HUMAN ACTION RECOGNITION FOR VARIOUS INPUT CHARACTERISTICS USING 3 DIMENSIONAL RESIDUAL NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

GÜLİN TÜFEKCİ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2019

Approval of the thesis:

HUMAN ACTION RECOGNITION FOR VARIOUS INPUT CHARACTERISTICS USING 3 DIMENSIONAL RESIDUAL NETWORKS

submitted by **GÜLİN TÜFEKCİ** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _________________

Prof. Dr. İlkay Ulusoy
Head of Department, **Electrical and Electronics Eng.** _________________

Prof. Dr. İlkay Ulusoy
Supervisor, **Electrical and Electronics Eng., METU** _________________

Examining Committee Members:

Prof. Dr. Abdullah Aydın Alatan
Electrical and Electronics Engineering, METU _________________

Prof. Dr. İlkay Ulusoy
Electrical and Electronics Eng., METU _________________

Prof. Dr. Gözde Bozdağı Akar
Electrical and Electronics Engineering, METU _________________

Assist. Prof. Dr. Elif Vural
Electrical and Electronics Engineering, METU _________________

Assist. Prof. Dr. Erdem Akagündüz
Electrical and Electronics Engineering, Çankaya University _________________

Date: 05.09.2019

ABSTRACT

HUMAN ACTION RECOGNITION FOR VARIOUS INPUT CHARACTERISTICS USING 3 DIMENSIONAL RESIDUAL NETWORKS

Tüfekci, Gülin
Master of Science, Electrical and Electronics Engineering
Supervisor: Prof. Dr. İlkay Ulusoy

September 2019, 100 pages

Action recognition using deep neural networks is a far-reaching research area which has been commonly utilized in applications such as statistical analysis of human behavior, detecting abnormalities using surveillance cameras and robotic systems. Previous studies have been performing researches to propose new machine learning algorithms and deep network architectures to obtain higher recognition accuracy levels. Instead of suggesting a network resulting in small accuracy gain, this thesis focuses on evaluating different input characteristics for increasing the learning capacity of the networks. To do so, 3-dimensional residual networks are utilized because of their effective learning process. Among all the modifications applied on the inputs, increasing the sample duration up to 60 frames and masking the RGB pixel values with the motion flow between consecutive frames provide high accuracy gains. Employing 60 frames instead of 16 frames quadruples the computation time while achieving an accuracy increase of 10%. Masking the frames results in 12% recognition accuracy gain. Both modifications contribute to the learning process of the network by emphasizing the relations between patterns through longer temporal extents and guiding the network to focus on the areas where the main action takes place. Obtaining significant amounts of accuracy gains by only modifying the input is outstanding. Moreover, the recognition accuracy is enhanced even more by pre-training the network on a large scale dataset. The contributions of the results of this thesis are worthwhile since the input characteristics yielding high accuracy gains can be used for different networks to increase the recognition accuracy.

Keywords: Action Recognition, Motion Flow, Residual Networks, Spatio-temporal

ÖZ

3 BOYUTLU ARTIK AĞLAR KULLANARAK ÇEŞİTLİ KARAKTER ÖZELLİKLERİNE SAHİP GİRDİLER İÇİN İNSAN EYLEM TANIMA

Tüfekci, Gülin
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği
Tez Danışmanı: Prof. Dr. İlkay Ulusoy

Eylül 2019, 100 sayfa

Derin sinir ağları kullanarak eylem tanıma çok kapsamlı bir araştırma alanıdır ve insan davranışlarının istatiksel analizi, güvenlik kameraları kullanarak olağandışı durumların tespiti ve robotik sistemler gibi uygulamalarda yaygın olarak kullanılmaktadır. Önceki çalışmalar daha yüksek tanıma doğruluğuna ulaşmak amacıyla yeni makine öğrenmesi algoritmaları ve derin ağ mimarileri önermek için araştırma yapmaktadırlar. Yeni bir ağ önerip küçük bir doğruluk kazancı sağlamak yerine, bu tez ağların öğrenme kapasitesini arttırmak için farklı karakter özelliklerine sahip girdileri değerlendirmeye odaklanır. Bu amaçla, efektif öğrenme süreçleri nedeniyle 3-boyutlu artık ağlar kullanılmıştır. Girdilere uygulanan bütün modifikasyonlar arasında, örnek uzunluğunu 60 resme kadar çıkarmak ve KYM piksel değerlerini ardışık resimler arasındaki hareket akışı ile maskelemek yüksek doğruluk kazancı sağlamıştır. 16 resim yerine 60 resim kullanmak işlem zamanını dört katına çıkarırken %10 doğruluk artışına ulaşmaktadır. Resimleri maskelemek %12 tanıma doğruluğu kazancı ile sonuçlanmıştır. İki modifikasyon da daha uzun zamansal kapsam sayesinde örüntüler arasındaki ilişkileri vurgulayarak ve ağın asıl eylemin meydana geldiği kısımlara odaklanmasını sağlayarak ağın öğrenme sürecine katkıda bulunmuştur. Sadece girdiyi modifiye ederek kayda değer seviyelerde doğruluk kazancı sağlamak çok önemlidir. Dahası, öncesinde ağı geniş kapsamlı bir veri kümesinde eğiterek tanıma doğruluğu daha da iyileştirilmiştir. Bu tezde elde edilen sonuçların katkısı; yüksek doğruluk kazançlarını sağlayan girdi karakter özelliklerinin, tanıma doğruluğunu arttırmak amacıyla başka ağlar tarafından da kullanılabilmesi nedeniyle faydalıdır.

Anahtar Kelimeler: Eylem Tanıma, Hareket Akışı, Artık Ağlar, Uzaysal-zamansal

To my dearest family…

ACKNOWLEDGEMENTS

First of all, I would like to express my appreciation to my supervisor Prof. Dr. İlkay Ulusoy for being open-minded, supportive and understanding throughout my study. Her positive attitude, constructive criticism and effort to make time despite her intensive work schedule were valuable for me. I would like to thank her for having such pleasant and motherlike personality.

I am grateful for my family of four for their endless support, understanding and love. My father Nihat Tüfekci, mother Gülçin Tüfekci and my charming brother Emre Tüfekci have stood by me along all the happy and the difficult times of my life. I cannot express my intense feelings towards them and how thankful I am to have them as my family. We will be there for each other whatever may come.

I am also thankful for my friends, especially Burçak Hazal Coşkuner, Anıl Bozyiğit and Andaç Yiğit, since they have always supported me, been there when I needed to have a heart-to-heart talk and shared a great deal of unforgettable memories.

Finally, I feel deeply attached to my beloved one Neşet Berkay Doğan for being so kindhearted and showing unconditional love. He has been there for me with his warm companionship ever since I started university. I am really glad to have a supporting and caring man by my side.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ResNet	Residual Network
RGB	Red Green Blue
RGBF	Red Green Blue Flow
SVM	Support Vector Machine
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
ConvNet	Convolutional Network
RELU	Rectified Linear Units
FC	Fully-Connected
2D	2 Dimensional
3D	3 Dimensional
SGD	Stochastic Gradient Descent
LTC	Long-term Temporal Convolutions
TSN	Temporal Segment Networks
I3D	Inflated 3 Dimensional
EG	Experiment Group
NORM	Normalized

CHAPTER 1

INTRODUCTION

1.1. Motivation

Observing an action that is present in the environment and describing the action to the extent of our knowledge have been performed involuntarily by all human beings. Since humans, by their nature, are capable of learning to interpret the patterns in daily life, such as observing a bird moving in the sky and labeling the action as “flying”, action recognition is a routine part of our lives. With recent technological developments along with studies conducted on artificial intelligence and building models for automating data analysis, the idea of performing action recognition by machine learning methods has been adopted.

The task of action recognition is defined as inferring the patterns and contextualized information, which identify an action, hidden in video clips by performing analysis both spatially and temporally. The desire to use machine learning algorithms for such task has arisen from the high availability of videos provided from various sources and the need to process them. With the extensive usage of surveillance cameras, people [1][2] started to analyze the videos in order to detect suspicious behaviors, theft, criminals, and traffic accidents. Moreover, biological studies [3][4] have also made use of such algorithms since the visuals belonging to a patient across time are also a form of data which gives information about the changes taking place spatially and temporally. Here, using action recognition algorithms are also beneficial; since they are able to grab and interpret the patterns, algorithms may detect unhealthy situations easily. Geographic studies [5] examining the alterations of earth’s surface in a specific area are also examples for employing action recognition since the subject data consist of spatial changes across time. Robotic systems use video processing which contains processes where the actions the robot observes should be recognized real-time so that the robot can act upon the correct instructions. As there is a wide range of applications to utilize action recognition approach, people started to come up with algorithms that are able to learn the relations between patterns and recognize actions with great accuracy.

People have been performing researches and conducting experiments in order to introduce methods that are efficient and able to yield credible results. A method should effectively reveal the patterns that carry the information about the on-going action and offer low computation time even though it utilizes complex computations. It is important to correctly classify the revealed representations of actions because high accuracy rates are demanded when the methods are used for substantial applications such as detecting abnormalities in a video or during criminal investigations.

1.2. Problem Statement

Action recognition from videos is a far-reaching and open-ended field in machine vision, which can be used in numerous applications. This thesis specifically focuses on human actions contained in short videos such as walking, drumming, and diving. The actions may last for a few seconds or can be performed during the whole video. Some actions resemble each other which complicates the recognition and results in wrong classification. Although action recognition is a specific area on its own, it has a significant role in more high-level, critical tasks such as robotics where every process is autonomous. Hence, there have been numerous studies regarding action recognition, which aim to reach a legitimate accuracy level and offer state-of-the-art results.

The studies have followed one of these approaches; they have either followed a path that includes finding out the representations of actions in the videos [6][7] and then classifying them by manifested classifiers[8], or they have introduced deep networks to merge these two steps in a generic manner as in [9][10]. The studies that were conducted in early years applied the first approach to recognize human actions since the networks then were shallow when compared to the networks used now. Also, utilizing deep networks for 3-dimensional inputs was not common. Therefore, researchers were suggesting hand-crafted representation methods which would be collaborating with classifiers. Even though these methods have provided promising results for specified tasks, they have disadvantages of not being generic. Deep networks have bridged this gap since they are not designed for detecting specific patterns and are able to accept various kinds of videos. In other words, a network used for detecting horizontal displacements can also be used for detecting vertical displacements. However, generally this is not the case for representation algorithms. Hence, it is a better choice to employ deep networks for action recognition.

Designing a network with a high recognition accuracy is challenging. Studies have followed different network architectures for this task. They have analyzed spatial and temporal components of videos with different network architectures and fused them with various techniques as in [11][12][13][14]. In addition to multi-stream approaches, a major part of the studies has tackled the videos spatio-temporally [15][16][17][18][10] by using 3-dimensional convolutional networks as the videos are spatio-temporal by their nature. Also, in [19][20] hybrid networks, where convolutional and recurrent neural networks were combined, were suggested. Throughout the years they all have reached higher recognition accuracy rates while applying different techniques to videos. Most of the studies have made a stride either by suggesting a network or modifying the existing networks. Hence, the most recent studies have shown minor improvements in accuracy rates. Even though the state-of-the-art results are legitimate, researchers still put emphasis on the importance of action recognition and make an effort to increase the recognition accuracy level by conducting experiments.

1.3. Our Approach and Contribution

In this work, instead of suggesting a network that results in increase in the recognition accuracy level by small amounts, the goal is to focus on the network input and to investigate how the input characteristics affect the recognition accuracy. The reason of analyzing input characteristics rather than the network itself is the high availability of suggested networks in previous studies. Since the state-of-the-art networks have reached a point where taking a major step forward seems challenging, it is a legitimate work to pay more attention on the characteristics of the video input.

After investigating the state-of-the-art methods, deep residual networks (ResNets) were found out to provide promising accuracy rates while they increase the learning capacity by increasing the flow of the information using its residual connection. There are also different versions of ResNets, which have different architectures in residual blocks as covered in [21]. However; since processing videos has high computational cost, the work in this thesis only focuses on ResNet architecture. Yet, the consequences are also applicable to other versions of residual networks and even to different network architectures. In other words, though the experiments were performed only on ResNet structure, the outcomes were generic since the key concern is the input characteristics.

