Enhanced Spatial Stream of Two Stream Network for Human Action Recognition

Author

SHAHBAZ KHAN

274099

Supervisor

DR. ALI HASSAN

DEPARTMENT OF COMPUTER & SOFTWARE ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

DECEMBER, 2021

Enhanced Spatil Stream of Two Stream Network for Human Action Recognition

Author

SHAHBAZ KHAN

274099

A thesis submitted in partial fulfillment of the requirements for the degree of

MS Computer Engineering

Thesis Supervisor:

DR. ALI HASSAN

Thesis Supervisor's Signature:____

DEPARTMENT OF COMPUTER & SOFTWARE ENGINEERING
COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING
NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY,
ISLAMABAD

DECEMBER, 2021

Declaration

I certify that this research work titled *"Enhanced Spatial Stream of Two-Stream Network for Human Action Recognition"* is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

Signature of Student
SHAHBAZ KHAN
2018-NUST-Ms-Comp-000274099

Language Correctness Certificate

This thesis has been read by an English expert and is free of typing, syntax, semantic, grammatical and spelling mistakes. Thesis is also according to the format given by the university.

Signature of Student
SHAHBAZ KHAN
2018-NUST-Ms-Comp-000274099

Signature of Supervisor
DR. ALI HASSAN

Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Signature of Student
SHAHBAZ KHAN
2018-NUST-Ms-Comp-000274099

Signature of Supervisor
DR. ALI HASSAN

Copyright Statement

- Copyright in text of this thesis rests with the student author. Copies (by any process) either in full, or of extracts, may be made online accordance with instructions given by the author and lodged in the Library of NUST College of E&ME. Details may be obtained by the Librarian. This page must form part of any such copies made. Further copies (by any process) may not be made without the permission (in writing) of the author.
- The ownership of any intellectual property rights which may be described in this thesis is vested in NUST College of E&ME, subject to any prior agreement to the contrary, and may not be made available for use by third parties without the written permission of the College of E&ME, which will prescribe the terms and conditions of any such agreement.
- Further information on the conditions under which disclosures and exploitation may take place is available from the Library of NUST College of E&ME, Rawalpindi.

Acknowledgements

I am grateful to Allah, my Creator, for guiding me through every stage of this project and for every new notion You implanted in my head to help me enhance it. Without Your invaluable assistance and instruction, I would have been unable to do anything. It was Your will that whomever assisted me over the duration of my thesis, whether it was my parents or anybody else, thus no one else is deserving of appreciation but You.

I am eternally grateful to my loving parents, who reared me from the time I was unable to walk and have continued to support me in every aspect of my life.

I'd also want to thank my supervisor, DR. ALI HASSAN, for his guidance during my thesis and for the Machine Learning course that he taught me. I can confidently state that I have never mastered an engineering topic as thoroughly as the one he has taught.

I'd like to express my gratitude to my co-supervisor, DR. FARHAN HUSSAIN, for his unwavering support and collaboration. He always had an answer for me when I was stuck on something. I would not have been able to finish my thesis without his help. Throughout the thesis, I appreciate his patience and help.

I’d also want to express my gratitude to DR. FARHAN RIAZ, DR. UMER FAROOQ, and A.P. JAHAN ZEB for serving on my thesis guidance and assessment committee, as well as MR. ASAD for his help.

Finally, I'd want to extend my thanks to all of the folks who have helped me with my research.

Dedicated to my parents whose unconditional love and support made me accomplish this achievement

Abstract

CNN have been proven effective in deep learning methods for Huaman Action Recognition (HAR) along with other computer vision tasks but the problem of overfitting in this domain remains till date, as deep learning models need large amount of data for training. This thesis is inspired by the two-stream network for HAR where CNN has been deployed as a base model to show that both, the spatial and the temporal aspects of an action are important for its recognition.

To deal with the mentioned issue we have proposed enhancement of the spatial stream, which consists of two parts. Primarily, we adopted transfer learning in the spatial stream, where we demonstrated that by using models which are pre-trained on larger datasets like ImageNet yields good performance instead of training the original model from scratch. Secondly, we offer dataset augmentation technique, where we increased the dataset size by performing various random transformations like rotations, cropping and flipping on the image. Further, fine-tuning the network of the enhanced spatial stream on the augmented dataset increases the accuracy.

Our architecture is trained and tested on UCF-101 dataset, which is the latest and standard benchmark for action videos. Our results are competent and are comparable with the state of the art two-strean network's results. Also, our network performed well in the spatial stream as compared to other models.

Key Words: *Human Action Recognition, Overfitting, Transfer Learning, Two Stream Network*

Table of Contents

List of Figure

List of Tables

CHAPTER 1: INTRODUCTION

1.1 Introduction to Human Action Recognition

Human Action Recognition (video-based) is the most significant study fields in computer vision, with algorithms becoming more effective by the day. Our job in Human Action Recognition is to automatically detect the activity type that is being done in the video. It is a difficult work owing to the many problems it entails, such as camera motion, varying lighting conditions, backdrop bombardment, various human forms, occlusion, perspective fluctuation, and so on. However, the impact of these problems varies depending on the activity. Gestures, Actions, Group Activities and Interactions are the four primary kinds of actions performed by humans.

Gestures: They are simple movements of human body components such as nodding and hand gestures. Gestures lasts for just a few seconds, and the intricacy required is minimal since the activity revolves around a single region of the body.

Action: It generally includes the majority of the body, although it may also be a mixture of many motions. A single person's action might take place for a short or comparatively lengthy time. Walking, punching and sitting are among examples.

Interactions: It may involve Human-to-human or human-to-object interaction. It is more complicated than simple actions since we now have two separate topics in a single frame. Instances of human-to-human interactions include handshakes, fighting and hugging, whereas examples of human-to-object interactions include a person using a smartphone, using an ATM and washing a vehicle.

Group Activities: This is the most difficult activity to identify since it consists of a variety of behaviors, interactions, and gestures. It must include two or more individuals, as well as a single or many objects. A few instances are cricket, a protest by a group of individuals and a group gathering. The dataset used in this thesis includes activities from 54ygestures, movements, and interactions only.

1.2 Motivation

One of the main objectives of artificial intelligence is to develop a model that can correctly comprehend human behaviour and intents [3], motion representation is a basic job for extracting motion information from numerous frames. Several strategies have been used to capture the motions from the images. While current research is mostly focused on temporal models, complex temporal structures are created by using sparse segments to emulate long-term temporal structure in Temporal Segment Networks (TSN)[4]. The concept that models learn hierarchical motion patterns in image space has been used to tackle various temporal modelling challenges in 3D CNN networks [5-7]. Tracking characteristics are employed to improve the effectiveness of temporal modelling in [8, 9].

In [6, 10, 11] authors emphasis on using Convolutional Networks for this job. Researchers have shown that temporal filters, such as local spatiotemporal filters can be applied to spatiotemporal objects such as actions, which then make it possible to use spatial recognition ideas on temporal objects [6, 10, 12]. This difference between time and space is significant, and numerous different techniques to these dimensions have been examined, such as adding optical flow networks (which simulate motion) [1] or modelling time sequences in recurrent structures (which represent patterns in nature) [13-15].

In traditional 3D CNN approaches, the models are often fed RGB stacked pictures as input. However, in [11, 16, 17], such input is insufficient since a stack of RGB photos includes a lot of useless information and may pay more attention to aspects such as backdrop and appearance elements rather than the motions themselves. As a result, these RGB frames are mixed with optical streams to account for movement in order to improve the overall performance of the two stream models. Two-stream network is based on a hypothesis that came out of neuroscience research called the two-stream hypothesis [18] , which says that the brain’s visual cortex contains two separate streams, one which processes information about the visual attributes of objects like shape and color, known as the ventral pathway, and other which responds to transformations in the object, and to spatial relationships as an object of motion known as the dorsal pathway.

All the mentioned approaches shows that convolutional neural network performs well for human action recognition. Most of the research on two stream network focuses on its temporal stream. In this paper we show that by using pre-trained models in the spatial stream yields good performance results as compared to training the entire model from scratch and it also saves time. We did that by keeping the classification layers of the original two-stream model [1] fixed and attaching the feature extraction layers of different pre trained model one by one and then training the fully connected layers only to check which model performs best. After selecting the best model, we fine-tune the whole network to see if we could improve results.we propose strategy to deal with the problem of overfitting. The main reason behind overfitting is limited dataset provided to a deep network to train its model. We incorporated dataset augmentation to increase our dataset size by using different augmentation techniques like image flipping and image rotation.

1.3 Problem Statement

According to the methodologies presented, CNNs are more efficient since it can learn spatial and temporal properties. However, in all of the approaches outlined, overfitting is a major issue that arises when using deep convolution networks on a limited dataset like UCF-101 which is used by most of the researchers for this field of research, since it is the latest dataset available.

1.4 Objectives

This research is carried out to attain following objectives:

- To validated and compare the performance of the purposed method with published methods.

- To assess the performance of the proposed methodology using the real datasets.
- To create a machine learning model for the recognition of human actions using video data.
- To assess the performance of the different feature extraction tools for Human Action Recognition

1.5 Structure of Thesis

Following is the structure of this thesis.

- Chapter 2 presents the previous work that has been done on human action recognition by using two stream network and other approaches
- Chapter 3 discusses selection of dataset and its features which are used in our research work.
- Chapter 4 describes the methodology that is used to generate the model.
- Chapter 5 discusses the experimental results in detail including coveted tables and figures.
- Chapter 6 informs about the conclusion and discloses the gaps of this research.

CHAPTER 2: LITERATURE REVIEW

The Human Action Recognition (HAR) algorithm is used to identify videos containing actions. Once video data has been collected either by video or by wearable or portable sensors, it is then processed beforehand to meet the needs of the desired application.. Figure 2.1 depicts a generic HAR system; It gives an overview of the process, which includes steps like dataset collection, pre-processing, feature extraction, encoding, as well as the use of machine learning methods for action classification and feature dimensionality reduction. Section 2.2 details prior work on techniques based on deep learning,

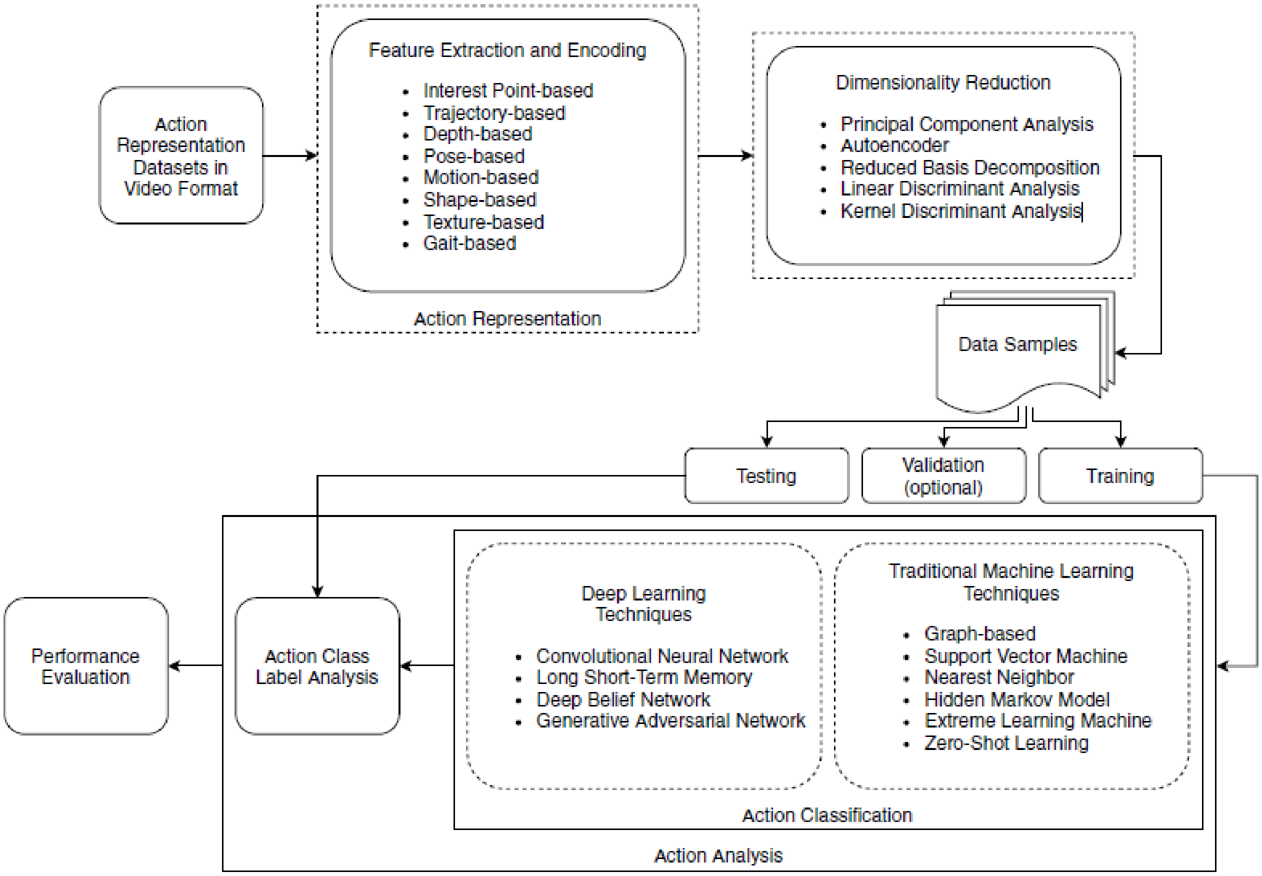


Figure 2.1. An overview of generic Human Action Recognition system.[2]

2.1 Previous work on Deep Learning methods

Deep Learning (DL) is a process that teaches computers to do tasks akin to those completed by the human brain. For this research, we looked at CNN, RNN, LSTM, DBN, and GAN, which are all common action recognition networks.

