C. P. Doll ar, V. Rabaud and S. Belongie, "Behavior recognition via sparse spatio-temporal features," 2005. 16

[6] I. Laptev., "On space-time interest points.," 2005. 16

[7] A. K. a. I. L. H. Wang, M. M. Ullah and C. Schmid., "Evaluation of local spatio-temporal features for action recognitio," 2009. 16

[8] Y. L. G. W. Taylor, R. Fergus and C. Bregler., "Convolutional learning of spatio-temporal features.," 201. 16, 17

[9] M. Y. S. Ji, W. Xu and K. Yu., "3d convolutional neural networks for human action recognition.," 2013. 16, 17

[10] S. S. T. L. R. S. a. L. F.-F. A. Karpathy, G. Toderici, "Large-scale video classification with convo-lutional neural networks," 2014. 16, 17

[11] K. Simonyan and A. Zisserman., "Two-stream convolutional networks for action recognition in videos," 2014. 16

[12] T. T. G. Willems and L. J. V. Gool., "An efficient dense and scale-invariant spatio-temporal interest point detector.," 2008. 16, 17

[13] C. S. I. Laptev, M. Marszalek and B. Rozenfeld, "Learning realistic human actions from movies," 2008. 16

[14] M. M. A. Kl aser and C. Schmid., "A spatio-temporal descriptor based on 3d-gradients.," 2008. 16

[15] A. K. I. L. H. Wang, M. M. Ullah and C. Schmid., "Evaluation of local spatio-temporal features for action recognition.," p. 11, 2009. 16

[16] X. Y. W. Z. J. Zhu, B. Wang and Z. Tu., "Action recognition with actons.," p. 11, 2013. 16

[17] S. Sadanand and J. J. Corso., "Action bank: A high-level representation of activity in video.," 2012. 16

[18] Y. Q. L. Wang and X. Tang., "Mid-level 3d parts for human motion recognition.," 2013. 16

[19] Y. Q. L. Wang and X. Tang., "Mining motion atoms and phrases for complex action recognition.," 2013. 16

[20] Y. Q. L. Wang and X. Tang., "Video action detection with relational dynamic-poselets.," 2014. 16

[21] K. Simonyan and A. Zisserman., "Very deep convolutional networks for large-scale image recognition.," 2014. 17

[22] Y. J. P. S. S. R. D. A. D. E. V. V. C. Szegedy, W. Liu and A. Rabinovich, "Going deeper with convolutions," 2014. 17

[23] M. D. Zeiler and R. Fergus., "Visualizing and understanding convolutional networks.," 2016. 17

[24] M. Y. S. Ji, W. Xu and K. Yu., "3d convolutional neural networks for human action recognition," 2017. 17

[25] T. S. M. K. A. D. T. B. Eddy Ilg, Nikolaus Mayer, "Flownet 2.0: Evolution of optical flow estimation with deep networks," p. 16, 12 2016. 19, 24

[26] "https://en.wikipedia.org/wiki/Multiresolution$_a nalysis$,″ 21

[27] "Ucf101: A dataset of 101 human action classes from videos in the wild.," 11 2012. 29

[28] "http://www.image-net.org/,"

[29] HMDB51.Dataset, "`http://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/`,"

[30] N. P. T. Brox, A. Bruhn and J. Weickert., "High accuracy optical flow estimation based on a theory for warping," 2004.

[31] A. Haar., "Zur theorie der orthogonalen funktionensysteme.mathematische annalen," 1910.

[32] JPEG.Compression, "`https://en.wikipedia.org/wiki/JPEG_2000`," 2000.

[33] K. T. S. Fujieda and T. Hachisuka., "Wavelet convolutional neural networks for texture classification," 2017.