The network and its parameters are kept constant while the used parameters were known to provide an optimal processing from previous studies. The experiments are conducted on UCF101 Action Recognition Dataset, which consists of realistic human action videos belonging to 101 classes. The experiment sets for studying the impacts of input characteristics are fivefold:

- The frame length of the input video
- The presence of color information in the video
- The normalization of pixel values
- Combined version of the input (RGB only series vs. RGB + flow masked RGB series)
- Content of the input (RGB only series vs. RGBF series vs. flow masked RGB only series)

Comparisons are performed between interrelated input types and the input characteristics set which yielded the best recognition accuracy will be offered. Effects of each modification will be analyzed separately. Moreover, the effects of modifying the input as mentioned above are different for different action classes; recognition levels for some classes increase evidently while for some classes the effects are not significant. Hence, an analysis which investigates the impacts of different input characteristics on recognition level will be performed class-wise. The accuracies will be compared with the networks that are trained from scratch in order to make a fair comparison. Lastly, the input characteristics set, yielding the highest accuracy, will be applied during fine-tuning the network on UCF101 dataset after pre-training on Kinetics dataset. By this way, the comparison with the state-of-the-art will be legitimate.

Another contribution of this thesis is that the modifications applied on input characteristics yielding high recognition accuracy gains can also be utilized by different networks for increasing the recognition accuracy. Since the modifications are not network specific, they can be described as being generic. Adopting the proposed modifications for the inputs of different networks helps them to increase their accuracy levels by keeping their network architectures as they are.

1.4. Organization of the Thesis

Chapter 1 of this thesis contains the motivation for the problem which is followed by stating the problem. The approach followed and the contribution to the problem are also provided.

The challenges in action recognition, the terms and concepts to be known are introduced in Chapter 2 as background information. The related works regarding action recognition are examined in detail. A literature survey of recent approaches and methods are also given at the end of this Chapter.

Chapter 3 presents the used network and the chosen parameters primarily. The datasets used and the evaluation parameters are covered. Then, the five experiment groups regarding different input characteristics are introduced and explained in detail.

Chapter 4 starts with providing all the results corresponding to the experiment groups and analyses for each of them. It continues with analyzing the effects of modifications on the input in a class-wise sense. Lastly, comparison with the state-of-the-art will be presented.

Possible improvements and future works are discussed in Chapter 5.

Chapter 6 concludes the thesis by emphasizing the importance of the problem and summarizing the work done.

CHAPTER 2

LITERATURE SURVEY AND RELATED WORK

2.1. Challenges in Action Recognition

Action recognition is a significant area in machine vision and can be thought of as a sophisticated version of image classification. It is the task of classifying the action happening in a video by either observing the relation between consecutive frames and matching the relation with the most relatable action, or by considering the action in the video as a whole and classifying it according to the patterns. Generally, during the classification of the actions, short snippets of the videos, namely clips, are considered. Still, algorithms are able to assign the correct actions to these short clips most of the time.

The task of action recognition has substantial challenges which affect the performance of the algorithms. First of all, it is computationally expensive when compared to image classification since images contain 1 frame of information whereas algorithms that use clips need at least 10 frames to produce a reasonable result. Moreover, apart from spatial information contained in a frame, clips also have temporal information about the motion. In order the algorithms to reveal the hidden information in clips, they need much greater number of parameters which result in high computation time. Secondly, variations in a specific type of action or resemblance between different actions complicate the process of classification. Variation in a specific type of action means that there are different ways to perform an action. For instance, "running" is an action of moving rapidly however not all "running" actions consist of fast running. If the human subject in the video runs slowly, this action should still be classified as "running". Resemblance between different actions causes an action to be classified incorrectly. For example, in short snippets of time front crawl swimming and breaststroke swimming may look similar. As Varol, G. points out in [18], “Splitting videos into short temporal intervals is likely to destroy such patterns making recognition more difficult.”. Finally, camera motion and noise in the background obstruct an action to be recognized correctly. There are pre-processing methods in order to eliminate noise. However, the viewpoint of the camera may mislead an algorithm and result in incorrect classification of an action.

2.2. Spatio-Temporal

“Spatio-temporal” is a crucial term to be clarified. Spatio-temporal data is a series of spatial data vary through time. They are dependent on both space and time. Weather patterns [22], moving-object data [23], biological data [24] are some examples of such type of data. Videos are most common expressions of spatio-temporal data since they contain 2-dimensional spatial information and 1-dimensional temporal information. Spatio-temporal analysis processes this kind of data to find out hidden patterns that relate spatial information to temporal information.

A human action is a form of 3-dimensional space-time representation. One of the most preferred techniques in action recognition from videos is spatio-temporal analysis since spatial data (frames) are collected across time [25]. An action has a spatial component that may be recognized only from a single frame and a temporal component which carries the main information of motion flow belonging to that action.

2.3. Optical Flow

Various methods do not only use RGB frame information from the videos, but also the optical flow between consecutive frames. Optical flow is the motion vector (horizontal and vertical) at each pixel between frames due to brightness variations [26]. The main pattern of the motion is presented by computing the optical flow. A widely used optical flow algorithm, Brox Optical Flow [27], is given in detail.

Brox flow is based on three assumptions, which are; Brightness Constancy Assumption, Gradient Constancy Assumption and Discontinuity-Preserving Spatio-Temporal Smoothness Constraint. It is stated that non-linearization manner was favored instead of linearization in order to grab large displacements of pixels between consecutive frames correctly. Also, coarse-to-fine warping methodology was adopted in their algorithm.

The Grey Value Constancy Assumption, in other words Brightness Constancy Assumption, is provided in (2.1) and it basically states that the gray value of a pixel at location (x, y) at time t is equal to the value of the same pixel which has a displacement of u units in x-direction and v units in y direction at time t+1.

$$I(x,y,t) = I(x+u, y+v, t+1) \tag{2.1}$$

The same relation can also be stated as in (2.2); however, it is only valid for the cases where the changes in the image are linear and in the direction of displacements. Here, subscripted I values stand for partial derivatives of I values in the specified direction.

$$I_x u + I_y v + I_t = 0 \tag{2.2}$$

However, since Brox et al. aimed to correctly compute optical flow, they used the non-linearized version of Grey Value Constancy which is provided in (2.1).

(2.1) turned out to be sensitive for small variations in gray values of pixels. In order to tolerate slight changes, they focused on spatial gradients and came up with Gradient Constancy Assumption. It is given in (2.3), where$\nabla = (\partial_x, \partial_y)^T$.

$$\nabla I(x,y,t) = \nabla I(x+u, y+v, t+1) \tag{2.3}$$

Throughout the constancy assumptions given above, the impacts of neighboring pixels were out of interest. Regarding these relations caused problems "as soon as the gradient vanishes somewhere, or if only the flow in normal direction to the gradient can be estimated". In order to favor discontinuities in flow field, a piecewise smooth flow field logic was embraced.

As the spatial displacements of pixels may be larger than one pixel between consecutive frames, the minimization algorithm could find a local minimum of the energy function, which will be described below, and be trapped there while trying to find a solution. Hence, a multi-scale approach was adopted; algorithm starts with solving a coarser (smoothed) version of input. The solution of coarser version is used as an initialization point for finer versions. By this way, global minimum is reached effectively.

Let $\bar{x} = (x, y, t)^T$be the pixel at position (x, y) at time t to be processed and $\bar{w} = (u, v, 1)^T$its displacement in x and y directions at time t+1. Then, deviations from Grey Value Constancy and Gradient Constancy Assumptions in an image can be calculated as an energy term as in (2.4).

$$E_{Data}(u, v) = \int_{\Omega}(|I(\bar{x} + \bar{w}) - I(\bar{x})|^2 + \gamma|\partial I(\bar{x} + \bar{w}) - \partial I(\bar{x})|^2)\, d\bar{x} \qquad (2.4)$$

γ is a weight parameter between two constancy assumptions. For an equation as in (2.4), outliers mislead the solution. Hence, an increasing function of $\psi(s^2) = \sqrt{s^2 + \varepsilon^2}$was introduced which is concave and ensures L1 minimization. E was fixed as 0.001. (2.4) becomes (2.5).

$$E_{Data}(u, v) = \int_{\Omega} \psi(|I(\bar{x} + \bar{w}) - I(\bar{x})|^2 + \gamma|\partial I(\bar{x} + \bar{w}) - \partial I(\bar{x})|^2) d\bar{x} \qquad (2.5)$$

For considering smoothness constraint, (2.6) was defined where $\nabla_3 = (\partial_x, \partial_y, \partial_t)^T$, the spatio-temporal gradient term.

$$E_{Smooth}(u, v) = \int_{\Omega} \psi(|\nabla_3 u|^2 + |\nabla_3 v|^2) d\bar{x} \qquad (2.6)$$

The final energy function to be minimized in order to compute the optical flow vector $\bar{w} = (u, v, 1)^T$was provided in (2.7) where α is a positive regularization term.

$$E(u, v) = E_{Data}(u, v) + \alpha E_{Smooth}(u, v) \qquad (2.7)$$

2.4. Related Work

Over time various techniques and algorithms were suggested and compete with each other in order to reach a high accuracy level of recognizing human actions in videos. There are two main approaches to process the video data and classify the partaking action. The first one is Representation and Classification Approach, which extracts the features with traditional methods and completes the task by using classifier algorithms [28]. The second one is Deep Architectures Approach, which utilizes artificial neural networks and offers more complex, sophisticated and generic methods.

2.4.1. Representation and Classification Approach

People have used combinations of representation and classification techniques in order to recognize specific actions from an input video. This approach is based on representing an action as feature vectors and obtaining the correct label by analyzing the vectors with classifiers, unlike deep networks which merge representing and classifying tasks [29]. Since this thesis focuses on deep networks for action recognition, this approach will not be examined in detail.

2.4.1.1. Representation

Representation algorithms focus on how an action is formed throughout the consecutive frames of a video and express the subject action in a form of feature vectors. By doing so, hand-crafted methods were used [30], which means the methods are task-specific or action-specific such that the parameters used in these methods are pre-defined [31]. Hand-crafted methods may result in correct representations; however, they are time consuming to optimize and come up with incorrect representations when they are exposed to different types of tasks. Especially, action recognition is a challenging task since same human actions appear in different forms, speeds and there’s also the effect of camera motion.

Representation algorithms aim to “extract representative and discriminative information of human actions, and minimize the variations” during expressing the action as feature vectors as [29] stated. They either deal with the subject of action with a holistic approach, by focusing on entire subject while being exposed to noise; or with local regions which tend to have the essential information about the action.

Motion Energy Image (MEI) and Motion History Image (MHI) [32] are examples of holistic representation approach since they focus on expressing the action information from the silhouettes in the input. They aim to find out how the motion appears and where the motion takes place. Also, Motion History Image (MHI) was extended to Motion History Volume (MHV) in [33] in order to eliminate the sensitivity to viewpoint.

Methods that follow local representation approach use the informative areas and detect the motion in space-time regions. Detection is followed by extraction of the motion features present in the corresponding areas. The pioneering idea was to focus on interest points which was suggested by the method Space-Time Interest Points (STIPs) [34]. Large motion changes were detected by Gaussian kernels [35]. Space-time extended Harris corner detector [36], Gaussian kernel, Gabor filter [37], Hessian matrix [6] are examples of techniques to find the interest points. After the detection, the motion information had to be extracted there. Optical flows and gradients are examples of such information. Histogram of Optical Flow (HOF) [7] and Histogram of Oriented Gradient (HOG) [38] were used for representing motion trajectories.

2.4.1.2. Classification

When representation phase is completed, the features are subjected to classifiers for the task to be completed. The classifiers should be trained with action classes in order to learn and distinguish the boundaries between classes.

One of the most preferred methods is to use hidden Markov models (HMM) which can be described as a sequential approach [39]. The video is processed as a sequence of frames which carry temporal information. Sequential state models are used to define actions such that specific sets of transitions between states classify actions.