The maps that CNN generates are based on the neighborhood information. Convolution, activation, and pooling are included in CNN architecture for feature extraction (avg., min., or max.). [19] presents a novel way to gather temporal and spatial data for video recognition called a 3D convolution. To accomplish a 3D convolution approach, a 3D kernel may be convoluted in stacked multiple frames. Is should be noted that 3D convolution technique is costly, and the results are unstable. Without a GPU to help speed things up, training time will be longer [19]. Features are extracted in different steps using CNN's architecture. Convolution, nonlinear neuron activation, and feature pooling are essential to all three steps. The design shown in Figure 2.2 demonstrates the fundamental concept of a deep CNN. CNNs are considered to be deep if too many layers of feature extraction are coupled together [1]. In [20] spatial and temporal domains are used to perform convolution in CNN.

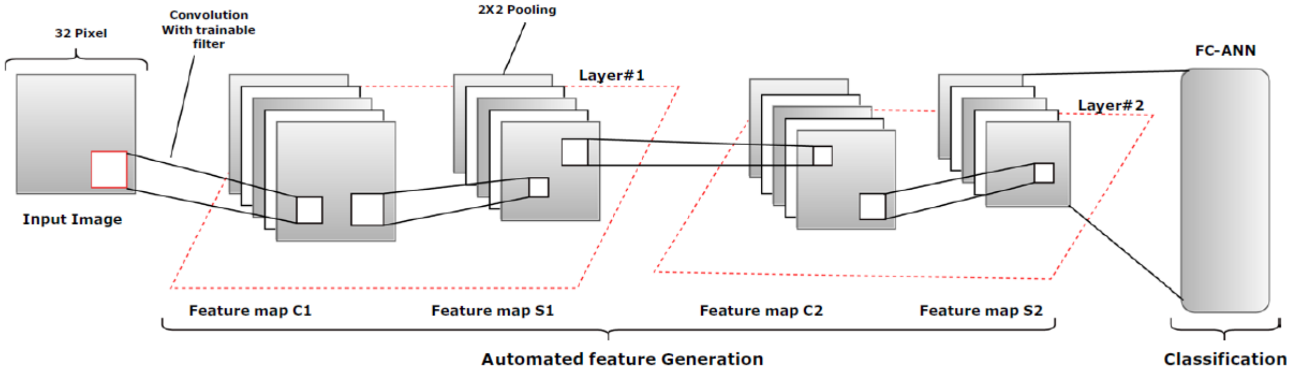


Figure 2.2. A typical Deep CNN Architecture [1].

Space–time volumes are used as input in 3D CNN. In addition, LSTM training is done by employing the 3D CNN features that have been collected. 3D CNN could derive spatial-temporal information from the input video. Because 3D CNN models have layered-on top of one other, training and memory use have both grown [21]. The method shown in [21] entails the use of a 3D feature map in conjunction with a 2D convolutional block to be coupled together serially. The model's computing cost is decreased by adding cross-domain residual techniques to the temporal dimension. Residual connection is advantageous since it extracts static 2D characteristics, as opposed to learning about static 3D properties.

The posture-based features of the 3D CNN are recovered from it, 3D pose combines the 2D appearance and motion stream [22]. Using the color joint features extraction from the 3D CNN results in an extensive process; thus, a fifteen-channel heatmap gets created, and convolution is performed on each of the maps. The pairwise distance between skeleton features in the case of skeleton-based HAR, is calculated according to [23]. The four networks get their CNN input from Joint Distance Maps (JDM) which is trained and fused (later) afterwards. Another way to say that skeleton-based input is handled differently is to say that it is analysed differently using multi-stream CNNs [24] which implement modified AlexNet [25] and colour input data is sent to each CNN. Each CNN generates class scores probabilities which is fused of all of the potential class scores.   The study exhibits the capacity of CNN to face changes in the image (when compared to other CNNs), differing levels of noise in the data, and different levels of skeletal similarity in the data. In addition, it has been shown that the suggested network is much better than LSTM techniques.

Deeper CNN, specifically ConvNetworks, are used to perform robust HAR with gyroscope and accelerometer on a smartphone [26], where a local dependence of timeseries 1D is used to produce signals, and features are autonomously retrieved using CNN (instead of having to utilize pre-processing methods). The dense layer is coupled with softmax to turn the output of CNN into a probability distribution. [14] proposes a two-stream convolution network for combining both temporal and spatial streams, in which optical flow (motion) and RGB information (spatial) are modelled individually and results are averaged in the final layer. Due to optical flow, the network is unable to grasp long-term motion; another disadvantage of the appearance stream is that performance is predicated on random single frame picked from input video. As a result of backdrop clutter and perspective fluctuation, difficulties exist [14].

Temporal Segment Network (TSN) is proposed in [8], which contains significantly similar frames after sampling, which removes the requirement for dense temporal sampling. TSN uses sparse sampling from long input films. Inception and Batch Normalization is used in [8]. Two-stream networks combine optical flow and RGB fields together with RGB and optical flow pictures to mimic change in appearance (to suppress background motion).

In [27], another two-stream network is described to enhance skeleton joint-based HAR performance. AGCN applies the adaptive graph convolutional network to provide joint and bone information, as seen in this video: These key components form the network. The softmax layer is used on the output. [28] suggests a CNN structure made up of several convolutions in an action graph and temporal convolutions, all layered on top of each other. The graph structure is learnt from data in order to connect the many junctions. [29] has a non-stationary camera and real-time footage from the system is recorded to disc. CNN is used to automatically extract frame-level characteristics using deep learning. Data from a video stream is used to pre-train a model, as shown in Figure 2.3 [29].

Changes in human behavior are learnt in low dimensions by using CNN (or deep autoencoder) to relate the observed changes to deep temporal models. In the case of categorizing human activities, the SVM (quadratic) classifier is often used. In [30] graphic displays the Pose Feature to Image (PoF2I) encoding method, which employs orientation and distance to represent skeletal data as an image. Fine-tuning these photographs with an inception-v3 deep ConvNet ensures a minimal amount of overfitting.

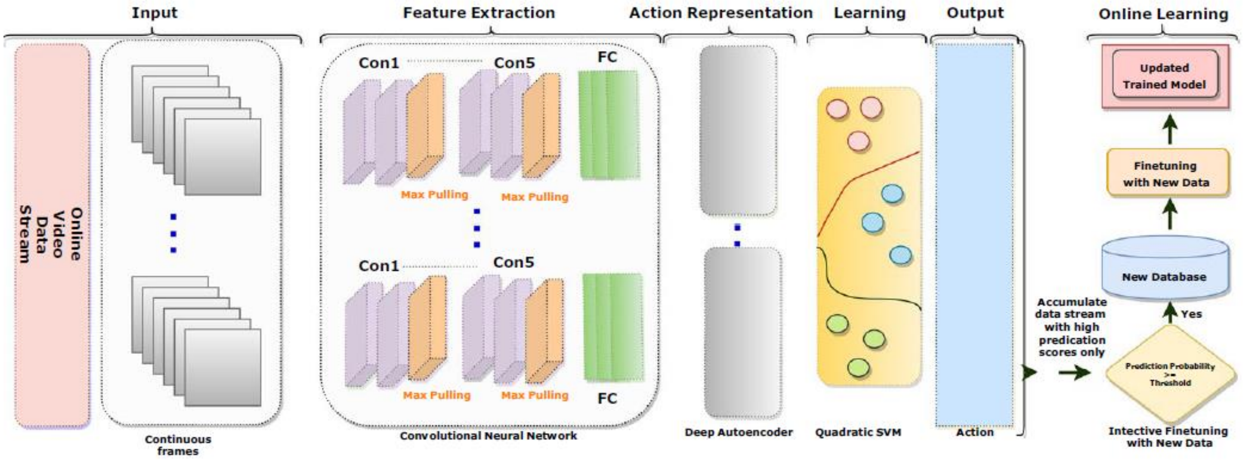


Figure 2.3. A typical pretrained CNN Model for HAR [2]

The deep learning may be found dominantly in action recognition, such as Convolutional Neural Networks, although shallow techniques, such as ML-based techniques, should be studied first. Whereas deep neural networks are well-suited for big data applications, shallow techniques perform well on small datasets. It is possible to use transfer learning in the presence of features that are common across both the base and target datasets. DL models may also be fine-tuned, increasing their performance. In [31] to aid daily activity recognition: spatial layout and temporal encoding is used. LSTM (Long-term state reconstruction Network) is used to track dependencies over longer periods of time utilizing 3-layered stacked LSTM. To gather pose-based static characteristics, CNNs are rather often used. An individual frame depicts the anatomy of a bodily area using the upper body, left right hand and the full body. Pre-trained Resnet152 is used for deeper feature extraction. Once the features are extracted and have been trained into a Support

Vector Machine (SVM), which produces results based on a cross-validation set, SVM uses the learnt features to do a classification.

In [32], it is shows that when individuals are coping with an action recognition challenge, they are unable to focus on an entire scenario at once. Regardless, useful information may be discovered by scanning the photo in many locations. To help with tasks that need more attention, attention models may be used to locate the focal point of the model, which increases to Interpretability [14]. The training input videos are performed utilizing Google Net and the features are extracted from final convolutional layer. A three-layered LSTM is used for classes label prediction. It is essential for the model to study each section of the frame, and it applies a cross entropy loss function with attention regularization. HAR may use an attention mechanism to target a certain body part. [33] presents a technique for end-to-end action detection that makes use of a 3D skeleton and spatial attention that have been pre-trained using a 3D CNN-based I3D model. A three-layer stacked LSTM is used to obtain temporal information in this example. Attention-based processing is used for human action recognition, which focuses on the essential aspects of the action.

When analyzing RGB data, CNN-based network appears to do well at finding spatial relationships. Although LSTM networks may be used to extract temporal correlations from videos, they are also useful for other applications. Although LSTM and CNN perform complementary networks, by using the results of merging later LSTM and CNN score fusion, the model's performance may be greatly improved. Additionally, CNN will need a great amount of data in order to avoid overfitting. Training dropout or data augmentation approaches might be employed to combat overfitting.

Discussions are still open for DL method for HAR due to advances in computer processing power, such as GPU. There are various HAR solutions like DL models which focuses on motion features learning and utilizing it for classification. In Table 2.1, various advantages and downsides of DL-based methods for action recognition are listed.

Table 2.1. Pros and Cons of DL-based techniques for action classification.

Classifier	Advantage	Disadvantage
DBN	Efficient in unsupervised learning[34]	Computationally expensive [34]
CNN	By deploying different filters and pooling layers it can extract temporal features efficiently [35]	Usually requires large datasets. Otherwise results overfitting [36]
RNN	Good for temporal variations based modeling of data [36]	Vanishing gradient occurs usually [36]
LSTM	In the temporal domain, LSTMs may be used to represent long-term contextual information. [37]	Spatial information is difficult to extract [37]
GAN	Good for semi-supervised [38]	Training the model is difficult [38]

2.2 Previous work on Two-Stream Models

The two-stream ConvNetworks [11] (Convolutional Neural Networks) successfully performed in the process of human action detection. Two-stream ConvNetworks, which have a spatial and a temporal stream, begin as separate entities and subsequently merge their recognition streams together. Spatial network uses video stills and the temporal network utilizes stacked optical flow motion information for recognition of action classes. The problem of disappearing or exploding gradients may make training deeper structures using two-stream ConvNetworks problematic [39]. Vanishing Gradients in neural networks with numerous layers has been documented by other researchers as well. Because gradient information is back-propagated, using the same weights again and again or repeatedly applying convolution or multiplication cannot accurately reflect the gradient. This is especially true in the early layers. A variety of other solutions, such as careful initialization [39, 40] and Batch Normalization [41], were tried, although these approaches were only partially successful in mitigating such an impact. [11] used three-fold to improve HAR accuracy. At the outset, they put out a concept of two-stream ConvNet that mixes spatial and temporal networks. Secondly, they show that despite little training data, a ConvNet can be trained on dense optical flow from several consecutive frames, and yet achieve outstanding results. It was concluded by showing how multitasking learning may be deployed to significantly increase the size of training data while enhancing overall performance. Although designed network still needs to catch up with current state-of-the-art shallow representation [42]. The most prominent feature is local trajectory pooling, with spatial and temporal tubes that are coordinated across spatiotemporal layers to concentrate on trajectories. Even while the network can detect the optical flow along the trajectories, it ignores trajectories in spatial pooling. Also by using mean displacement subtraction, camera motion can be compromised.

In [16], spatiotemporal ResNetworks is used as a synthesis of these two methods. To begin, residual connections between the appearance and motion channels of a two-stream architecture are injected to allow for spatiotemporal stream interaction. Then, learnt convolutional filters are applied to adjacent feature maps in time to convert pretrained image ConvNetworks to spatiotemporal networks. [17] claims that spatial and temporal networks should be merged together at a convolutional layer, with no loss in performance but large reductions in parameters;

A spatiotemporal architecture has been designed for two-stream networks comprising a novel temporal fusion layer and novel convolutional fusion layer, which are connected to the networks (incorporating 3D convolutions and pooling). With regard to performance, the innovative design exceeds the top of the rank on two common benchmark datasets without significantly increasing the number of parameters. According to the findings, it was found that learning correspondences between ConvNet characteristics that are very abstract in both space and time is extremely important. One interesting discovery found is that FV encoded IDT features performed better when included with ConvNet predictions. This is a time of when more investigation is needed in the future. They concluded by exploration of used datasets by addressing the point that either they are either small or too noisy.

A deep fusion architecture is built by [43] which uses temporal features from LSTM and spatial features from CNNs in a more effective manner. In addition, it provides a detailed analysis of their strengths and weaknesses. It was also observed that fully connected features are used to steer the LSTM to portions of the convolutional feature sequence that are of interest. Because it is simpler and more effective than competing technologies, the fusion method is also vital. Multi-stream hierarchical fusion strategy has shown to be superior to single-stream mapping techniques in UCF11, UCFSports, and jHMDB, exhibiting good accuracy and outperforming current top of the rank techniques for all datasets. In [44] remodeling of dataset is deployed for initializing model learning by using the augmentation of data and ResNet101 layers parameters trained on the dataset like ImageNet is used to deal with the overfitting issues caused by lesser data. Deeper

ConvNet have been developed for learning complexity of action. Using a disorder testing and training method, the model and procedure may provide a substantial boost in action recognition. The experiment proved that the strategy beat current top of the rank methodologies on two advance datasets, the UCF101 and KTH. Temporal network with deeper Convolutional Networkss do not perform as well as the appearance networks on the UCF101 during the experimental evaluation. The following potential alternative might help to overcome this constraint where it proposes to capture information on motion with a deep temporal structure by adopting deeper RNNs.

Carreira and Zisserman created the Kinetics dataset [5] to address the overfitting issue since it is big enough to train without overfitting. Hara et al. [45] used residual frames to four distinct datasets and had outstanding results. However, the major purpose of [45] was to see whether it could handle the large number of parameters of 3D CNN.

Two-stream Adaptive Graph Convolutional Network (2s-AGCN) was designed specifically for action recognition in [27] which uses skeleton technique. It is possible that the BP technique will learn the network architecture either uniformly or individually as it goes along. By making this data-driven technique part of the model, it raises the model's flexibility for constructing graphs and increases the model's generality for varying data samples. To explain both first-order and second-order information, a two-stream framework is developed, and as a time, a large improvement in recognition accuracy is achieved. In [46] residual images are used to feed to the temporal stream of the network rather than conventional optical flow images. This reduces the computation power and also increased the accuracy as compared to many states of the art models. Because residual frames offer minimal information on object appearance, they utilized a 2D convolutional network to extract appearance features and combine them with residual frame findings to build a two-path solution. Table 2.2 summarizes the previous work done on two stream networks discussed in this chapter.

Table 2.2. Summary of previous work on Two Stream Networks.

Ref.	Contribution	Modality	Features	Model	Dataset	Results
[9]	Two-Stream, Dense Optical Flow Input, Multitask Learning	RGB, Optical Flow	Deep features	Two-Stream ConvNetworks	UCF-101, HMDB51	88%, 59.4%
[15]	Introducing ResNetworks instead of conventional ConvNetworks	RGB, Optical Flow	Deep features	Two-Stream ResNetworks	UCF-101, HMDB51	93.46%, 66.41%
[16]	Fusing the two streams at different layers rather than fusing at fully connected layers	RGB, Optical Flow	Deep features	Two-Stream ConvNetworks	UCF-101, HMDB51	93.5%, 69.2%
[43]	Deep Fusion Framework	RGB	Deep features	Two-Stream LSTM	UCF11, UCF Sports	94.6%, 99.1%, 69.0%
[44]	Transfer Learning, Augmented Data Variation	RGB, Optical Flow	Deep features	Two-Stream ConvNetworks	UCF101, KTH	95.1%, 90.2%
[27]	Adaptive Graph ConvNet, formulation of Skeleton Data	Skeleton Graphs	Deep features	Two-Stream Adaptive Graph ConvNetworks	NTU-RGBD and	95.1%, 58.7%
[45]	Residual Images	RGB, Residual Images	Deep features	Two-Stream ConvNetworks	UCF101, HMDB51	98.6%, 86.6%

CHAPTER 3: DATASETS

3.1 UCF-101

UCF101 is a data collection of realistic action videos taken from YouTube with 101 action categories. This dataset is a supplement to the UCF50 data collection, which has 50 activity categories.

UCF101 has the biggest variety in terms of actions, with 13320 films from 101 action categories, and it is the most complex dataset to date in this domain, with substantial differences in camera motion, object look and position, object size, perspective, cluttered backdrop, light conditions, and so on. Because the majority of accessible action recognition datasets are unrealistic and produced by actors, UCF101 intends to inspire future action recognition research by learning and exploring new realistic action categories. The videos in the 101 activity categories are divided into 25 groups, each of which may have 4-7 movies of an activity. Videos from the same group may have certain qualities in common, such as a similar backdrop, a similar perspective, and so on. The activity categories are classified into five types: 1. Body Motion Only, 2. Human-Object Interaction, 3. Musical Instrument Playing, 4. Human-Human Interaction and 5. Sports. Following are the categories for UCF101 dataset are:

ApplyLipstick, Apply_Eye_Makeup, Archery, BabyCrawling, BalanceBeam, BandMarching, BaseballPitch, BasketballShooting, BasketballDunk, BenchPress, Biking, BilliardsShot, BlowDryHair, BlowingCandles, BodyWeightSquats, Bowling, BoxingPunchingBag, BoxingSpeedBag, Breaststroke, BrushingTeeth, CleanandJerk, CliffDiving, CricketBowling, CricketShot, CuttingInKitchen, Diving, Drumming, Fencing, FieldHockeyPenalty, FloorGymnastics, FrisbeeCatch, FrontCrawl, GolfSwing, Haircut, HammerThrow, Hammering, HandstandPushups, HandstandWalking, HeadMassage, HighJump, HorseRace, HorseRiding, HulaHoop, IceDancing, JavelinThrow, JugglingBalls, JumpRope, JumpingJack, Kayaking, *Knitting, LongJump, Lunges, MilitaryParade, MixingBatter, MoppingFloor, Nunchucks, ParallelBars, PizzaTossing, PlayingGuitar, PlayingPiano, PlayingTabla, PlayingViolin, PlayingCello, PlayingDaf, PlayingDhol, PlayingFlute, PlayingSitar, PoleVault, PommelHorse, PullUps, Punch, PushUps, Rafting, RockClimbingIndoor, RopeClimbing, Rowing, SalsaSpins, ShavingBeard, Shotput, SkateBoarding, Skiing, Skijet, SkyDiving, SoccerJuggling, SoccerPenalty, StillRings, SumoWrestling, Surfing, Swing, TableTennisShot, TaiChi, TennisSwing, ThrowDiscus, TrampolineJumping, Typing, UnevenBars, VolleyballSpiking, Walkingwithadog, WallPushups, WritingOnBoard,YoYo.* [47]

Table 3.1. Summary of Characteristics of UCF101.

Actions	101
Clips	13320
Groups per Action	25
Clips per Group	4-7
Total Duration	1600 mins
Min Clip Length	1.06 sec
Resolution	320*240
Max Clip Length	71.04 sec
Frame Rate	25fps

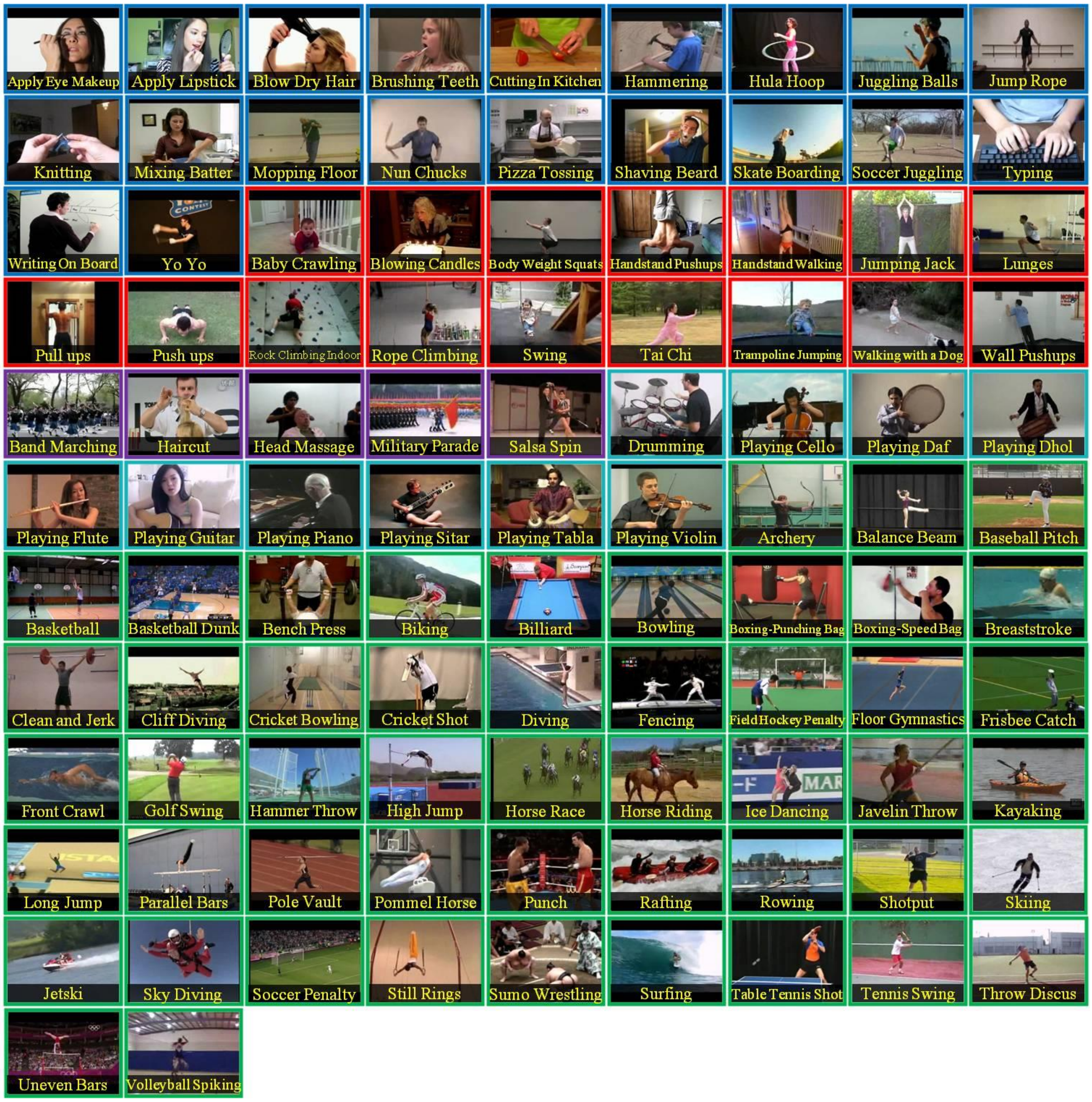


Figure 3.1. UCF101 Dataset.