Conditional Random Fields (CRF) [40] and structured Support Vector Machines (SSVM) [41] are also examples of sequential approaches.

Direct classification is also widely used in order to encode the motion information from feature vectors. Support Vector Machines (SVM) [8] and k-Nearest Neighbor (k-NN) [42] classifiers are the most common one for this approach. Also, there is bag-of-visual-words model, which is an extended version of bag-of-words model used for text data. It’s used for representing an input as a set of local visual features, such as gradients and optical flows around interest points [43]. It focuses on key points and descriptor features, in other words visual words, to define an action. These words are computed by using Fisher vector [44] or k-means clustering [45] and represented as frequency histograms. SVMs can be used to classify these words.

2.4.2. Deep Architectures Approach

Since Representation and Classification Approach mostly consists of hand-crafted methods, the usage of them requires specialties in the field and great effort. Additionally, these methods are generally task-specific or dataset-specific and do not guarantee the same performance when they are utilized in different tasks or datasets. Hence, they suffer from not being generic [46].

With prevalent usage of artificial neural networks, increased availability of computational power and expertise that researchers develop through years, deep networks are enhanced and used for such purposes. Deep architectures are said to be preferable to hand-crafted algorithms since they learn the hidden representations, automatically extract features and yield a better accuracy performance. Moreover, they are generic such that a network may be used both for classification of animals and classification of leaves. There is no need to specify the pattern of the subject like traditional methods do.

For the task of action recognition, Deep Architectures Approach has 3 main strategies. The first strategy is to separate spatial and temporal information into multi streams and then fuse them. The second one uses the 3-dimensional input as a whole in a spatio-temporal manner. The last one consists of hybrid networks which make use of CNNs and LSTMs together.

2.4.2.1. Multi-Stream Networks

2.4.2.1.1. Fusion

Karpathy et al. came up with different fusion models in [11] by asking the question "what temporal connectivity pattern in a CNN architecture is best at taking advantage of local motion information present in the video?". Throughout their work, a multi resolution CNN was used in order to decrease the runtime. The network had a context stream, which processes the whole frame in a lower resolution; and a fovea stream, which focuses on the center of the high resolution frame in order to detect the motion information. First, they used the network for single frame as reference. They suggested 3 fusion models. Early fusion takes 10 consecutive frames as input and they are fused at the beginning of the network. Late fusion takes 2 frames, which are 15 frames apart, and uses 2 separate network towers. They are fused at the end of the networks. Slow fusion is a balanced mix between Early and Late fusion models. It processes 10 consecutive frames as Early fusion model; however, they are fused with a temporal extent of 4 frames with stride 2. Hence, the 10 consecutive frames are fused progressively. Karpathy et al. described this behavior in [11] as "Higher layers get access to progressively more global information in both spatial and temporal dimensions". These models are provided in Figure 2.1. Their work showed that these fusion architectures do not have crucial effects on the performance; but Slow fusion model reaches a better accuracy level among the models.

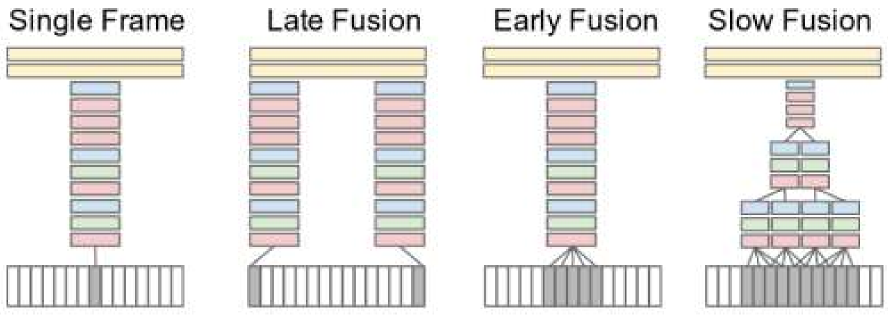

Figure 2.1. Fusion strategies suggested in [11]

2.4.2.1.2. Two-Stream Architecture

In [12] Simonyan and Zisserman suggested the most common two-stream architecture, which isolates the spatial and temporal sections of the video. The two streams are processed in two different ConvNet architectures and then fused by a late fusion model.

To do so, RGB frames (still video frames) are used for spatial stream. In order to investigate the effect of pre-training, first they used the RGB frames of UCF101 [47] dataset to train the network from scratch; then pre-trained on ImageNet ILSVRC 2012 [48] + fine-tuned on UCF-101; finally, pre-trained on ImageNet ILSVRC 2012 + only trained the last (classification) layer. They showed that pre-training has an inevitable effect on increasing the accuracy of the network.

For temporal stream, optical flows are computed by Brox flow [27]. They used optical flow stacking for 1, 5 and 10 frames and trajectory stacking for 10 frames. They conclude that optical flow stacking for 10 frames achieves the best performance and also temporal ConvNet out-performs spatial ConvNet.

Finally, they examined how to fuse the two streams; by averaging the softmax scores or by using a multi-class linear SVM [8] classifier. Their resultant network contains “a spatial net, pre-trained on ILSVRC, with the last layer trained on UCF or HMDB. The temporal net was trained on UCF and HMDB using multi-task learning, and the input was computed using uni-directional optical flow stacking with mean subtraction. The softmax scores of the two nets were combined using averaging or SVM.” as they described in [12]. The increase in the accuracy with the usage of optical flow was emphasized by the authors.

2.4.2.1.3. Two-Stream Network Fusion

Feichtenhofer took off where Simonyan and Zisserman left and investigated where to fuse the spatial and temporal networks in [13]. Instead of fusing the two streams by their softmax scores with late fusion, the convolutional layers belonging two streams can be fused at network level. Because, when the streams are fused at the end, the pixel-wise relations and patterns between the spatial streams and temporal stream cannot be learned efficiently. The fusion should be performed on the networks rather than the classification scores.

3 questions should be answered: how to fuse the networks spatially, where to fuse the networks, and how to fuse the networks temporally? For the first question; fusion by summation, maximizing, concatenation, convolution and bilinear (outer product) were performed. Sum fusion took place in the softmax layer, the others took place in RELU5 layer. The best performance was achieved by convolutional fusion; however, fusion by summation (which was commonly used) also achieved a good accuracy rate. For the second question, performing the fusion after both RELU5 and FC8 layers resulted in a high accuracy level. The first fusion creates a spatiotemporal structure by using the spatial and temporal streams. The second fusion takes place between the composed spatiotemporal stream and on-going spatial stream as shown in Figure 2.2. Not truncating the spatial stream enhanced the performance. For the last question, 3D convolution followed by 3D pooling yielded the best accuracy.

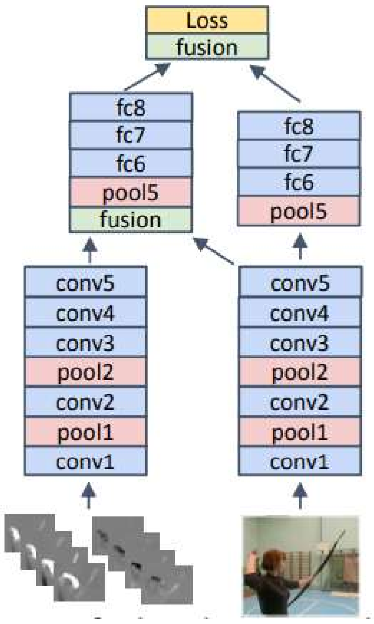

Figure 2.2. Layers where fusion is applied by [13]

2.4.2.1.4. Temporal Segment Networks

In [14] L. Wang et al. were in a search for an effective ConvNet architecture which adopts the principle of long-range temporal processing. They suggested the idea that since consecutive frames are highly redundant, they carry similar information. Hence, there is no need to sample the frames densely. Instead, sparse temporal sampling will result in better recognition of the action in the video; because sparse temporal sampling focuses on the whole video rather than consecutive frames. As seen in Figure 2.3, the video is divided into K equal parts, a single frame for spatial ConvNet and a short snippet of frames for temporal ConvNet are used randomly for each of the K parts. K many spatial ConvNets and temporal ConvNets are fed into spatial and temporal segmental consensuses. Then they are fused to form the final class scores. The network is called as Temporal Segment Networks.

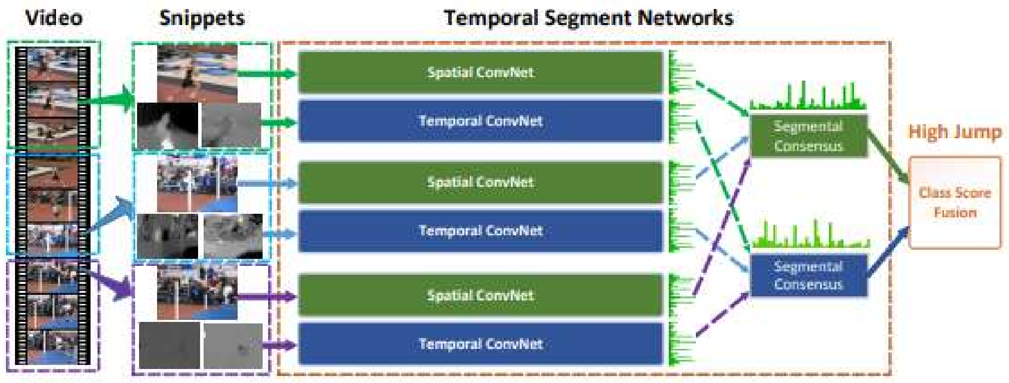

Figure 2.3. Temporal Segment Networks [14]

Moreover, they used enhanced input modalities in order to enhance the performance. In addition to RGB frames for spatial ConvNet, RGB difference frames are used. Also, optical flow frames were warped in order to compensate the camera motion. They compared the network performances when different modalities and combination of the modalities were used and reached the best accuracy level by combining optical flow, warped optical flow and RGB frames.

They also evaluated the performance of the chosen consensus function by using max pooling, average pooling and weighted average functions. For the two-stream structure, average pooling acquired the highest accuracy rate.

2.4.2.2. Spatio-Temporal Networks

2.4.2.2.1. Deep 3-Dimensional ConvNet + SVM

In [16] D. Tran et al. stated that 2D ConvNets were inadequate capturing the motion information contained in the frames; hence, 3D ConvNets would be more efficient in using the temporal relations. Their main point was using 3-dimensional convolutional kernels. 3-dimensional kernels result in volume instead of an image; so, they are able to preserve the temporal patterns during convolution. Fusion models that use 2-dimensional convolutions lose the temporal signal, resulting in lower recognition accuracies.

In order to find the appropriate kernel, they performed experiments for the time dimension of kernel, the depth. The first approach was to keep the depth constant in all the convolutional layers. Accuracy was measured for 4 networks with kernel depths 1, 3, 5, 7. Kernel depth 1 was equivalent to a 2-dimensional kernel. This work showed that using 3x3x3 had the best performance. The second experiment aimed varying temporal lengths; one network with increasing depth in layers (3-3-5-5-7) and one with decreasing depth in layers (7-5-5-3-3). Keeping the depth constant yielded better accuracy levels.

Their proposed network (C3D) was used as a feature extractor and should be combined with a classifier, such as a multi-class SVM. C3D revealed the appearance information by using the first frames and was able to focus on the motion flow in the following frames. They performed evaluation on UCF101 dataset while training in three ways; C3D trained on I380K (an internal dataset), C3D trained on Sports-1M [11], C3D trained on I380K and fine-tuned on Sports-1M. They concluded that using 3-dimensional convolutions provided a significant increase in the accuracy and fine-tuning had a positive effect on the accuracy.