3.2 Statistics

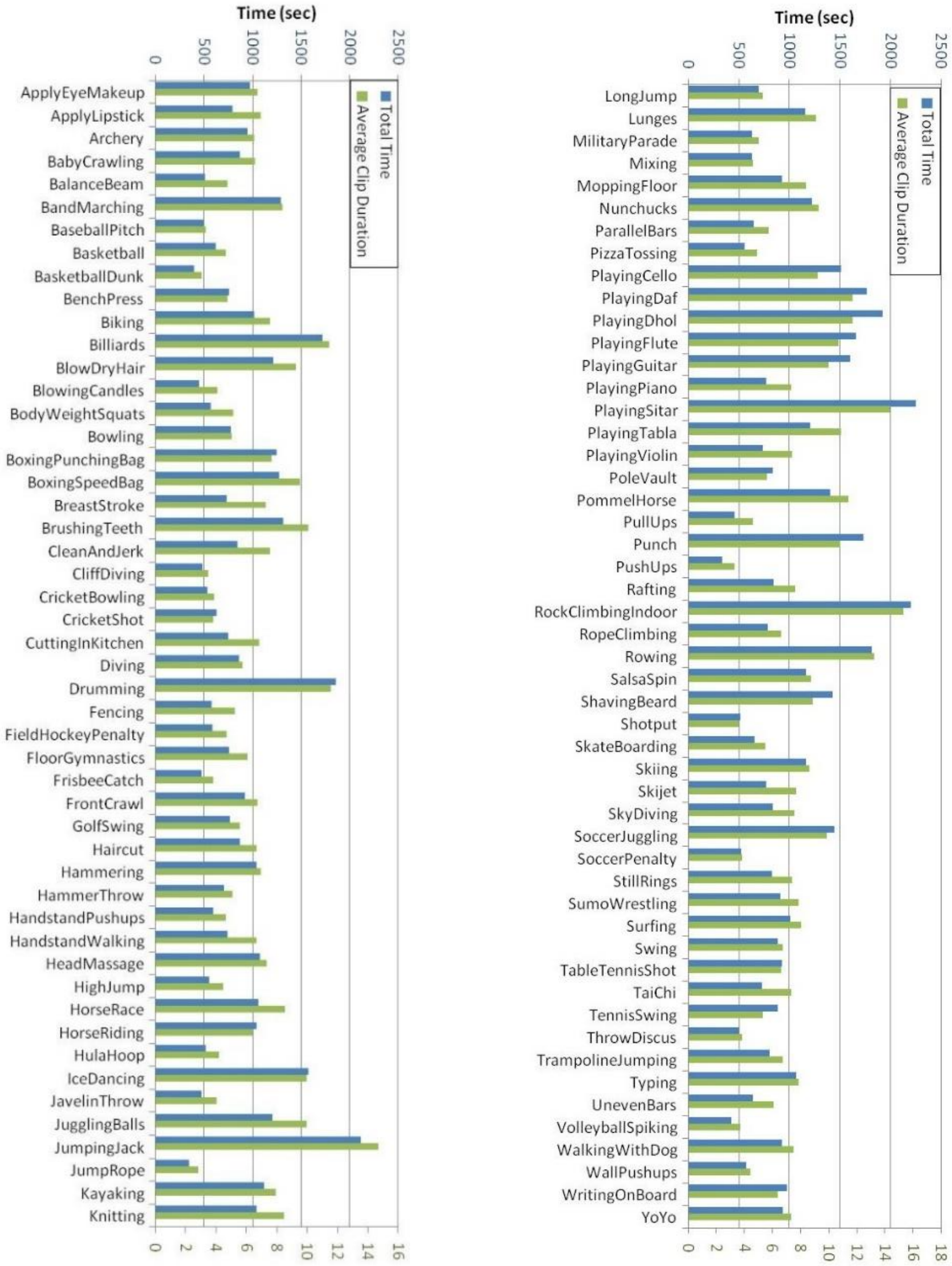


Figure 3.2. Total and average time of videos for each actions shown in deffierent colors.

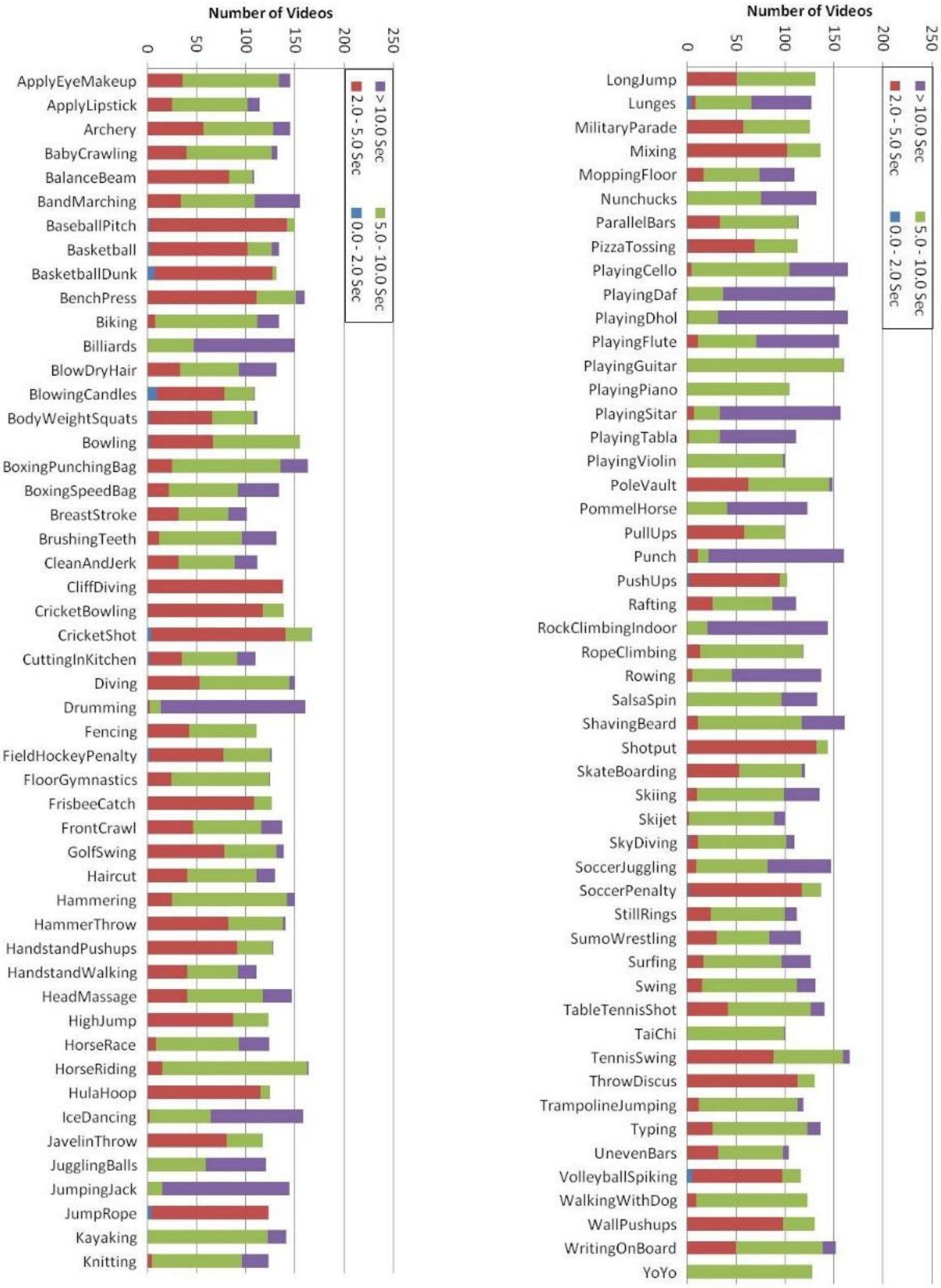


Figure 3.3. Clips per action shown in different colors.

CHAPTER 4: METHODOLOGY

Proposed methodology will be presented deliberatively in this chapter. A deeper-level method based on two-stream Conv Networks will be split in two parts. First, spatial stream of the network will be discussed in contrast to the original spatial stream. After that, techniques for the enhancement of the spatial stream will be proposed. Next, we discuss the temporal stream of our network along with the techniques for extracting the optical flow inputs and different variation of the inputs provided to the network. Finally we propose different techniques to minimize the problem of overfitting. Implementation details are also discussed at the end of this chapter.

The original two-stream ConvNetworks [11] performed well in the job of human action recognition. Proposing two separate streams and combining results later by fusing the streams was a break through in terms of performance by that time. The initial two stream ConvNetworks are comprised of two independent recognition streams consisting of the spatial and the temporal streams, both are subsequently merged via late fusion through averaging. Spatial net recognizes actions from static video frames and can be considered as a simple image recognition stream, whereas the temporal network is taught to identify action classes from dense optical flow data, which are unlike ordinary RGB frames and contains information of the motion between two consecutive frames. Details of such input will be discussed later in this thesis.

4.1 Two-Stream Network Architecture

Figure 4.1 depicts a summary of the proposed deep two-stream networks where we can see both stream producing ther softmax scores. These scores are later fused together by averaging to produce final results. In the original network from [11] both streams initially have the same convolutional network as shown in Fig 4.2. The network is a 7 layer network with five convolutional layers followed by two fully connected layers. Summary of the model is shown in Table 4.1.

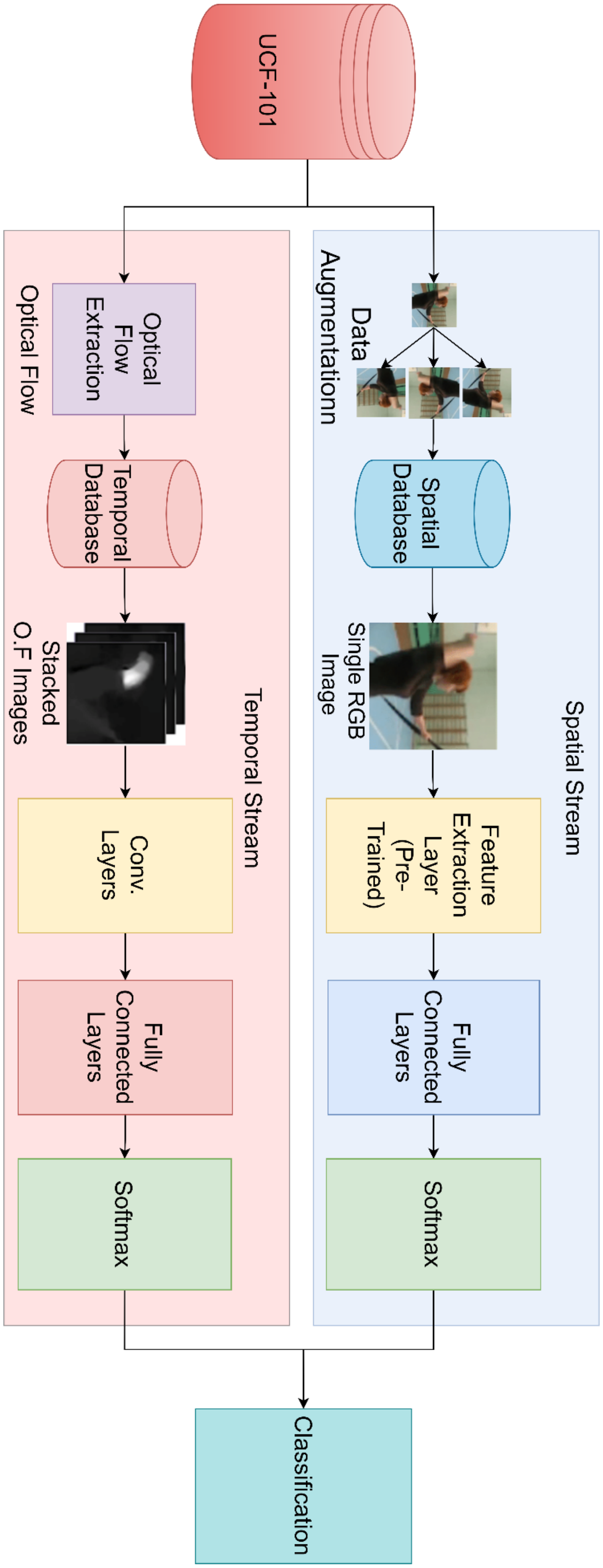


Figure 4.1. Initial pipeline of our architecture

The network take as input stacks of optical flow fields (224x224x2F, F is the number of stacking flows) which contain information of the motion between two consecutive frames and is unlike standard RGB frame which will be used as an input for the spatial stream because we don't need motion feature in that stream.

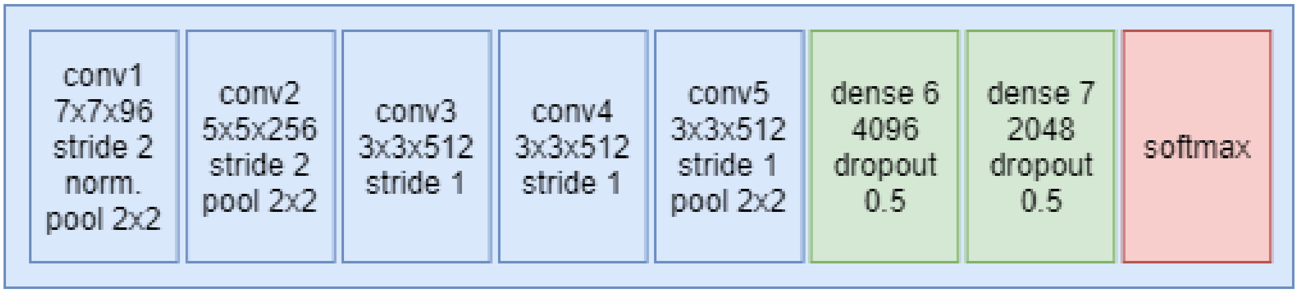


Figure 4.2. Convolutional Network

Table 4.1. ConvNet Architecture for Temporal Stream

Layers Name	**Size**	**Output**
Conv 1	96, 7x7	108x108
Conv2.x	256, 5x5	52x52
Conv3.x	512, 3x3	50x50
Conv4.x	512, 3x3	48x48
Conv5.x	512, 3x3	46x46
Dense Layer 1	-	-
Dense Layer 2		
Softmax	-	-

4.2 Spatial Stream Conv. Networks

For some activities a single frame from the entire video can be enough to recognize the activity. This can be true usually for human-object interaction activities like playing guitar or hammering because recognizing an object in the image can lead to recognizing the associated activity. For this reason, spatial streams take a single RGB image as an input for image recognition or eventually activity recognition in our case.