2.4.2.2.2. Two-Stream Inflated 3D ConvNet

In [9] Carreira et al. proposed that current methods could be used by modifying them and utilizing them on a new, extensive dataset called Kinetics [49]. Their starting point was inflating 2-dimensional ConvNets. As it can be seen in Figure 2.4 a-c, current models until then were combination of ConvNets and LSTMs [50], 3-dimensional ConvNets that use the input spatio-temporally, two-stream architectures with ConvNets. Their two suggestions are provided in Figure 2.4 d-e; the two-stream architecture in Figure 2.4 c can be used for consecutive frames which is followed by 3D ConvNet (Figure 2.4 d), or using consecutive frames as inputs to two-stream networks and convolving them with 3D ConvNets instead of 2D ConvNets (Figure 2.4 e).

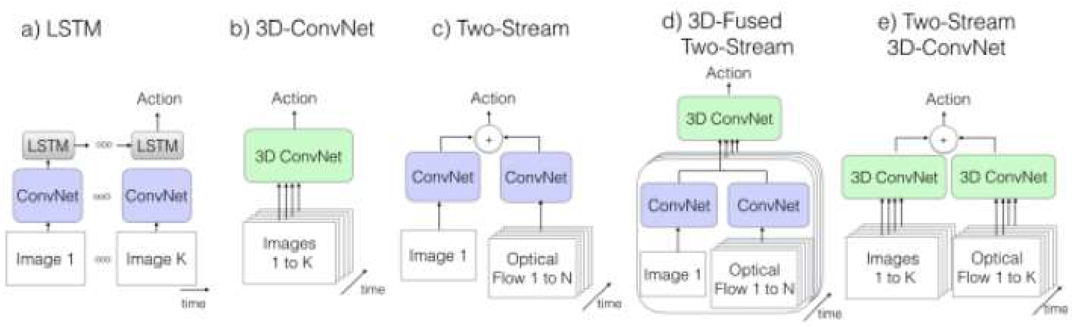

Figure 2.4. a, b, c: Network architectures suggested before Inflated 3D Networks. d, e: Network architectures using inflation [9]

In other words, the proposed network inflated the kernels and pooling of 2-dimensional state-of-the-art architectures into 3-dimensions. They showed that utilizing Inflated 3D models after pre-training them on Kinetics [49] dataset enhanced the state-of-the-art results. They made use of inception modules [51] and TV-L1 algorithm [52] for optical flow input of the temporal stream. Their work was important since they suggested an architecture by re-using existing ones.

2.4.2.2.3. Pseudo-3D

In [17] Qiu et al. had a question of "why not recycle off-the-shelf 2D networks for a 3D CNN" in consideration of the large learning capacity of Residual Networks (ResNets) [53]. The residual structure of a block in ResNets is shown in Figure 2.5 a. The idea was to create "multiple variants of bottleneck building blocks in a residual learning framework by simulating $3 \times 3 \times 3$ convolutions with $1 \times 3 \times 3$ convolutional filters on spatial domain (equivalent to 2D CNN) plus $3 \times 1 \times 1$ convolutions to construct temporal connections on adjacent feature maps in time". These variants could be used in a parallel or cascaded manner and called as Pseudo-3D blocks.

The variants were threefold: P3D-A performs 1x3x3 2D spatial convolution cascaded with 3x1x1 1D temporal convolution (Figure 2.5 b), P3D-B performs 1x3x3 2D spatial and 3x1x1 temporal convolution in parallel (Figure 2.5 c), P3D-C performs 3x1x1 1D temporal convolution in parallel with concatenation of 1x3x3 2D spatial and 3x1x1 1D temporal convolution (Figure 2.5 d).

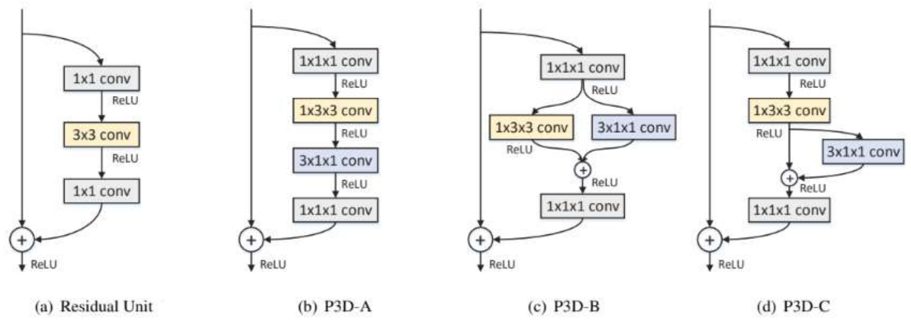

Figure 2.5. a: Residual block. b: P3D-A residual block. c: P3D-B residual block. d: P3D-C residual block. [17]

When the modified bottleneck block structures were replaced with original ones in ResNet and compared with recognition accuracy levels, it was shown that P3D-A performed best among them. Their final network consisted of the concatenation of P3D-A + P3D-B + P3D-C as a bottleneck block and this block was replaced with original ones in ResNet architecture.

2.4.2.2.4. Long-Term Temporal Convolutions

It is suggested that recent convolutional neural networks process videos for short intervals of time and this situation drops the network performance and prevents the network to recognize action correctly in [18]. Distinct human actions last several seconds as well as the repetitive ones; hence, it is not the best idea to focus on only 10 consecutive frames to label an action. Temporal characteristic patterns are embedded in long-term structures. Therefore, their suggested network is called as Long-term Temporal Convolutions (LTC).

They performed 3 studies for determining the network properties and finding out the impacts of frame length, temporal and spatial input resolutions and input modalities.

First, they compared the network performance by inputting 16 frames and 60 frames and found out that 60 frames network had better accuracy rate. Then, they varied the temporal resolution of input between 20, 40, 60, 80 and 100 frame lengths. In the meantime, they varied the spatial resolution of input between 58x58 (low resolution) and 71x71 (high resolution). Their comparison resulted in high resolution input having higher accuracy rates than low resolution input for all frame lengths. They also showed that "Networks with higher spatial resolutions give better results for lower t, however, the gain of increased spatial resolution is lower for networks with long temporal extents." [18] as shown in Figure 2.6. Finally, they used raw RGB values (3-channel) and optical flow values using Brox flow [27] (2-channel) as inputs. They compared RGB, optical flow and combination of them (by averaging). The best result was achieved by Flow+RGB as expected, which is followed by Flow and then RGB.

Their comments were as follows: as frame length increases, classification of 25 classes showed full increment whereas none of them showed full decrement. They gave an example of two actions which could be confused by networks which do not focus on long-term actions. "Gymnastics" and "Javelin Throw" both start with running, but they diverge from each other in latter frames. Using larger number of frame lengths is effective in discriminating such actions. Last point they emphasized was focusing on longer temporal extents may not affect the accuracy for short actions; however, considering short temporal extents may mislead the network for long-lasting actions and result in incorrect classification.

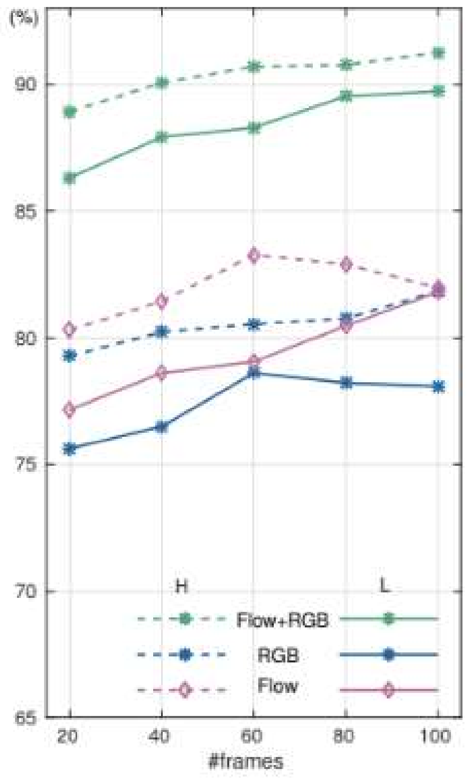

Figure 2.6. Recognition accuracy results of [18] for varying frame lengths, spatial resolution and input type

2.4.2.2.5. Deep 3-Dimensional Residual ConvNet

Tran et al. suggested in [54] the usage of Residual Networks instead of ConvNets, which they proposed before in [16], resulted in higher recognition accuracy levels. First, they listed the difficulties of action recognition task. Since videos have large amounts of data, usage of 3-dimensional ConvNets result in high computation time and inadequacy of memory. Also, there are lots of benchmarks that are used in action recognition, detection, representation tasks; hence, it is difficult to design a video classification architecture and compare with existing ones. The pre-processing of the input, sampling rates, what type of convolutions to perform, the depth of the network and modeling the temporal structure are all need to be answered. They conducted experiments to answer all these questions on UCF101 dataset. They also pointed out that UCF101 was not a large dataset and over-fitting occurred; however, single changes in the architecture could be observed easily.

They fixed the frame length to 4 and tried to find the adequate network, which is simplified 3D-ResNet18. Then the appropriate temporal stride was selected from the set {1, 2, 4, 8, 16, 32}. Here, stride 1 corresponds to consecutive 4 frames whereas stride 32 corresponds to equally scattered 4 frames spanning 128 frames. The accuracy rates showed that strides between 2 – 4 were appropriate. Then, spatial input resolutions were tried as 224x224, 112x112 and 56x56; 112x112 resolution was found to be ideal for ensuring accuracy as well as not increasing the computational complexity. Another experiment was for the type of convolutions; mixed 3D-2D convolutions (starting with 3D convolutions and continuing with 2D convolutions), 2.5D ConvNets (1xdxd convolutions and 3x1x1 convolutions) or fully 3D convolutions? As expected, using only 3D convolutions improved the accuracy more. The last parameter to choose was the depth of the ResNet. Out of depths 10, 16, 18, 26 and 34, ResNet18 achieved the highest accuracy.

2.4.2.3. Hybrid Networks

Hybrid networks make use of different deep networks by concatenating them in order to achieve a higher accuracy rate. The following hybrid networks both suggested using Recurrent Neural Networks [55], specifically Long Short Term Memory (LSTM) [50], after Convolutional Neural Networks.

2.4.2.3.1. Long-Term Recurrent Convolutional Networks

It is argued that for assignments involving processing sequences, it is better to employ deep convolutional architectures which are also recurrent. “In contrast to current models which assume a fixed spatio-temporal receptive field or simple temporal averaging for sequential processing, recurrent convolutional models are “doubly deep” in that they can be compositional in spatial and temporal “layers”.” as stated in [19]. They emphasized the positive impact of using variable length inputs for large-scale learning tasks. Their main point was making use of Recurrent Neural Networks as they are said to be “deep in time”.

The proposed network architecture inputs variable length visual sequence; after extracting the visual features by Convolutional Neural Networks, they are fed to LSTM units for sequence learning and finally variable length predictions are formed as shown in Figure 2.7. In other words, the input frames are subject to feature transformation and a sequence of visual input features are formed. In each of the LSTM structures; an output and an updated hidden state element are formed by an input and a previous hidden state element. By this way, LSTM units have control over the whole input. Finally, a distribution is predicted by using softmax function on the LSTM outputs. This process requires sequential running because of its structure.

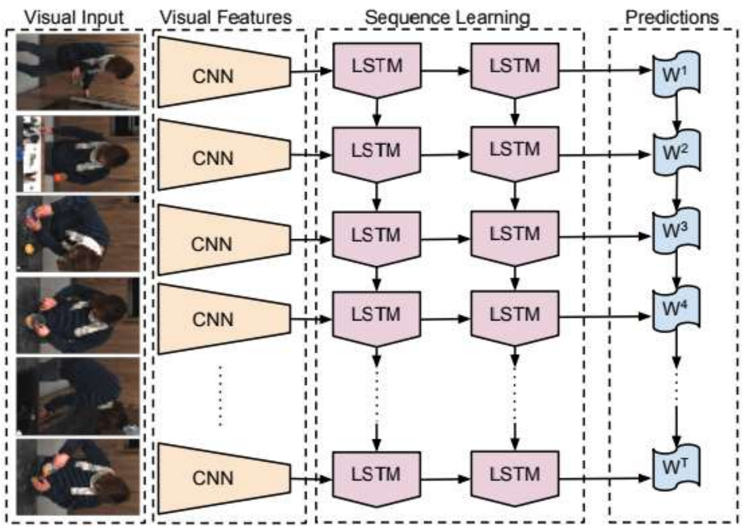

Figure 2.7. Long-Term Recurrent Convolutional Network [19]

They used this architecture for action recognition (sequential input, fixed output), image description (fixed input, sequential output), and video description (sequential input, sequential output). For action recognition, T input units, T ConvNets, a single layer of LSTMs were employed; where T was selected as 16 frames. Even though they specified the value of T, they indicated that T could be larger, and the network is able to learn complex sequences thanks to its architecture.

2.4.2.3.2. Feature Pooling vs LSTM

In [20] Ng et al. were in a search of deep networks for processing videos through larger periods of time, unlike other works until then. Previous methods, generally, processed video frames with Convolutional Neural Networks as still images and averaged the predicted values at video level. They stated that this technique resulted in not discriminating between classes because of processing incomplete information. In order to perform correct classification, the videos should be learned globally and the whole temporal relationship between frames should be focused on. Their work was in twofold: evaluating different pooling methods for temporal features and modeling the input video as a sequence of frames by using Recurrent Neural Networks.

Because of high computational complexity caused by 3-dimensional convolutional filters applied to the video, Ng et al. did not focus on implicit motion features. They also adopted a processing speed of 1 frame per second, causing the lack of implicit motion. Hence, they used explicit motion information which appeared as optical flow images. They computed optical flow for consecutive frames using TV-L1 algorithm [52].

For feature pooling, CNNs were used for each frame and the resultants were combined by various pooling strategies at frame-level instead of video-level. The attempted pooling methods are shown in Figure 2.8. They experimented to find the best state to locate the temporal pooling layer, which performs max pooling. Their experiment showed that Conv Pooling reached the highest accuracy and Time-Domain Convolution had the lowest accuracy; hence, employing a single time-domain layer had no beneficial effect on accuracy. The ideal performance was reached by processing 120 frames of video, approximately 2 minutes.

Their second suggestion was to use Recurrent Neural Networks; aiming an architecture based on LSTM layers which act as memory cells that can store the relations between frames, allowing modifications. By this way, long-range temporal correlations could be revealed. Deep Video LSTM architecture, which consists of CNNs for each frame, followed by five layers of stacked LSTMs and softmax layers for each time step, is provided in Figure 2.9.

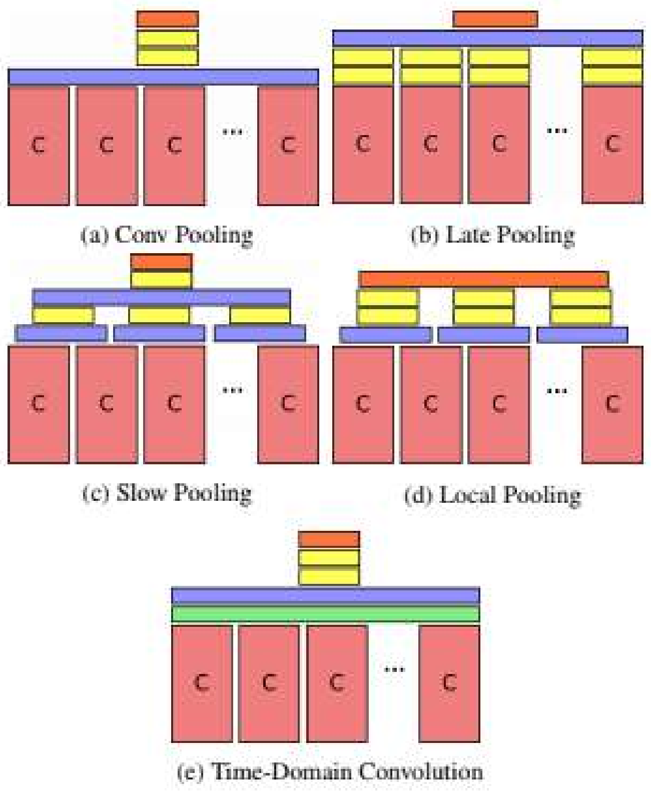

Figure 2.8. Applied pooling methods by [20]

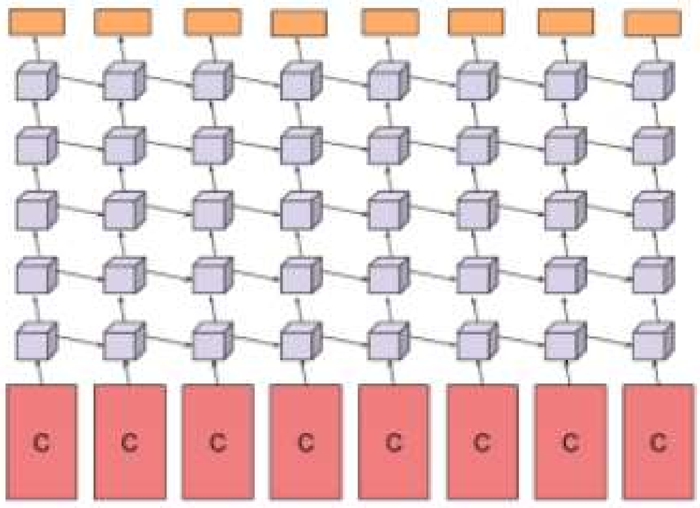

Figure 2.9. Deep Video LSTM network architecture suggested in [20]

Conv Pooling of 120 frames and LSTM with 30 frames resulted in close accuracy rates. However, their work suggested that in order to benefit the optical flow information, “it is necessary to employ a more sophisticated sequence processing architecture such as LSTM” in [20].

2.5. Different Input Modalities in the Sense of Optical Flow

Some networks use optical flow information in addition to RGB frames to maintain better operation while classifying the input videos. Networks adopting two-stream approach [11][12][14] have utilized motion flow as 2-dimensional input for each frame. For this purpose, optical flow between consecutive frames are computed which consist of optical flow values in x direction and optical flow values in y direction. They have made use of these raw flow values by setting the range to 0 to 255 in order to use the same range for RGB and flow frames.

Lately, networks are in a search of different usages of flow information. [56] observed the effects of five different flow modalities while calculating the flow using OpenCV Farneback Dense Optical Flow [57]. They used magnitude of optical flow normalized to the range (0,255), angles of directions of flow in the range of (0,180), Color Wheel representation where color represents the direction while intensity represents the magnitude of flow, direct flow vectors and scaled magnitude of optical flow. The most successful modalities were obtained by scaled magnitude and normalized magnitude modalities.

In [58], using RGB frames only, flow frames only, RGB+flow frames by fusion and RGBF frames were compared. The magnitudes of the flow were computed for every pixel and the range was set to (0,1) which was followed by masking the RGB values by the flow values resulting in RGBF frames. It was concluded that masking the RGB values with flow achieved higher accuracy rates than RGB only and flow only frames.

[59] also evaluated different forms of flow information. Two approaches using different compositions were compared; the flow information was either concatenated with RGB frames or flow information and RGB information were fused. For concatenation, three different forms were used. First one consisted magnitude of flow information in 1 channel. Second one used magnitude and direction of flow information in 2 channels. Last one utilized flow in Color Wheel format in 3 channels. Among three modalities, the magnitude of flow yielded highest accuracy level. For fusing the two streams, flow information was either represented as Color Wheel format in 3 channels or 3 layers of magnitude of flow information were used. Color Wheel resulted in highest accuracy; however, it resulted in high computation time.

These studies have shown that when flow information is used as different formats rather than raw x and y flow vectors, the operations of the networks are enhanced.

2.6. Literature Survey

As mentioned in Section 2.4.2 the suggested approaches using deep networks for action recognition are threefold: multi-stream networks, hybrid networks that benefit combining different deep architectures and networks that process spatio-temporal input as a whole.

Karpathy et al. proposed different fusion techniques (early, late and slow) in [11] and slow fusion was found out to be performing best among them. Following fusion strategy, Simonyan and Zisserman [12] suggested processing spatial and temporal components of a 3-dimensional input independently by 2-dimensional convolutional neural networks, which was pursued by fusion of these streams. They also analyzed both streams and emphasized the positive effect of pre-training the spatial networks on recognition accuracy. Then, in [13] Feichtenhofer et al. continued the fusion strategy but suggested fusing networks at an early convolutional layer level rather than fusing their classification scores at softmax layer. In addition to examining where to fuse the networks such that the highest accuracy can be reached, they also studied how to fuse the networks spatially and temporally. Moreover, they suggested multiplier networks [60] that allow appearance and motion streams to interact and spatio-temporal residual networks [61] whose residual connections were between the spatial and temporal components. These studies showed the impact of interrelations between these streams. Temporal Segment Networks were proposed by Wang et al. [14], which adopted the idea that sparse temporal sampling would supply more useful information than dense sampling since consecutive frames convey similar motion information. They also considered the effect of input modalities such as pure RGB frames, RGB difference, optical flow and warped optical flow. These studies showed the effect of fusing streams correctly and choosing appropriate input types improved recognition accuracy.

In the meantime, hybrid networks were started to be considered such that the outputs of the convolutional neural networks were inputted to recurrent neural network, specifically LSTM [50], units. Although such combined networks had high computation time, this strategy was reasonable since actions have temporal dependency between consecutive frames. LSTM units allowed the information contributing the action to be passed forward and the unnecessary information to be eliminated. In [19] Donahue et al. proposed such network as Long-Term Recurrent

Neural Network. Also, Ng et al. made use of hybrid networks while examining the most beneficial strategy for feature pooling in [20].

The last approach was using 3-dimensional convolutional kernels for spatio-temporal inputs which could be said to out-perform two-stream networks using 2-dimensional kernels. To start with, in [15] Ji et al. suggested using 3D kernels in convolutional layers were more beneficial than 2D kernels and this strategy allowed extracting spatial and temporal information which was legitimate for action recognition. Then, Tran et al. [16] introduced a 3-dimensional convolutional network, C3D, which was accepted as a reference network and study for all 3D networks proposed henceforth. Their study also compared various kernel sizes and showed that 3x3x3 kernel reached the best performance. In [18] Varol et al. accepted the success of 3D networks and focused on the length of temporal convolutions. Their key point was temporal patterns were embedded in long-term structures; hence, expanding temporal length would improve accuracy. They supported this idea by their experiments on varying the temporal length while considering the effect of spatial and temporal resolution of the input. They also showed the impact of using optical flow as input. Later, current 2D methods were re-evaluated by Carreira et al. in [9] with an approach of inflation. They basically expanded the filters and pooling kernels which were used in 2D methods into 3D. Their work was important by means of re-using the existing networks and analyzing them. A different approach, called as Pseudo-3D Residual Network (P3D), was introduced in [17] and based on re-evaluating the structure of residual blocks. P3D approach involved 3 versions of employing spatial, temporal kernels and their residuals. They finalize their work by concatenating all 3 versions to achieve the best accuracy. Following that, Hara et al. utilized 3-dimensional residual networks and found out that they were effective for action recognition tasks when trained on large-scale video datasets in [10]. Meanwhile, Tran et al. also used ResNets in [54] and performed experiments in order to choose the network structure that reached the best performance. To do so, they focused on the depth of the ResNet, the type of convolutions to perform, the stride during convolutions and the crop size of the input.

After the success of ResNets in large-scale video datasets, Hara et al. suggested observing the effect of the depth of Basic ResNet architecture and employing different structures for ResNet blocks such as Bottleneck, Pre-act, ResNeXt and DenseNet in [21]. Their work showed that the best accuracy was achieved by using ResNeXt-101.

CHAPTER 3

EXPERIMENTAL SETUP

3.1. Purpose

The focus of this thesis is investigating the effects of different input characteristics on the human action recognition accuracy rate and establishing the action classes which benefit from the modified inputs. Proposing a set of specified input characteristics which leads to an increase in the accuracy rate is the ancillary goal of the study. Also, comparison with the state-of-the-art methods is provided.