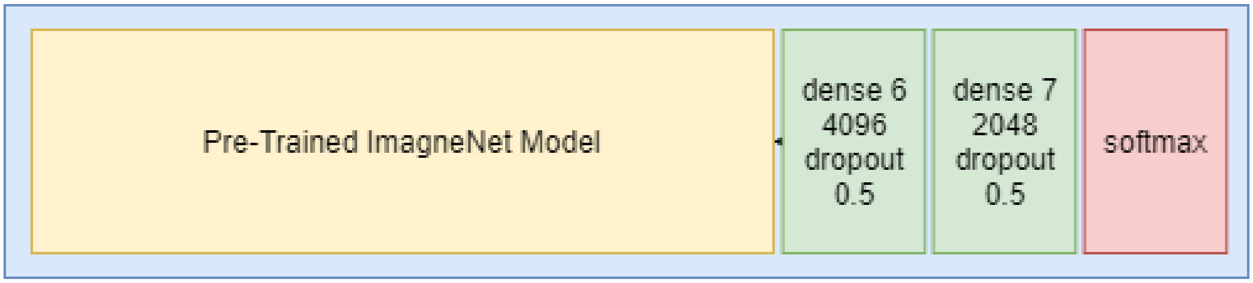


Figure 4.3. Enhanced Spatial Stream

4.2.1 **Satial Stream**

As spatial stream is in actual an image recognition architecture, we can use advance models like ILSVRC winners pre-trained on large datasets like ImageNet dataset and fine-tune them on UCF-101 to form an Enhanced Spatial Stream. The architecture of the enhanced stream is shown in Figure 4.3 where convolutional layers of the pre-trained models are used only and are merged with the fully connected layers (6 and 7) of the riginal model (Figure 4.2).

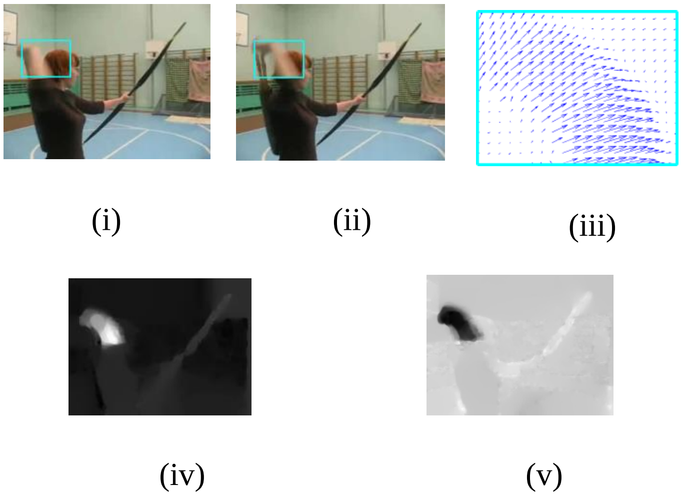
We have found that there are some common statistics between ImageNet and UCF-101, e.g., “WalkingWithDog” in the UCF involves a dog class, whereas the ImageNet also contains samples of dog, like “MalteseDog” etc. Because of this link, we models trained on ImageNet like the ILSVRC winners.Different models listed in Table I will be evaluated one by one, by training the fully connected layers only. The best performing model will be selected and further fine-tuned to check for any further improvement in performance. This will be further discussed in chapter 5.

Table 4.2. Pre-trained Models

Models (Pre-trained on ImageNet)	**Training Dense Layers**	**Fine-Tuning Whole Network**
InceptionV3	159	23,851,784
VGG16	23	138,357,544
Xception	126	22,910,480
MobileNet	88	4,253,864
MobileNetV2	88	3,538,984
DenseNet121	121	8,062,504
DenseNet169	169	14,307,880

4.3 Temporal Stream

Spatial stream could have been enough for activity recognition problem but there are time oriented activities like Running and Jogging. We simply cant differentiate between these two activities by just looking at a single frame of an entire video of these two activities. So, a single RGB input in spatial stream is no longer useful for such activities. To cope this problem there comes the temporal stream. We describe a ConvNet model for temporal stream in this section and the inputs used for that stream. Optical flow displacement fields are stacked multiple times in order to form the input to our model. With explicit description of motion, the network is freed of the need to estimate motion and can focus on pattern recognition. Below, we present several different variations of the optical flow-based input.


(i)

(ii)

(iii)

(iv)

(v)

Figure 4.4. Optical flow. (i),(ii): caonsective video frames. (iii): dense optical flow close-up; (iv),(v): horizontal and vertical components of optical flow:

4.3.1 **Stacked Optical Flow**

As a dense flow of optical vectors, dense optical flow looks like a set of displacement vector fields, with one vector field for each pair of consecutive frames. Displacement vector from point (u, v) in frame t, which moves the point (u, v) from its current location in frame t to the new location in frame $t + 1$, is given by $\mathbf{d}_t(u, v)$. When viewed as image channels (Figure 4.3), the horizontal and vertical elements of the vector field, d^x_t and d^y_t, are well suited to use in a convolutional network for recognition. We stack the $d_t^{x,y}$ t flow channels of L consecutive frames to make a total of $2L$ input channels, with the added benefit of being able to follow motion in a sequence of frames. A ConvNet input volume for an arbitrary frame is then constructed as follows:

$$\begin{aligned}&I_\tau(u,v,2k-1) = d^x_{\tau+k-1}(u,v),\\ &I_\tau(u,v,2k) = d^y_{\tau+k-1}(u,v), u = [1;w], v = [1;h], k = [1;L].\end{aligned}$$ 4.1

The following channel encodes the motion at the point (u, v) over a sequence of L frames: $I_\tau(u,v,c), c = [1;2L]$ (as illustrated in Figure 4.4 - left).

4.3.2 Stacked Trajectories.

Using the motion trajectory-based descriptors [48], As an alternative to optical flow, which is collected at the same locations over many frames, a motion representation that tracks motion path, termed trajectory-based descriptors, is used. In this case, the input volume $I\tau$ is expressed as:

$$\begin{aligned}&I_\tau(u,v,2k-1) = d^x_{\tau+k-1}(\mathbf{p}_k),\\ &I_\tau(u,v,2k) = d^y_{\tau+k-1}(\mathbf{p}_k), u = [1;w], v = [1;h], k = [1;L]\end{aligned}$$ 4.2

The k-th point along the trajectory is represented by the following recurrence relation: (u, v) and $\boldsymbol{p}_k$ is the k-th point.

$$\mathbf{p}_1 = (u,v); \mathbf{p}_k = \mathbf{p}_{k-1} + \mathbf{d}_{\tau+k-2}(\mathbf{p}_{k-1}), k > 1$$ 4.3

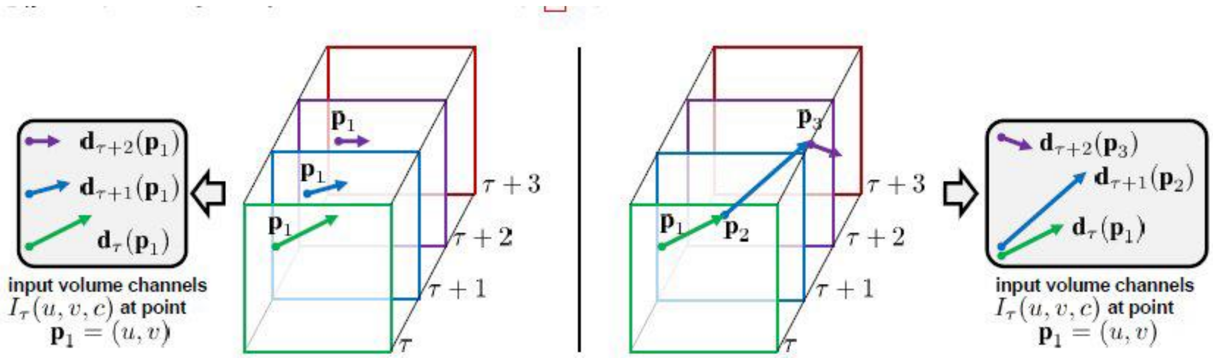


Figure 4.5 ConvNet input derivation from the multi-frame optical flow. Left: optical flow stacking (4.1). Right: trajectory stacking (4.2). Same color used for the frames and the corresponding displacement vectors [11]

The displacement vectors in channels $I_\tau(u, v, c)$ are stored at locations (u, v) in the input volume (4.1), whereas displacement vectors sampled at locations $\mathbf{p}_k$ along the trajectory are stored in the input volume (4.2). (as illustrated in Figure 4.4 - right).

4.3.3 Directional and Bi-directional Optical Flow.

The forward optical flow (i.e. the displacement field dt of the frame t specifies the location of its pixels in the following frame t + 1) is covered by optical flow representations (4.1) and (4.2). Adding a new set of displacement fields to a bi-directional optical flow yields an extension to an optical flow that is bi-directional in nature. The final input volume $I\tau$ is formed by stacking L/2 forward flows between frames $\tau + L/2$ and τ and stacking L/2 backwards flows between frames $\tau - L/2$ and τ. Even though $I\tau$ now has two channels, the input still has the same number of channels (2L). Flow is presented using two techniques (4.1) and (4.2).

4.3.4 The Mean Flow Subtraction

Reducing the system input's total cost by zero-centering the network input generally helps the model, since it makes the system better use the rectification non-linearities. When displacement vector field components have both positive and negative values, they naturally have values that are centred around zero. However, in a pair of frames, just one displacement, such as the movement of the camera, would significantly affect the optical flow (a computation of how objects move between two pictures). This highlighted the significance of camera motion correction since in [42, 49] an overall motion component was calculated and than removed from the densed flow. Suggested strategy is to simply remove the mean vector from each displacement field.

4.4 Getting Rid of Overfitting

4.4.1 Overfitting

Making an extremely complicated model that describes irregularities in the data under examination is an example of overfitting the model. In truth, most data studies contain some level of error or random noise. Attempting to make the model adhere to slightly inaccurate data too closely will infect the model with significant flaws and diminish its prediction potential. For example, finding trends in large datasets of historical market data using computer algorithms is a typical challenge. With enough research, it is often able to build complicated theorems that appear to accurately anticipate stock market returns.

For example, finding trends in large datasets of historical market data using computer algorithms is a typical challenge. With enough research, it is often able to build complicated theorems that appear to accurately anticipate stock market returns. When unseen data is shown to them, however, such model may turn out to be nothing more than the overfitting of a model to what were in reality just random events. It’s critical to test a model on unseen data as well to get rid of such scenarios before finalizing the model.

4.4.2 Precautions

k-fold cross validation is a technique where the data used to train the model is cut into folds or divisions and deployed and the model is run for each fold, is one way to avoid overfitting. We average the error across each fold to get an estimate. Some other techniques include ensembling, which combines predictions from at least two different models, data augmentation, which makes the existing data set appear more diverse, and data simplification, which streamlines the model to minimise overfitting.

4.4.3 **Overfitting in artificial intelligence**

In artificial intelligence or more specifically in machine learning, overfitting is also a factor. It may appear when a computer has been taught to scan for certain data in one manner, but the findings are inaccurate when the same method is applied to a new collection of data. This is due to mistakes in the model, which most likely has a low bias and a large variance. It's possible that the model has duplicate or overlapping characteristics, making it overly complex and unproductive.

4.4.4 **Overfitting vs. Underfitting**

An overfitted model may be very intricate, rendering it useless. However, a model might be underfitted, which means it is too simplistic, with too few characteristics and too little data, to be useful. A low bias and high variance model is called an overfit model, whereas a high bias and low variance model is called an underfit model. Adding extra characteristics to a model that is too simplistic might assist in reducing bias.

4.4.5 **Example of overfitting**

For example, a university with a higher than desired college dropout rate chooses to develop a model to estimate the chance that an applicant would complete their studies.

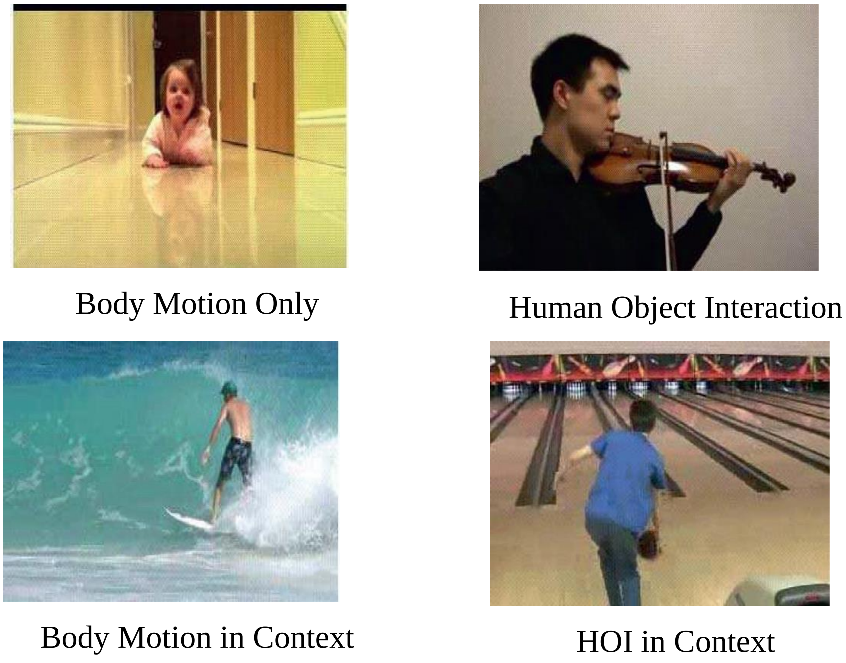
To do so, the institution uses a dataset of 5,000 candidates and their outcomes to train a model. It then applies the model to the original dataset—the 5,000 applicants—and the model correctly predicts the outcome 98 percent of the time. They also ran the programme on a second dataset of 5,000 more candidates to assess its accuracy. This time, however, the model is only 50% correct since it was fitted too closely to a small data sample, in this case, the first 5,000 applicants.

4.4.6 Transfer Learning in Spatial Stream:

In transfer learning, the information of a previously trained machine learning model is transferred to a new but related issue. For example, if you trained a basic classifier to predict if a picture includes a backpack, you might utilise the information that the model acquired throughout its training to detect additional items like sunglasses. To put it another way, transfer learning is the process of attempting to use what has been learnt in one activity to enhance generalisation in another. At "task A," we transfer the weights that the network has learnt to a new "task B," which is a new task. To summarise, the basic concept is to apply the knowledge a model has gained from a task with a large amount of accessible labelled training data to a new task with limited training data. Instead of starting at the beginning of the learning process, we start using patterns that have been learnt by completing a similar assignment.

Transfer learning is most often employed in computer vision and natural language processing applications such as sentiment analysis because to the large amount of computing resources needed for these types of jobs. Even while transfer learning isn't technically a machine learning approach, it may be thought of as a "design methodology" within a particular area, such as active learning, for example. It is also neither a subset or a study-area that is unique to machine learning. Nonetheless, it has gained widespread acceptance when used in conjunction with neural networks, which need massive quantities of data and computing capacity.

If we take computer vision as an example, neural networks are often trained to identify edges in the first few layers, forms in the middle layer, and certain task-specific characteristics in the later levels as they go through the layers. Transfer learning makes use of the early and intermediate layers, with the later layers being retrained only after they have been utilised. It helps utilise the labelled data of the job it was originally trained on.



Body Motion Only

Human Object Interaction

Body Motion in Context

HOI in Context

Figure 4.6. Action Classes of UCF-101.

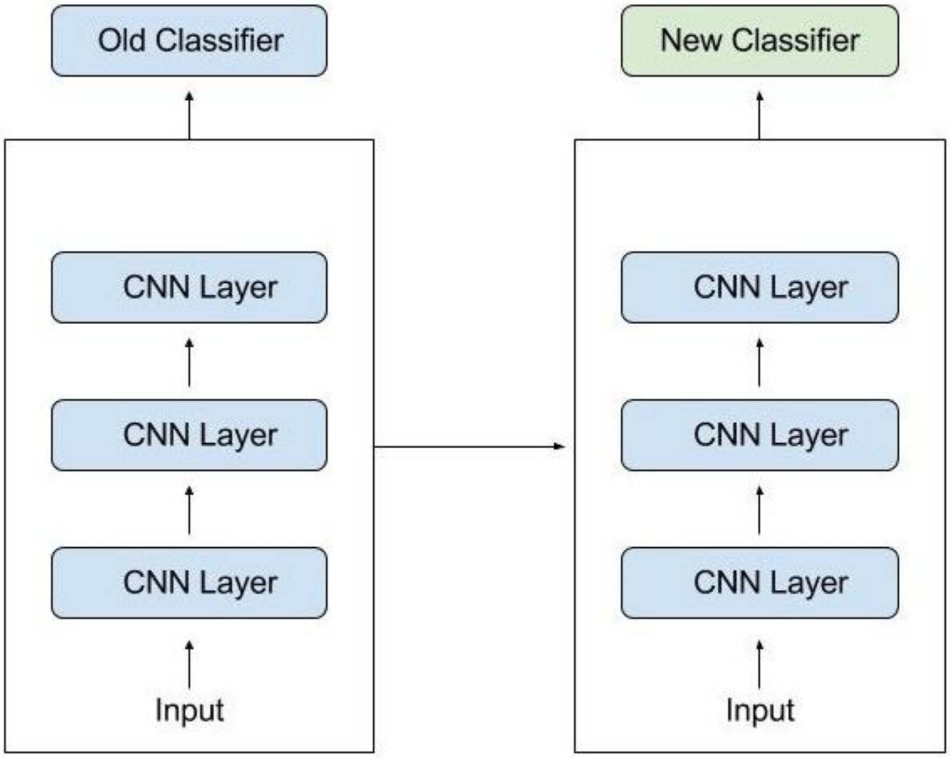


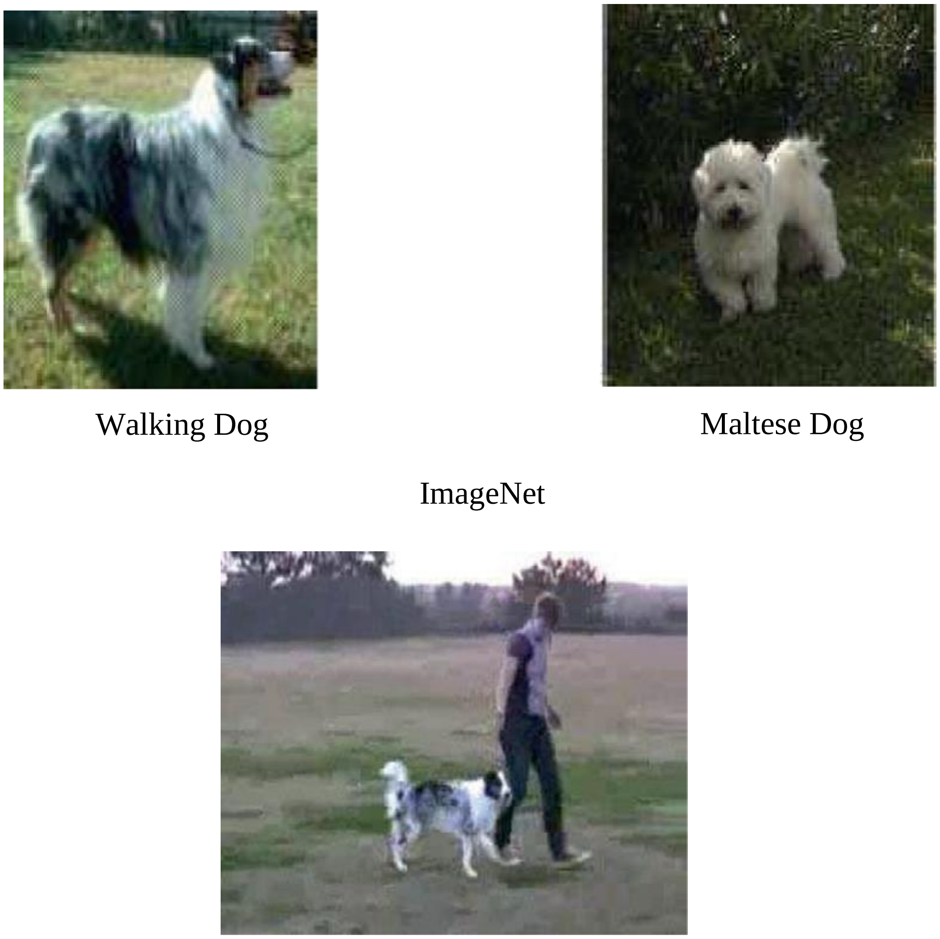
Figure 4.7. Classifier Transfer Learning.

Let’s go back to the example of a model trained for identifying a backpack on a picture, which will be used to identify sunglasses. Due to the fact that the model has already learnt to identify items in the early levels, we will simply retrain the subsequent layers to understand what distinguishes sunglasses from other objects in the later layers.

In transfer learning, we attempt to transfer as much information as possible from the prior task the model was trained on to the current task at hand. Depending on the issue and the data, this knowledge may take on a variety of shapes and forms. For example, it may be the way models are put together that makes it easier for us to recognise new things in our environment.

UCF for human-action-recognition are mostly from daily life, and their classification may be broadly classified into 4 categories, shown in Figure 4.6: (1) Only body-motion, motions described by by human movement only, such as "Running"; (2) Interaction of object and human, actions described by involment of some object, like "Discuss Throw"; (3) body-motion in context, action described by movement of body occurring in a specific environment, such as "Water Surfing"; (4) human-object interaction in context, actions containing representative objects and occurring in a specific context, such as "Surfing";

As all human-action type must be recognised by high-level visual signals such as interaction of human-object, scene-context, and posture-change [8], ImageNet trained model maybe thought of as understandings of object categories in the mid-level. Based on our research, it is found that there are ‘common statistics' in the ImageNet and UCF-101, e.g, “WalkingWithDog” in the UCF involves a “dog” class, whereas the Image Net also contains samples of dog, like “MalteseDog” etc., shown in Figure 4.8. Because of this link, we transmit weights from a model which is pre-trained on ImageNet to deeper ConvNetworks for model initialization.



Walking Dog

Maltese Dog

ImageNet

WalkingWithDog

UCF101

Figure 4.8. Commonalities between the two datasets.

During this procedure, the parameters trained on ImageNet were transferred to our deeper spatial network. However, since the input of the deeper temporal net was volumes of stacking optical flow fields (rather than colored pictures), the channel of the first layer C1 in the temporal network were not equal to ResNet-101 (20 vs 3), we were unable to perform transfer learning in this stream.

4.4.7 Dataset Augmentation for Spatial Stream

Data augmentation as shown in Figure 4.8 is a technique that allows practitioners to substantially enhance the variety of data available for training models without gathering new data. When training large neural networks, data augmentation methods such as trimming, padding, and vertical flip are frequently employed. A simple augmentation is utilised in the majority of neural network-training methods. Data augmentation and data augmentation strategies that capture data invariances have received less attention than neural network designs. This approach was applied to our dataset to increase the number of images. Vertical flip, horizontal flip and 90 degree rotation was used to increase the dataset by three folds. As a result it enhance the current state of the art results by overcoming the overfitting problem as explained above

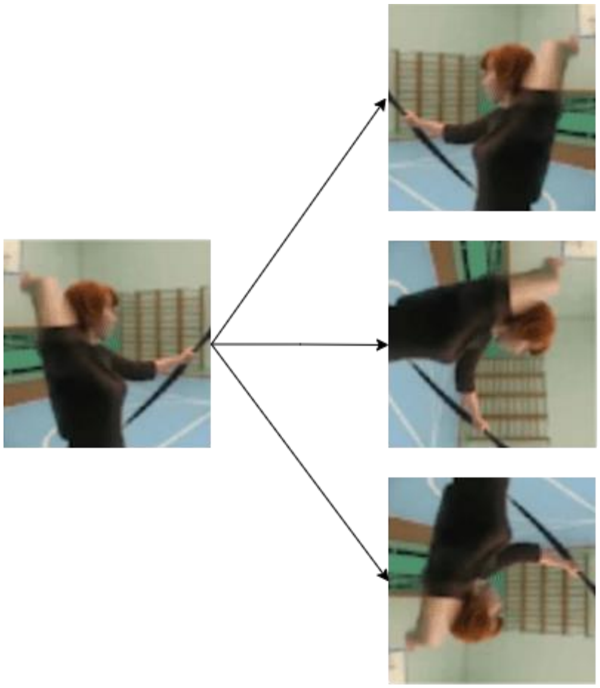


Figure 4.9. Data Augmentation using flip and rotate operations to image.

4.4.8 Data Variation for Temporal Stream

As opposed to images, video is a three-dimensional data set with changeable temporal duration. As a result, pre-processing is typically required when using ConvNetworks for recognition of action . Original two-stream ConvNetworks [11] split movies into frames based on time intervals, and tempotal information was represented by acquairing the optical-flow fields between those frames. Data redundancy across successive frames, on the other hand, would result in a lack of discriminative ability for action recognition. Rather than just clipping the prominent areas of the picture centre as in [25], we incorporated a method of data variation in the proposed work's training to enhance data variety. With a fix frame size of 256x340, each of the frame was chopped four corners and one centre by randomly choosing width and height from 256, 224, 192, 168, which was intended to take use of multiple scale representations. After resizing the clipped areas to 224x224 and flipping them horizontally, there are 10 inputs for the proposed model training (4 corners, 1 centre, and their horizontal flipping). This kind of augmentation method significantly increases the variability of inputs, which also helps to eliminate the issue of overfitting.

4.5 Implementation Details

4.5.1 Network Configuration

Figure 4.1 depicts the layer structure of our temporal ConvNetworks. It is comparable to the network of [11] and corresponds to the CNN-M-2048 design of [50]. The rectification (ReLU) activation function is used for all hidden weight layers; maxpooling is done across 3x3 spatial stream windows with a stride of 2; and local response normalisation is performed using the same parameters as in [25].

4.5.2 **Training**

The training method is similar for both spatial and temporal Networks and may be regarded as a modification of that of [25]. The mini-batch stochastic gradient descent with momentum method is used to learn the network weights (set to 0.9). Each cycle generates a mini-batch of 256 samples. A 224x224 sub-image is randomly clipped from the chosen frame during spatial net training. The movies were previously rescaled such that the least side of the frame equaled 256. Unlike [25], the small image is sampled from the whole frame, and not just from the 256x256 centre. We calculate an optical flow volume I for the chosen training frame during temporal net training. A fixed-size 224x224x2L input is randomly chopped and flipped from that volume. The learning rate is first set at 10-2 and then gradually reduced according to a predetermined schedule that is maintained throughout all training sets. Because of the benefit of transfer learning, the temporal stream is trained for 50K iterations whereas the spatial stream is taught for just 10K iterations.

4.5.3 **Testing**

For temporal stream, we select a predetermined number of chunks (5 in our case) from each video with equal temporal gap between the chunks. We then extract 10 frames [25] from each chunk and pass those as input for validation. The class results throughout the whole video are then calculated by averaging the results from all chunks. Spatial stream validation is also carried out in the same manner with the difference that only a single frame from the predetermined number of chunks is passed to the network for validation.