To completely focus on the results caused by the modified inputs, a network is selected for this task and its parameters are kept constant such that the alterations in recognition rate can be directly associated with the applied modifications. By this way, a controlled test environment would be guaranteed, and it would be convenient to observe the effects independently. To do so, among standing-out networks suggested in previous studies, an extended version of residual networks (ResNet) is endorsed because of its high learning capacity. The parameters such as learning rate and optimizers to be used are also kept constant. A detailed examination of ResNets and the selected parameters will be defined in Section 3.2 as well as the specifications of the training environment.

The experiments are performed on UCF101 Action Recognition Dataset which consists of videos belonging to 101 human action categories. Even though it is a small dataset, it is used as one of the standard benchmarks for action recognition tasks. In recent years, a comprehensive large scale dataset, called as Kinetics, was introduced. The latest studies have also reported the recognition results of their networks on this dataset. Kinetics dataset is more convenient than its alternatives in many aspects. As a final task of the thesis, according to the results of the_experiments performed on

UCF101 dataset, the parameters of the network are fine-tuned after pre-training on Kinetics dataset. The datasets will be examined in detail in Section 3.3.

In order to demonstrate the performance of a network, some evaluation metrics should be defined. In action recognition tasks, these metrics are specified as the loss and the accuracy of the networks during training and validation. Also, for testing the networks, top-1 and top-5 accuracies are used in this work. These metrics will be covered in Section 3.4.

Five experiment groups are formed in compliance with the modifications applied to the input videos. These modifications are made for evaluating the necessity of color information and the normalization of the input frames, the effect of using longer duration of videos, the impact of combining different types of input and the influence of the presence of optical flow information between consecutive frames. Different input characteristics yield different recognition rates as they have an inevitable effect on how the network learns and adapts its parameters. The presentation of the input to the network is as crucial as the network architecture itself. Hence, the experiment groups are formed in order to observe the impacts of different input characteristics on the operation of the network. These groups will be introduced in Section 3.5.

3.2. The Network

3.2.1. Residual Networks

3.2.1.1. 2-Dimensional Residual Network

Deep learning algorithms are based on the idea that higher representations of patterns can be reached by utilizing a hierarchy of layers. Therefore, to reach a high accuracy level, studies tend to increase the hierarchical compositions through deep networks. However, as the networks go deeper only by increasing the number of layers, some difficulties have arisen.

First of all, degradation is a crucial problem for deep networks. It means that the accuracy of the networks converges to a level and becomes saturated which is followed by a rapid degradation. The presence of degradation shows that optimizing that network and approximating its non-linear functions are not easy. The solution still consists of deeper layers; however, it is indicated in [53] that the added layers should have shortcut connection. Hence, they suggested a residual block architecture with identity mapping as shortcut connections.

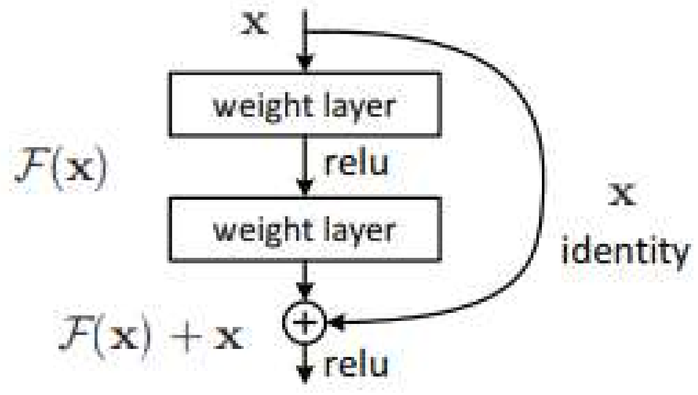
x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 3.1. Basic residual block [53]

Figure 3.1 shows the basic residual block. The input of the network is denoted as $\mathbf{x}$; 2 weight layers with RELU layer in between represent the function F($\mathbf{x}$). The weight layers are denoted by W_1 and W_2. With the identity mapping used as a shortcut connection and σ being RELU function, the output before the last RELU function is given by (3.1).

$$y = H(\boldsymbol{x}) = F(\boldsymbol{x}) + \boldsymbol{x} = F(\boldsymbol{x}, \{W_i\}) + \boldsymbol{x} = W_2\sigma(W_1\boldsymbol{x}) + \boldsymbol{x} \qquad (3.1)$$

Their hypothesis is that if complicated functions can be approximated by multiple non-linear layers in a network, then residual functions can also be approximated accordingly. H($\mathbf{x}$) is the original mapping function which is hard to optimize whereas $F(\boldsymbol{x}) := H(\boldsymbol{x}) - \boldsymbol{x}$, the residual mapping, is easier to optimize. Learning F($\mathbf{x}$) by employing residual blocks is preferred to learning F(**x**) with a simple deep network which may cause degradation. These identity connections do not add extra parameters and computational complexity nearly stays the same. Hence, networks adopting such residual blocks can be trained end-to-end and perform back-propagation.

Moreover, since each layer does not need the information only from the previous one, the information from the layers that are stacked on top of it can be propagated by utilizing residual blocks. In other words, inputs of lower layers are forwarded to deeper layers in the form of shaped information instead of abstract information. Thus, the operation of the network is favorable to increase the accuracy.

The suggested 2-dimensional Residual Network (ResNet) for 34 layers in [53] is shown in Figure 3.2. After each convolutional layer, batch normalization is used.

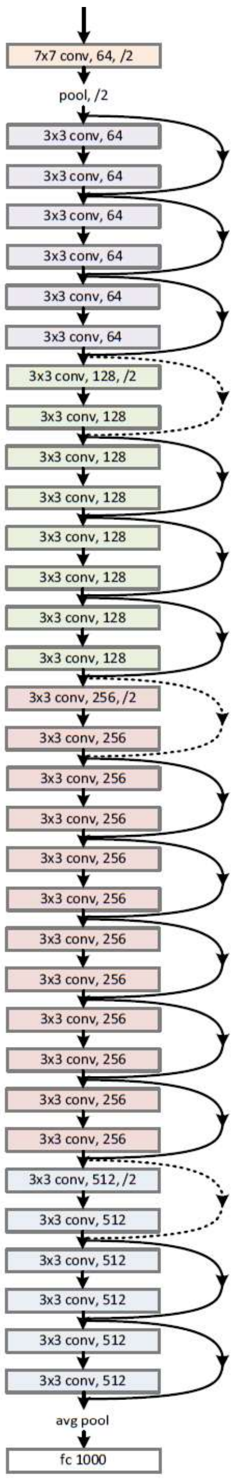

Figure 3.2. ResNet-34 network architecture [53]

3.2.1.2. 3-Dimensional Residual Network

As residual networks have shown success in image recognition tasks by learning the hidden representations effectively and reaching a high accuracy rate, their operation has gained recognition. Hence, as studied in [10] and [54], 2-dimensional residual networks used for image recognition can be extended to 3-dimensional residual networks used for action recognition. This idea has been adopted quickly since ResNets have manifested themselves and modifying them for 3-dimensional tasks has been almost effortless.

In order to utilize ResNets for 3-dimensional tasks, a third dimension is added to all kernels, pooling and convolutional layers. The advantage of using residual blocks is still valid since the residual structure of the network is preserved.

2-dimensional ResNets are modified by:

- Changing the input size from 224 x 224 to (temporal length) x 112 x 112
- Replacing the 7 x 7 convolutional kernel in conv1 layer with 7 x 7 x 7
- Performing the down-sampling in conv1 layer as 1 x 2 x 2; in other words, down-sampling the input only in spatial domain
- Performing the down-sampling in conv3, conv4, and conv5 layers as 2 x 2 x 2; in other words, down-sampling is performed both spatially and temporally
- Changing all the convolutional kernels from d x d to 3 x d x d

The resultant network architecture is provided in Table 3.1 which contains the layers, input and output sizes of 3-dimensional Residual Network for 34 layers. The dimensions are provided for a network taking 16 frames of RGB input and classifying them into 101 classes. The ‘x’ in the ‘Layer Name’ column indicates the repetition of residual blocks as indicated in the ‘Layer Details’ column. The dimensions are provided as C x T x W x H; where C is number of channels, T is temporal (frame) length, W is width and H is height.

The network in Table 3.1 is used throughout the thesis. The varied parameters are the frame length (16 in Table 3.1) and the number of input channels (3 in Table 3.1).

Table 3.1. *Network architecture for 3-dimensional residual network (34 layers)*

Layer Name	***Input Dimensions***	***Layer Details***	***Output Dimensions***
$Conv1$	$3\ x\ 16\ x\ 112\ x\ 112$	$7\ x\ 7\ x\ 7$ $stride\ 1\ x\ 2\ x\ 2$	$64\ x\ 16\ x\ 56\ x\ 56$
$Pooling$	$64\ x\ 16\ x\ 56\ x\ 56$	$maxpool\ with\ stride\ 2$	$64\ x\ 8\ x\ 28\ x\ 28$
$Conv2\_x$	$64\ x\ 8\ x\ 28\ x\ 28$	$\begin{pmatrix}3 \text{ x } 3 \text{ x } 3\\3 \text{ x } 3 \text{ x } 3\end{pmatrix} \text{ x } 3$	$64\ x\ 8\ x\ 28\ x\ 28$
$Conv3\_x$	$64\ x\ 8\ x\ 28\ x\ 28$	$\begin{pmatrix}3 \text{ x } 3 \text{ x } 3\\3 \text{ x } 3 \text{ x } 3\end{pmatrix} \text{ x } 4$	$128\ x\ 4\ x\ 14\ x\ 14$
$Conv4\_x$	$128\ x\ 4\ x\ 14\ x\ 14$	$\begin{pmatrix}3 \text{ x } 3 \text{ x } 3\\3 \text{ x } 3 \text{ x } 3\end{pmatrix} \text{ x } 6$	$256\ x\ 2\ x\ 7\ x\ 7$
$Conv5\_x$	$256\ x\ 2\ x\ 7\ x\ 7$	$\begin{pmatrix}3 \text{ x } 3 \text{ x } 3\\3 \text{ x } 3 \text{ x } 3\end{pmatrix} \text{ x } 3$	$512\ x\ 1\ x\ 4\ x\ 4$
	$512\ x\ 1\ x\ 4\ x\ 4$	$average\ pool$ $101 - d\ fc$ $softmax$	$101\ x\ 1$

3.2.2. Specifications of the Training, Validation and Test Environment

The trainings, validations and tests are performed on Ubuntu 16.04 using PyTorch and its libraries. The GPU is NVIDIA GeForce GTX1050. The networks are trained for 150 epochs and a batch size of 16 samples for all the experiment groups except for one where the frame length is set to 60 frames. Since the size is large for the GPU, the batch size is set to 8.

During training and validating the 3-dimensional ResNet, cross-entropy loss and accuracy of the output are used for measuring the performance of the network. Cross-entropy loss will be covered in Section 3.3.

As the gradients of the cross-entropy loss are back-propagated during training, Stochastic Gradient Descent (SGD) optimizer is used. SGD is useful since the parameters are updated in an iterative manner and converge in a fast fashion. SGD updates the model parameters using the gradients and the learning rate in order to minimize the loss function. For this purpose, the model's parameters are defined as follows:

- Initial learning rate is set to 0.1 and divided by 10 whenever the validation loss saturates
- Weight decay is set to 0.001 in order to prevent the weights to grow too fast
- Momentum and dampening are set to 0.9 for a proper optimization

In training process, a clip with frame length T is generated randomly from each video in every epoch. To do so, a temporal duration is selected randomly and T frames around it are inputted to the network as a clip. In other words, for each video, only one clip with length T is used. The frames are spatially cropped at one of the random positions: top left, top right, bottom left, bottom right, center. The width and height are set to 112 x 112 pixels. Frames are flipped horizontally with a probability of 50%.

During validation, again cross-entropy loss is used; however, back-propagation does not take place. Hence, optimizers are not used. In validation, 3 clips with length T are used. These clips are generated by dividing the video in 3 equal parts. The frames are spatially cropped at the center; the width and height are set to 112 x 112 pixels.

For testing the network, all frames in videos are used by dividing them into clips with frame length T. For the clips having less than T frames, the video is looped. The frames are cropped at the center and the sizes are set to 112 x 112 pixels.