CHAPTER 5: EXPERIMENTAL RESULT

In this section, evaluation protocols for both the streams are discussed first. After that, we state the results of the temporal stream obtained by providing different schemes of optical flow input. Then, results from the spatial stream are discussed where we contributed the most. Results from both streams are also compared with other method. Finally, we discuss the final results obtained by the fusion of spatial and temporal stream which are then compared with the state of the art results.

5.1 Evaluation Protocol

We performed the experiment on UCF-101 dataset which is the benchmark for action videos and is currently the largest dataset available as well in this field of computer vision. It contains 101 different classes which can be split into four categories. There are almost 13k videos in the entire dataset. Other details are already mentioned in Chapter 3. For evaluation we have used k-fold cross validation instead of conventional train/test split. The training set contain almost 9k videos and test set contain almost 4k videos. The data list used for splitting the dataset into train and test set for all three splits is publicly available on web. Performance is measured by using classification accuracy across each split and mean classification accuracy of three splits in the end. Comparisons are done with different architecture based on accuracy across split-1 and for comparison with the state of the art we used mean classification accuracy of three splits.

5.2 Temporal ConvNet

We first evaluated the temporal stream architecture by providing the network with the single and dense optical flow input which is discussed previously. Performance was measured by training the architecture from scratch on UCF-101 with different input configurations. First, we used a single optical flow as an input with a dropout rate of 0.5 for better generalization. Single optical flow frame did not provide impressive results with only 71.6% accuracy, so we used dense input this time by stacking 5 frames and observed an increase of almost 7% in the results. Further increasing the stacking (L=10) does not help significantly as compared to previous setting, so we kept it to L=5. Results in Table 5.1 clearly shows that using dense staking of optical flow (L>1) yields good results as compared to using a single frame. This proves the importance of the temporal aspect of an activity. Figure 5 shows the accuracy curve of the temporal stream by plotting the training and testing results whereas Figure 6 shows the loss curves of the stream.vely.

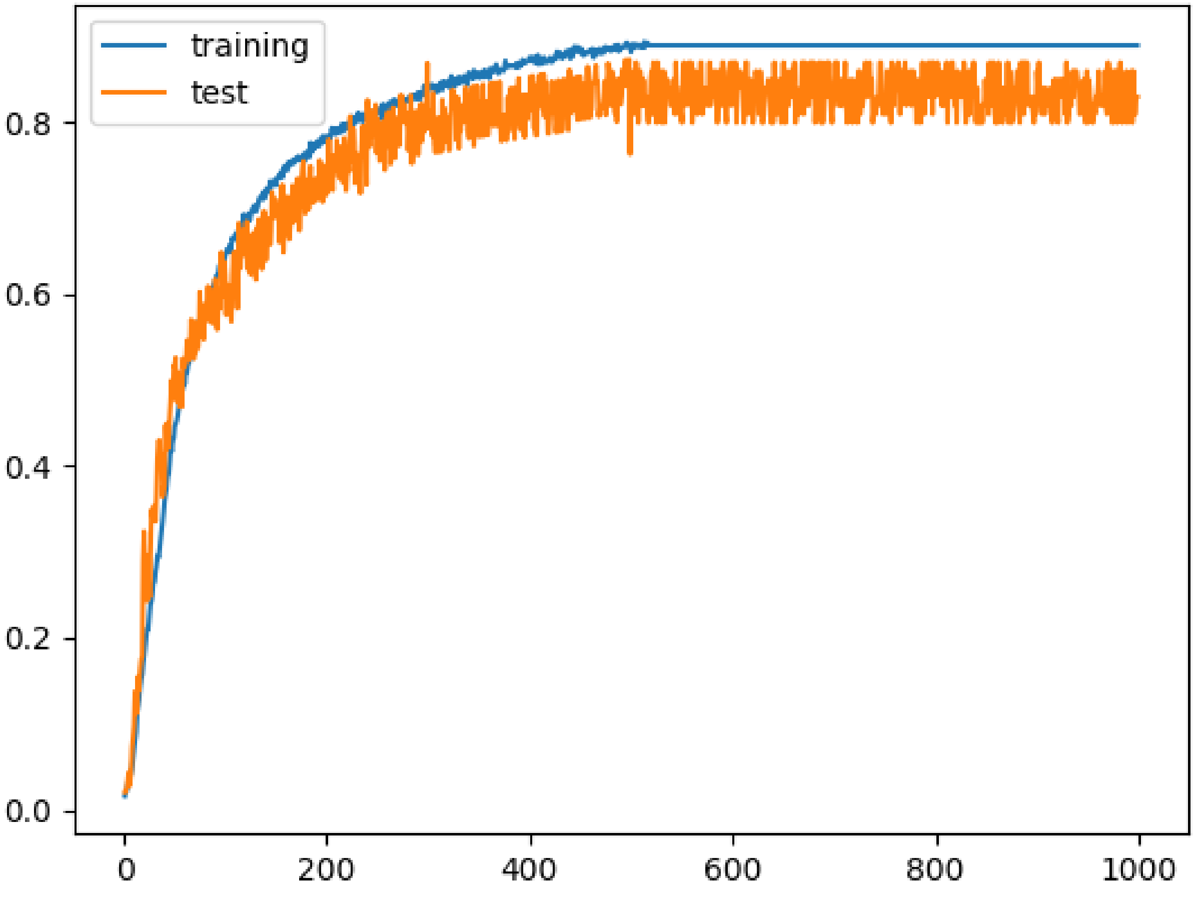


Figure 5.1. Training and testing Accuracy curve for temporal stream

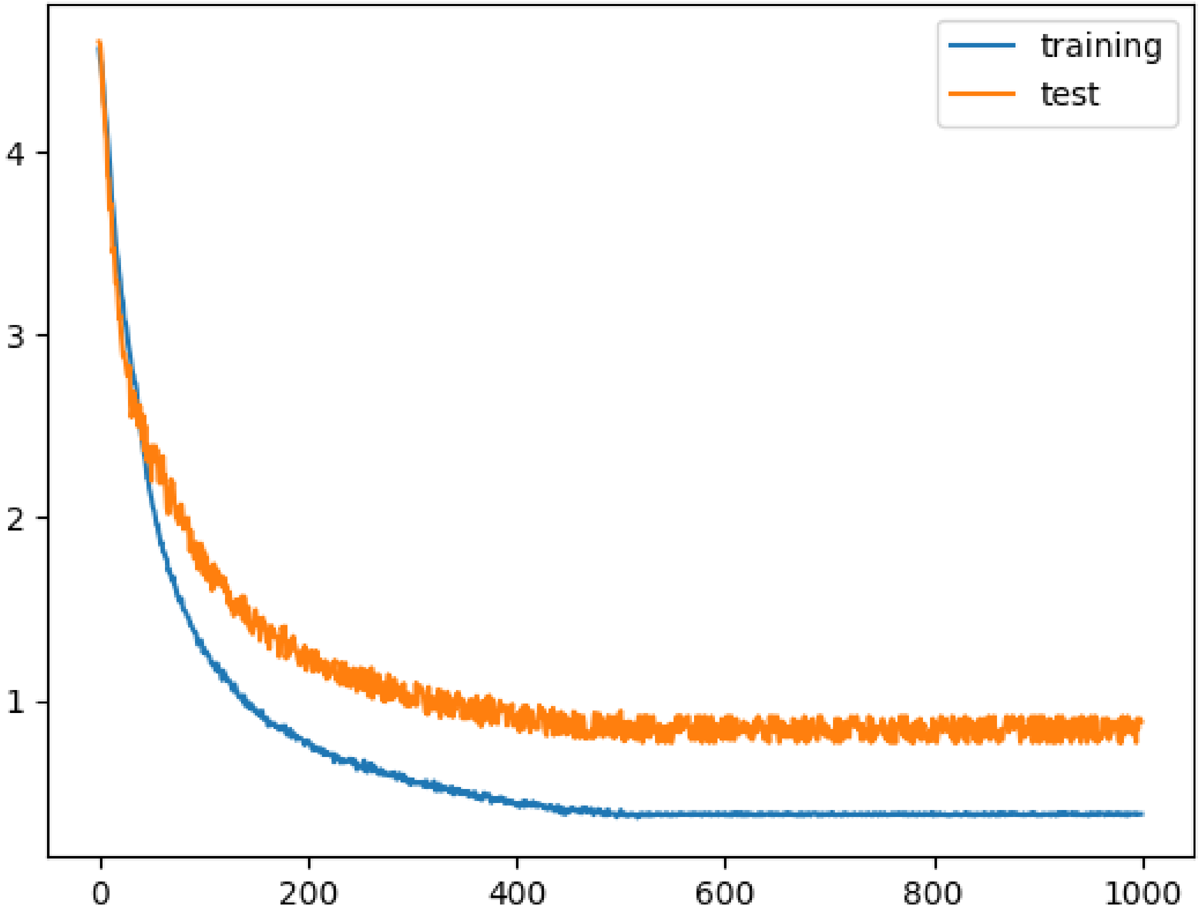


Figure 5.2. Training and testing Loss curve for temporal stream

Table 5.1. Temporal Stream Performance On UCF-101 (split 1)

Input Configuration	**Dropout rate**
	0.5
Single-frame Optical Flow (*L*=1)	71.6%
Multiple Optical Flow (*L*=5)	78.3%
Multiple Optical Flow (*L*=10)	80.2%

5.3 Spatial ConvNet

To assess the spatial stream, we consider three scenarios. First, we deployed the original architecture shown in Figure 2 and trained it from scratch on UCF-101 with same configurations as of temporal stream. This took a lot of time to train and showed poor results with just an accuracy of 41.60%. Secondly, we adopted the enhanced spatial stream, for which we first evaluated the pre-trained models listed in Table 4.2 by training them on our dataset. And finally, we fine-tune the best performing model on augmented dataset.

Figure 7 states the performance of enhanced spatial stream by using different pre-trained models and then fine-tuning them on UCF-101. We can see in Table 5.2 and 5.3 that MobileNet gives the best performance with an accuracy of 75.23%. Moreover, Figure 7 also gives us the idea that almost every pre-trained model we utilized performed better than the original model. Fine-tuning the enhanced spatial stream on UCF-101 leads improvement because ImageNet and UCF-101 datasets are slightly different in nature and the feature extraction part still need to learn the dataset through fine-tuning.

We picked MobileNet model since it has best performance comparatively. After that we trained classification layers and fine-tuned the entire model with the augmented dataset this time. It can be seen in Table 5.4 that it outperformed the standard UCF-101 dataset with an increase of 1.5% in accuracy.

Table 5.2. Training (classification layers) resuts on pre-trained models for spatial stream

Models (Pre-trained on ImageNet)	**Training Classification Layers only**
InceptionV3	68.09%
VGG16	55.62%

Xception	63.46%
MobileNet	**74.26%**
MobileNetV2	66.51%
NASNetMobile	57.50%
DenseNet121	66.09%
DenseNet169	66.26%

Table 5.3. Fine tuning resuts on pre-trained models for spatial stream

Models (Pre-trained on ImageNet)	**Fine-Tuning Whole Network on UCF-101**
InceptionV3	69.93%
VGG16	56.69%
Xception	66.42%
MobileNet	**75.23%**
MobileNetV2	68.25%
NASNetMobile	60.13%
DenseNet121	68.23%
DenseNet169	67.95%