3.3. Evaluation Parameters

For training the network, cross-entropy loss and accuracy are used as evaluation metrics. Cross-entropy loss is defined as in (3.2) where $\bar{x}$ is the network output before softmax, class is the ground truth and summation over j consists of all classes. The incorrect classification increases the loss in high amounts; hence, it is an effective loss calculation.

$$\begin{aligned} LOSS_{CE}(\bar{x}, class) &= -\log\left(\frac{\exp(\bar{x}[class])}{\sum_j \exp(\bar{x}[j])}\right) \\ &= -\bar{x}[class] + \log\left(\sum_j exp(\bar{x}[j])\right) \qquad (3.2) \end{aligned}$$

The accuracy is calculated as the ratio of correct classifications over batch size.

For testing the network, the scores of all classes after softmax layer belonging to clips of a video are summed, sorted descending and top 10 classes having maximum scores are assigned to that video with their scores. Then, test accuracy is calculated by checking whether the correct class label is contained in top k suggestions. In this work, top-1 and top-5 accuracies are calculated.

3.4. The Datasets

3.4.1. UCF101 Dataset

UCF101 Dataset provided in [47] has 101 human action classes as its name suggests. The videos are collected from Youtube so the videos are user uploaded. This results in unconstrained videos since the background is uncluttered and there exists camera motion. During the collection of the videos, the ones that seem irrelevant are manually detected and not used for the dataset.

There are 25 groups of clips for each action class and each group has a number of clips between 4 and 7. In total, there are 13320 videos that last for 1600 minutes. Average clip length is 7.21 seconds. All clips have a frame rate of 25 fps and a resolution of 320 x 240.

The actions can be divided into 5 categories which are; Human - Object Interaction, Body - Motion Only, Human - Human Interaction, Playing Musical Instruments, and Sports. Examples of frames belonging to 6 human action classes are provided in Figure 3.3.

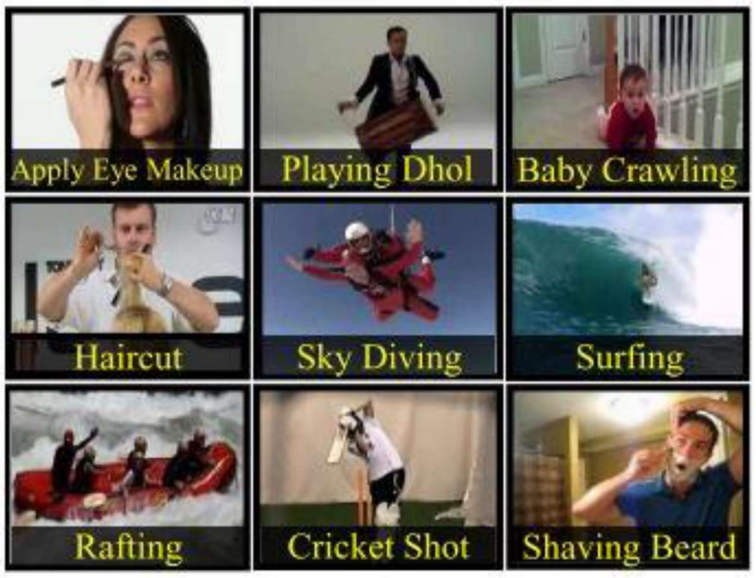

Figure 3.3. Examples for 6 action classes [47]

During training and validation, UCF101 Split 1 is used as stated in [62] in order to perform a comparison between state-of-the-art methods. Split 1 has 9537 clips for training and 3783 clips for validation.

When UCF101 Dataset was suggested, it was twice larger than the largest dataset used then. Hence, most of the studies have been conducted on UCF101 and it has become a standard benchmark for action recognition tasks. However, for 3-dimensional ResNet, UCF101 causes over-fitting when training deep networks from scratch since it is not large enough. Even though the network overfits, the alterations in the recognition accuracy rate, which are caused by modifying the input, are still observable. By this way, the alterations can be directly associated with the changes in the input.

3.4.2. Kinetics Dataset

Kinetics Dataset was suggested in [49] because of the lack of a large dataset with variations for action classes. Standard benchmarks such as UCF101 are not efficient as the networks go deeper. Kinetics set is a more challenging benchmark when compared to its alternatives and large enough to train deep networks from scratch without over-fitting.

Kinetics videos are collected from Youtube and has 400 human action classes. Each class has at least 400 clips from different videos and last 10 seconds on average. Since the videos are user uploaded, the speed of the actions, posture, camera framing, viewpoint vary. Moreover, camera shake, shadows, illumination variations exist. Also, the frame rate of the clips and the resolutions are not fixed; hence, Kinetics has a great variety in the videos.

Currently, there are 300k videos and they are divided into three parts: 250 – 1000 videos per class for training, 50 videos per class for validation and 100 videos per class for testing.

In this work, it was aimed to perform the trainings on Kinetics dataset. However, because of its large capacity and really high computation time, this idea fell through. The trainings and evaluations are performed on UCF101 as stated in 3.4.1. In [10], the trained parameters of ResNet-34 on Kinetics Dataset were shared. Hence, they are used for pre-training the network which is followed by fine-tuning on UCF101 Dataset according to the modifications made on the input as this work suggests.

3.5. Experiment Groups

Input characteristics of videos are as important as the deep network architecture itself. Input characteristics include the sample duration, color information (whether the frames have RGB values or grayscale values), the normalization of the pixels of frames, the inclusion of optical flow between consecutive frames, and the content of the clips. These modifications have various impact on the action recognition rate. In order to analyze the effects of such modifications, 5 experiment groups are specified with interrelated modifications.

3.5.1. EG 1: Frame Length

Most studies used few frames to input the deep networks. However, majority of human actions span over longer extents. By their nature, consecutive frames are related to each other and as a result, an action is defined. The temporal evolution of patterns specifies the action with the presence of the flow information. Supposing that only 10 frames are considered, an action could not be recognized completely.

Consider “running” and “long jump” actions: running is present in all of its frames; however, long jump consists of running in most of its earlier frames and the last ones carry the action of jumping. Considering few frames may mislead the network and end up in incorrect classification such that both actions may be classified as “running”.

The characteristic patterns may be lost easily if short intervals are focused on. Therefore, it is a legitimate idea to perform long-term convolutions to extract the hidden information. Longer temporal extents are expected to increase the recognition accuracy. However, it is nonsense to input the whole video since it increases computational cost much more that in increases the accuracy.

In this work, for observing the effect of sample duration, frame lengths of the input clips are varied. Three trainings sets are formed with sample durations of 16, 30 and 60 frames. During these trainings; RGB-only frames are used without normalization.

Examples of RGB frames are provided in Figure 3.4.

Figure 3.4. RGB frames for “Biking” action

3.5.2. EG 2: Color Information

The videos in UCF101 dataset have frames in color, RGB. Even though the main component that forms the actions is the optical flow between frames, the color information contained in frames contributes to the recognition of the actions.

It is obvious that using RGB frames as input gives a higher accuracy level than using grayscale frames as input. However, the critical question is that: how much difference exists between the accuracy rates when RGB input and grayscale input are used?

RGB inputs have 3 channels whereas grayscale inputs have only 1 channel. Since deep networks used in action recognition have high computational cost, the decrement in the number of input channels may lower the computation time.

The second question arises here: if the decrease in accuracy when grayscale inputs are used instead of RGB inputs is in small amounts, will it compensate the loss by decreasing the number of parameters (and computation time)? In other words, decreasing the number of parameters, hence the computation time, results in a penalty of decrease in accuracy rate by small amounts. For some applications, using grayscale inputs would be preferable. In order to analyze such modifications, the grayscale inputs used to train the network are provided in Figure 3.5. The inputs have sample duration of 16 frames.

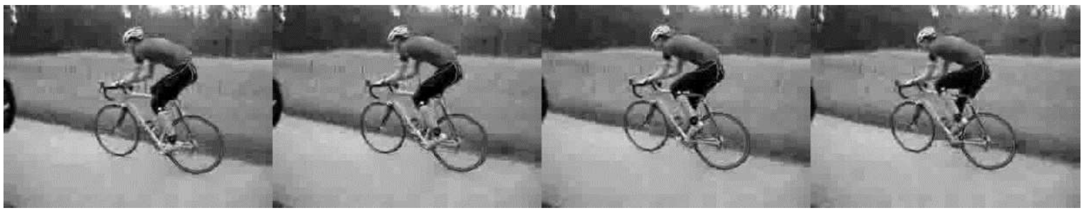

Figure 3.5. Grayscale frames for "Biking" action

3.5.3. EG 3: Normalization of the Frames

The frames in the dataset have their pixel values between 0 and 255. Some approaches suggest that normalizing pixel values result in better operation of the network and better accuracy rates. The main idea is to prevent the different behaviors of learning rate caused by multiplication with varying pixel values between 0 and 255. Normalization consists of mean extraction and division by standard deviation for each of the channels (red, green, blue).

For observing the effect of normalized inputs, three input types are used. The first one is the original, not normalized inputs which have the pixel values between 0 and 255 for each channel. For the second input type, the aim is to obtain a Gaussian distribution of pixel values. In other words, for each channel, the mean of the pixel values is calculated and extracted from the original ones to obtain a zero-mean distribution. Also, the standard deviations for each channel are calculated and the original values are divided by them to obtain a unit-variance distribution. For the last input type, the original pixel values are divided by 255; hence, their values are between 0 and 1.

The inputs have sample duration of 16 frames.

3.5.4. EG 4: Combination of Different Input Types

One of the proposed ideas in this thesis is that optical flow between consecutive frames provides the real relation between patterns that constitute the action. Hence, when the optical flow information is also presented to the network as well as the RGB information, the accuracy rate is expected to increase.

In this experiment group, the aim is to compare an input sequence that consists of RGB-only frames and an input sequence that combines RGB and "flow masked RGB" frames. "Flow masked RGB" means that the frames are still RGB; however, they are masked according to the optical flow. The following steps are followed for constituting the combined input.

1. Brox optical flow are computed as in [27]. The result consists of optical flow in x and y dimensions between consecutive frames for every pixel (F_x and F_y).
2. The magnitude of optical flow for every pixel is computed by (3.3).
$$Flow_{mag}(i,j) = 128 + 16 * \sqrt{F_x(i,j)^2 + F_y(i,j)^2} \tag{3.3}$$
3. Magnitude values are converted to mask values as in (3.4). By this way the range for mask values is set to [0, 1].
$$Flow_{mask}(i,j) = \frac{Flow_{mag}(i,j) - \min(Flow_{mag})}{\max(Flow_{mag}) - \min(Flow_{mag})} \tag{3.4}$$
4. RGB pixel values for 3 channels are multiplicated by mask values separately. Flow masked RGB sequence is obtained at this step.
5. In order to obtain combined input; RGB frames are selected for the frames with odd numbers and flow masked RGB frames are selected for the frames with even numbers.

The mask values are between 0 and 1; they take values proportional to the magnitude of the optical flow. If the optical flow value is 0 for a pixel, it indicates that there is no motion. Hence, it is reasonable to mask that pixel by multiplying zero.

As a preliminary work before evaluating an input sequence which consists of only flow masked RGB frames, a combination of RGB and flow masked RGB sequence is formed. The combined input is a transition state between RGB-only and flow masked RGB-only input types. This experiment evaluates the usage of such transition sequence as input as well as observes the effects of having an input with varying characteristics at every frame. Example frames for the combined sequence are provided in Figure 3.6. The inputs have sample duration of 30 frames.