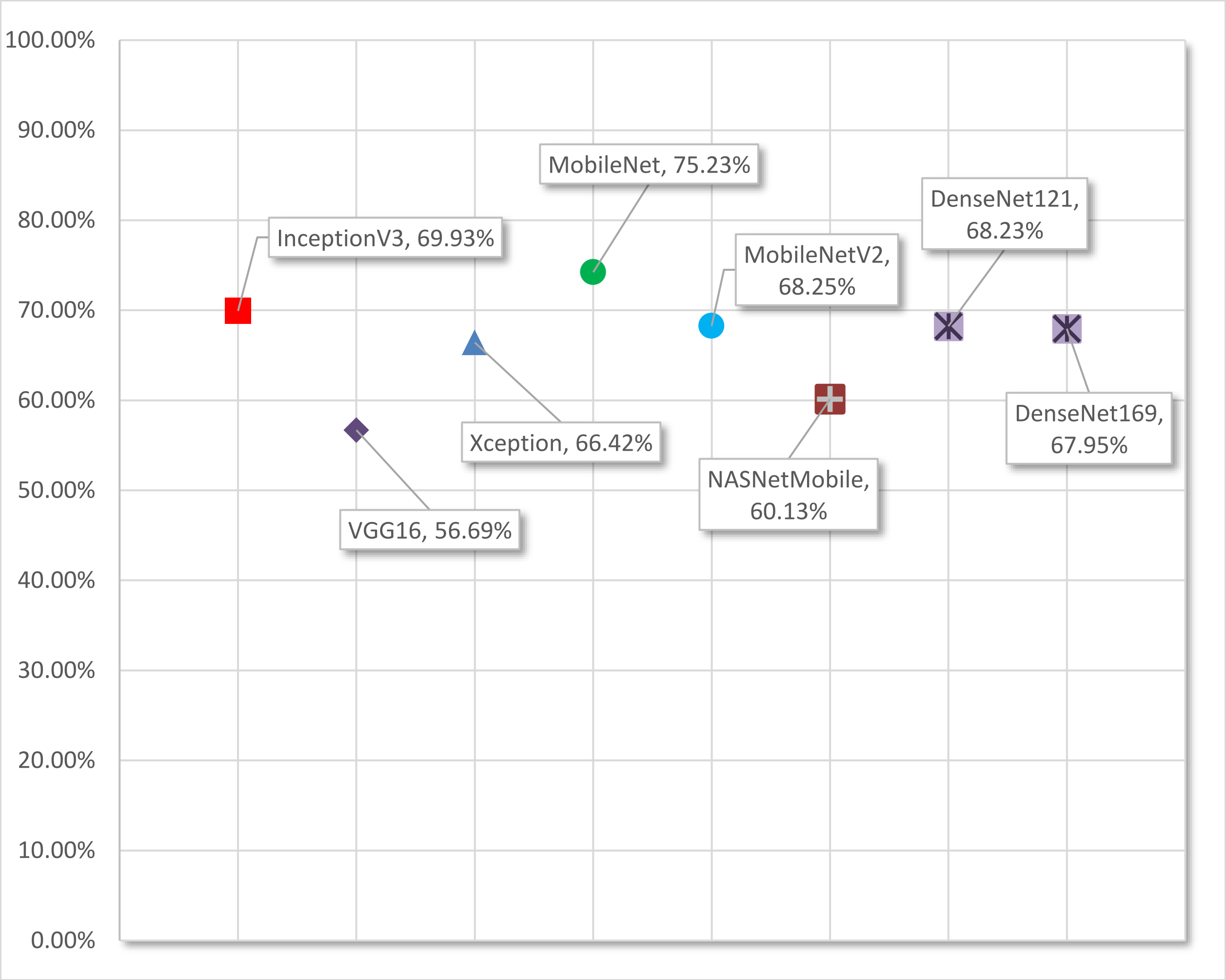


Figure 5.3. Pre-trained model results comparison for spatial stream

Table 5.4. Spatial stream performance

Training Configuration	**Accuracy**
Training from scratch	41.60%
MobileNet (fine-tuning on UCF-101)	75.23%
MobileNet (fine-tuning on augmented UCF-101)	**76.70%**

5.4 Two Stream Network

In this section we combine assess the overall two-stream model. This is done by combining the temporal and enhanced spatial stream. Multiple strategies have been adopted in recent years by researchers to combine the two-streams. One possible approach was to make a stack of joint layers on top of classification layers and then train it, but this led to overfitting, so we fused the streams by averaging their softmax scores. Results in Table 5.5 shows the significance of combining both stream as the overall accuracy is 9.43% greater than the temporal stream results and 12.93% greater than the spatial stream.

Table 5.5. Fusion results of both streams on UCF-101 (split 1)

Training Configuration	**Accuracy**
Temporal Stream	80.20%
Enhanced Spatial Stream	76.70%
Fusion by Averaging	**89.63%**

5.5 Comparison with state of the art

At last, we compare overall results of our approach with the state-of-the-art methodologies by comparing the mean accuracies over three splits of UCF-101. For that the temporal stream was trained on dense optical flow images which were extracted beforehand, with a stack of L=5 frames. Spatial stream on the other hand used a pre-trained model (MobileNet) trained on ImageNet dataset. Spatial stream was further fine-tuned on augmented dataset which led to some improvements in the results Softmax scores from both the streams were fused together in the end by averaging softmax scores to produce results. We first compared the results of both the streams with other state of the art methods. Table 5.6 shows the comparisons as well as the models used by other methodologies in the motion and appearance stream. We can see that our spatial stream performed much better than the original spatial stream in [1] i.e. an increase of 6% in accuracy. The temporal stream was not our main concern in this research so we adopted the exact model as in [1]. Results in Table 5.7 shows the overall results comparison with state-of-the-art, and we can see that our results when compared to others, performed well from almost all of them with an accuracy of 91.20%.

Table 5.6. Result comparison of appearance and motion path with other models

Method	**Appearance**		**Motion**	
	Model	**Accuracy**	**Model**	**Accuracy**
[51]	Alex Net	73.00%	CNN	83.70%
[17]	VGG-16	82.61%	VGG-16	86.25%
[16]	Res Nets	82.29%	Res Nets	79.05%
[46]	Res NeXt	85.20%	Res NeXt	87.00%
Ours	MobileNet	79.00%	CNN [11]	82.60%

Table 5.7. Comparison with other models (mean accuracy)

Model	**Accuracy**
Two Stream Network [11]	86.90%
Two-stream Network Fusion [17]	91.40%
Residual Two-stream Network [16]	89.47%
Residual Frames Two-stream Network [46]	90.60%

Temporal Stream Network	82.60%
Enhanced Spatial Stream Network	79.00%
Two Stream Network (Our Model)	**91.20%**

5.6 Discussion

In this research, our goal was to recreate the existing two-stream network for HAR by enhancing its spatial stream and therefore we only compared our method with some corresponding methods as shown in Table 5.6 and 5.7. Our spatial stream result outperformed the original two-stream [11]. In [16, 17, 46] researchers have used very deep networks which requires a lot of computation power and time to train them. Keeping in view the edge they have over us in terms of computational power, our model still outperformed most of them when comparing the overall accuracy over three splits in Table VI.

CHAPTER 6: CONCLUSION & FUTURE WORK

6.1 Conclusion

The current research's key goal is to develop a reliable HAR and user authentication framework using data from smartphone sensors. Various feature extraction tools were investigated in this report, and a comparison was made between them. The suggested framework's robustness was also contrasted with other published research. Here we proposed a deep architecture for two streamed ConvNetwork training. Several beneficial practises were also deployed to reduce the overfitting issue posed by a lack of sufficient samples in order to ensure the learning performance. Using a disordering tactic between video listed in the training/testing splits, a critical improvement in human action recognition has been attained in the testing phase, according to the results. The empirical experiments have proved that our proposed architecture have beaten the top of the rank models in term of accuracy, with 95.1 percent on UCF, respectively. When we tested the temporal stream network with deeper ConvNetworks on the UCF101 dataset, we discovered that it performed worse than the spatial Networks in terms of accuracy and performance. We believe that one guranteed way to overcome this limitation is to acquire temporal information with a deep temporal structure and that motivated us to use deep recurrent neural network models to model long-term motion dynamics in the potential research.

6.2 Contribution

- Built the model for user authentication.
- Explored various feature extraction tools.
- Made comparison among the feature’s extraction tools.
- Review & comparison of recent development techniques for activities recognition and user authentication.

6.3 Future Work

While the currently applied methodolgy provides good results for used dataset, future research into new alternatives for the proposed system may be beneficial in enhancing precision. Using Transfer Learning in the Temporal Stream can provide good results. Also, by adding more activities to the existing datasets, overfitting can be further minimized. Furthermore, the current study offers a foundational ideological principle for future researchers to investigate more configurations in the stated architecture, which will aid in the HAR system's high achievement level.

REFERENCES

[1] D. Weimer, B. Scholz-Reiter, and M. Shpitalni, "Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection," *CIRP Annals*, vol. 65, no. 1, pp. 417-420, 2016.

[2] P. Pareek and A. Thakkar, "A survey on video-based human action recognition: recent updates, datasets, challenges, and applications," *Artificial Intelligence Review*, vol. 54, no. 3, pp. 2259-2322, 2021.

[3] W. Seok and C. Park, "Recognition of human motion with deep reinforcement learning," *IEIE Transactions on Smart Processing & Computing*, vol. 7, no. 3, pp. 245-250, 2018.

[4] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5533-5541.

[5] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299-6308.

[6] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489-4497.

[7] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794-7803.

[8] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and V. Gool Luc, "Temporal segment networks: Towards good practices for deep action recognition," in *European conference on computer vision*, 2016: Springer, pp. 20-36.

[9] Y. Zhao, Y. Xiong, and D. Lin, "Trajectory convolution for action recognition," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 2208-2219.

[10] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725-1732.

[11] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *arXiv preprint arXiv:1406.2199*, 2014.

[12] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *European conference on computer vision*, 2010: Springer, pp. 140-153.

[13] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and D. Trevor, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625-2634.

[14] S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," *arXiv preprint arXiv:1511.04119*, 2015.

[15] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4694-4702.

[16] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spatiotemporal residual networks for video action recognition. corr abs/1611.02155 (2016)," *arXiv preprint arXiv:1611.02155*, 2016.

[17] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1933-1941.

[18] M. A. Goodale and A. D. Milner, "Separate visual pathways for perception and action," *Trends in neurosciences*, vol. 15, no. 1, pp. 20-25, 1992.

[19] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221-231, 2012.

[20] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential deep learning for human action recognition," in *International workshop on human behavior understanding*, 2011: Springer, pp. 29-39.

[21] Y. Zhou, X. Sun, Z.-J. Zha, and W. Zeng, "Mict: Mixed 3d/2d convolutional tube for human action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 449-458.

[22] Y. Huang, S.-H. Lai, and S.-H. Tai, "Human action recognition based on temporal pose CNN and multi-dimensional fusion," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0-0.

[23] C. Li, Y. Hou, P. Wang, and W. Li, "Joint distance maps based action recognition with convolutional neural networks," *IEEE Signal Processing Letters,* vol. 24, no. 5, pp. 624-628, 2017.

[24] M. Liu, H. Liu, and C. Chen, "Enhanced skeleton visualization for view invariant human action recognition," *Pattern Recognition,* vol. 68, pp. 346-362, 2017.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems,* vol. 25, pp. 1097-1105, 2012.

[26] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert systems with applications*, vol. 59, pp. 235-244, 2016.

[27] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12026-12035.

[28] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3595-3603.

[29] A. Ullah, K. Muhammad, I. U. Haq, and S. W. Baik, "Action recognition using optimized deep autoencoder and CNN for surveillance data streams of non-stationary environments," *Future Generation Computer Systems,* vol. 96, pp. 386-397, 2019.

[30] T. Huynh-The, C.-H. Hua, and D.-S. Kim, "Encoding pose features to images with data augmentation for 3-D action recognition," *IEEE Transactions on Industrial Informatics,* vol. 16, no. 5, pp. 3100-3111, 2019.

[31] S. Das, M. Koperski, F. Bremond, and G. Francesca, "Deep-temporal lstm for daily living action recognition," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS),* 2018: IEEE, pp. 1-6.

[32] R. A. Rensink, "The dynamic representation of scenes," *Visual cognition,* vol. 7, no. 1-3, pp. 17-42, 2000.

[33] S. Das, A. Chaudhary, F. Bremond, and M. Thonnat, "Where to focus on for human action recognition?," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV),* 2019: IEEE, pp. 71-80.

[34] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing,* vol. 234, pp. 11-26, 2017.

[35] F. Moya Rueda, R. Grzeszick, G. A. Fink, S. Feldhorst, and M. Ten Hompel, "Convolutional neural networks for human activity recognition using body-worn sensors," in *Informatics*, 2018, vol. 5, no. 2: Multidisciplinary Digital Publishing Institute, p. 26.

[36] M. I. Razzak, S. Naz, and A. Zaib, "Deep learning for medical image processing: Overview, challenges and the future," *Classification in BioApps,* pp. 323-350, 2018.

[37] C. Li, P. Wang, S. Wang, Y. Hou, and W. Li, "Skeleton-based action recognition using LSTM and CNN," in *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 2017: IEEE, pp. 585-590.

[38] U. Ahsan, C. Sun, and I. Essa, "Discrimnet: Semi-supervised action recognition from videos using generative adversarial networks," *arXiv preprint arXiv:1801.07230,* 2018.

[39] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial*

intelligence and statistics, 2010: JMLR Workshop and Conference Proceedings, pp. 249-256.

[40] W. Liu, H. Zhang, D. Tao, Y. Wang, and K. Lu, "Large-scale paralleled sparse principal component analysis," *Multimedia Tools and Applications*, vol. 75, no. 3, pp. 1481-1493, 2016.

[41] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, 2015: PMLR, pp. 448-456.

[42] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3551-3558.

[43] H. Gammulle, S. Denman, S. Sridharan, and C. Fookes, "Two stream lstm: A deep fusion framework for human action recognition," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017: IEEE, pp. 177-186.

[44] Y. Han, P. Zhang, T. Zhuo, W. Huang, and Y. Zhang, "Going deeper with two-stream ConvNets for action recognition in video surveillance," *Pattern Recognition Letters*, vol. 107, pp. 83-90, 2018.

[45] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 6546-6555.

[46] L. Tao, X. Wang, and T. Yamasaki, "Rethinking motion representation: Residual frames with 3d convnets for better action recognition," *arXiv preprint arXiv:2001.05661*, 2020.

[47] "UCF-101." https://www.crcv.ucf.edu/data/UCF101.php (accessed.

[48] H. Wang, A. Kläser, C. Schmid, and L. Cheng-Lin, "Action Recognition by Dense Trajectories," in *CVPR 2011 - IEEE Conference on Computer Vision & Pattern Recognition*, Colorado Springs, United States, 2011-06-20 2011: IEEE, 2011, pp. 3169-3176, doi: 10.1109/cvpr.2011.5995407. [Online]. Available: https://hal.inria.fr/inria-00583818

[49] M. Jain, H. Jégou, and P. Bouthemy, "Better exploiting motion for better action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2555-2562.

[50] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, 2014: Springer, pp. 818-833.

[51] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034,* 2013.