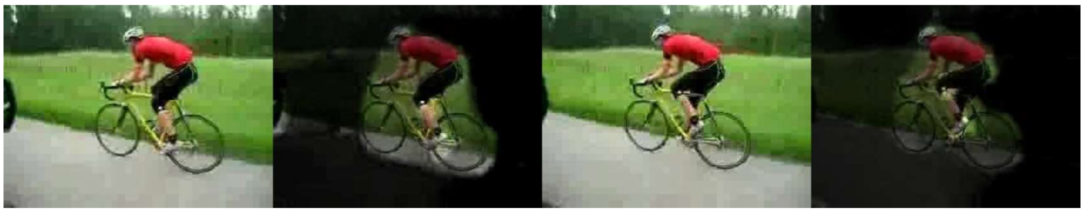

Figure 3.6. RGB and flow masked RGB combined frames for "Biking" action

3.5.5. EG 5: Content of the Input

Since optical flow information is expected to increase the recognition capacity of the network, the last group determines how to include the optical flow. There are three training sets to perform a comparison. First one inputs RGB-only frames. Second training set consists of flow masked RGB-only frames. This sequence is formed by following the steps 1 – 4 mentioned in Section 3.5.4. The third sequence contains flow information in a different form, which is called as RGBF. RGBF uses the optical flow as a fourth channel of the input. Steps 1 – 3 are followed and instead of multiplying the mask values with RGB pixels, the mask values are stored in a fourth dimension.

RGBF causes a small increase in the number of parameters. However, flow information is present to enhance the operation of the network. As it turns out, there is an image format called as RGBA where A stands for transparency parameter, alpha. Alpha being 0 means complete transparency whereas alpha being 1 means complete transparency. If a pixel has a flow value close to 0, this means there is no motion. Hence, the pixel would be transparent. This is analogous to masking the pixels with no motion by multiplying 0.

Example frames for flow masked RGB frames and RGBF frames are provided in Figure 3.7 and Figure 3.8 respectively. The inputs have sample duration of 16 frames.

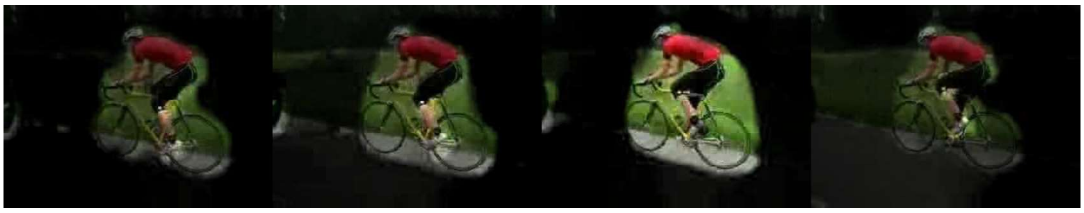

Figure 3.7. Flow masked RGB frames for “Biking” action

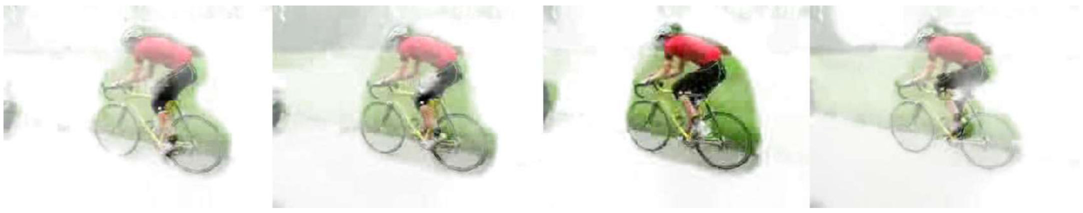

Figure 3.8. RGBF frames for “Biking” action

CHAPTER 4

RESULTS AND ANALYSIS

4.1. Results and Class-wise Analysis of Experiment Groups

This section examines the results of interrelated modifications made on input characteristics according to the experiment groups. Each modification is compared with a reference input characteristic, which is RGB-only frames without normalization. 4 of the groups use RGB-only frames for 16 frames and 1 of them uses RGB-only frames for 30 frames as reference. The alterations in validation accuracies, top-1 and top-5 values for test are considered. The effects are interpreted according to the applied modification. Then, the alterations are analyzed in a class-wise manner rather than focusing on overall results. This analysis includes:

- Number of classes with increased top-1 accuracy
- Maximum amount of increase in class-wise top-1 accuracy
- The analysis of the classes which benefit from the modification (Why are these classes affected much? Which similar action class caused misclassification?)
- Examination of classes with 0 recognition accuracy before the modifications are applied and the impact of the modifications

Last but not least, the arguments of the experiment groups are stated one by one. Finally, all results of the experiment groups are provided as a whole with their numbers of parameters and approximate computation times.

4.1.1. EG 1: Frame Length

Frame length is a crucial parameter for deep networks performing action recognition task because human actions generally last for few seconds. The patterns constituting the action should be interpreted effectively such that they provide correct classification of the action. In order to achieve higher accuracy rates, longer temporal extents should be focused on. This experiment's aim is to observe the effects of using various frame lengths of input. As a reference, 16-framed input is used. Two networks with inputs 30 and 60 frames are utilized to examine the effect. For 60 frames, the computation time is high; hence, 60-framed network is trained for 80 epochs whereas 16 and 30-framed networks are trained for 150 epochs. The graph of validation accuracies and comparison of test accuracies of three networks are provided in Figure 4.1 and Table 4.1 respectively.

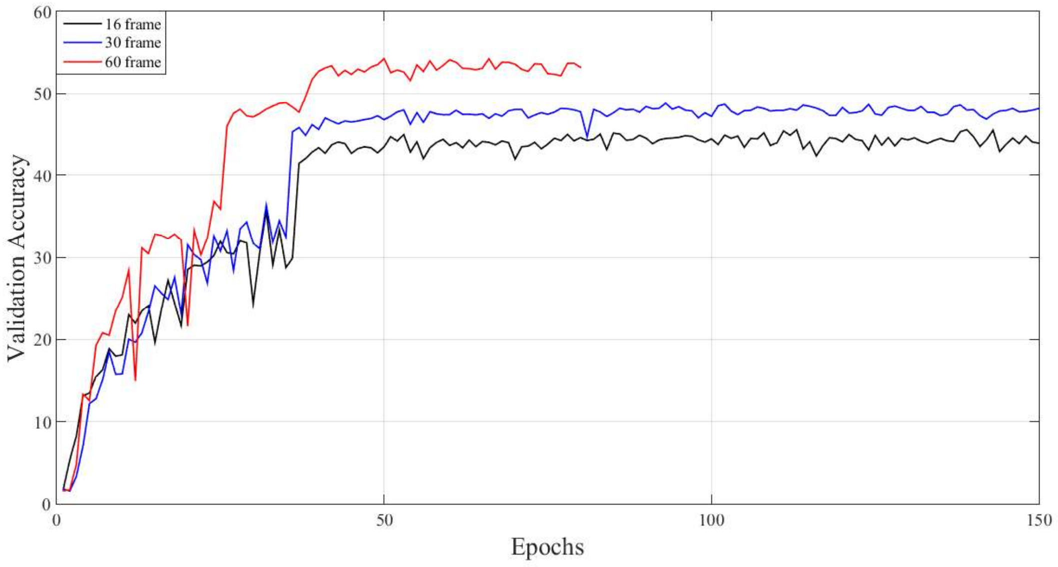

Figure 4.1. Validation accuracies for varying input frame lengths

Table 4.1. *Top-1 and top-5 test accuracies for varying input frame lengths*

Network	*Top-1 (%)*	*Top-5 (%)*
16-framed	47.58	72.51
30-framed	52.18	76.69
60-framed	**57.92**	**80.68**

Maximum recognition accuracy values reached by 16, 30, 60-framed networks are 45.57%, 48.79% and 54.25% respectively for validation. It is obvious that increasing the temporal length by doubling the input frame length results in 5% gain in accuracy. Among three networks, 60-framed network achieves the highest test accuracy of 57.92% even though it is trained for a smaller number of epochs.

The reason for the increase is that when few frames are focused on, actions may look similar. One of the discriminative properties between similar actions is the temporal duration. Longer temporal extents have an inevitable success of characterizing actions. This situation is valid for human eyes. By observing only few frames of an action, humans also misclassify because of lack of information. This deficiency may easily be overcome by increasing the present information by focusing on longer extents. Moreover, since frames are related to each other, one can benefit using larger sample durations in high amounts.

The results indicate that the frame length may be larger to gain more accuracy; however, in this work, 60 frames are selected as the maximum number of frames regarding the high computation time. The computation times will be covered at the end of this Section..

When the top-1 test accuracy rates of each class are examined, 77 and 79 of 101 classes have their accuracies increased when 30-framed and 60-framed networks are utilized. The maximum amounts of accuracy increase are 36.59% and 63.33% for 30-framed and 60-framed networks respectively. These quantitative results support focusing on longer temporal extents as most of the classes benefit from the applied modification.

By increasing the frame length from 16 to 30, 2 of the classes have amounts of increase more than 30%. These are: “FloorGymnastics” and “RockClimbingIndoor”. “FloorGymnastics” action class consists of videos where repetitive somersaults and tumbles on gymnastic mats present. When “FloorGymnastics” videos that are misclassified are examined, most of them are labeled as “BalanceBeam”; where few somersaults and tumbles are performed on a balance board. When 16 frames are used to classify the videos, it is highly possible to misclassify the actions since they are almost same for a short snippet of time. Processing the input for longer extents of time enhance the accuracy rate by eliminating such confusions. Some of the "RockClimbingIndoor" videos are confused with "CliffDiving" videos. These actions may look similar for few frames since they both consist of similar human postures; a reaching out human subject. However, the directions of the actions are the opposite; "RockClimbingIndoor" defines an upward movement whereas "CliffDiving" is downwards. Increasing frame length clarifies such classifications by carrying more information about the direction of the motion.

4 of the classes have an accuracy rate of 0 for 16-framed network; these are: "HandStandWalking", "Nunchucks", "PlayingDaf" and "SalsaSpin". Increasing the frame length to 30 does not help the network classify "HandStandWalking" and "Nunchucks", but increases the accuracies of "PlayingDaf" and "SalsaSpin" to 19.51% and 27.91% respectively. "PlayingDaf" is misclassified by "PlayingFlute" since they both include a stationary background and only the fingers of the human subject are moving. For the 16-framed network, these two actions are almost same. However, 30-framed network can distinguish the difference since "PlayingFlute" has two moving hands instead of one and generally the human subject moves in small amounts. "SalsaSpin" is misclassified by "SumoWrestling" as there are 2 humans reaching out each other with hand contacts in both classes. 30-framed network can detect the details that 16-framed network cannot; two actions are different by means of the displacements occurred and human postures. For the network, "Nunchucks" is similar to "TaiChi". It is comprehensible because both actions have standing humans behaving similarly with their hands. "HandStandWalking" gets confused with "BandMarching" and "GolfSwing". All 3 actions have human subjects with small movements. Increasing the temporal length to 30 frames is not enough for the network to interpret the human subject positioned as upside down.

The classes with accuracy gain more than 30% when 60 frames are used as input are as follows: "FloorGymnastics", "FrontCrawl", "JumpRope", "PlayingDaf", “PlayingGuitar”, “PlayingTabla”, “PushUps”, “SalsaSpin”, “ShavingBeard” and “TaiChi”. “FloorGymnastics” has no gain between 30-framed and 60-framed networks; so, 30 frames are enough to distinguish “FloorGymnastics” from “BalanceBeam”. The action which benefits most from the modification is “PushUps”. Among 30 “PushUps” videos, 5 of them are labeled as “CleanAndJerk” while the number of correct classifications is also 5 for 16-framed network. For short snippets of time two actions look similar as they include upward motion; however, using 60 frames distinguishes these actions from each other and increase the accuracy rate of “PushUps” up to 80%. Another outstanding class with accuracy gain is “FrontCrawl”. Among 37 classes, 13 of them are labeled as “FrontCrawl” whereas 15 of them are labeled as “BreastStroke”. These are two swimming styles that 16-framed network may easily confuse. Focusing on the characteristics of these styles for longer extents enhances the accuracy by great amounts. Another example is “ShavingBeard”; among 43 classes, only 4 of them are classified correctly while 15 of them are labeled as “ApplyEyeMakeup” or “BrushingTeeth”. As these 3 actions are performed on human faces, the differences may not be observed in few frames. The motion becomes evident when 60 frames are used.

For the actions having 0 accuracy rates for 16-framed network, all of them have their accuracies increased by 60-framed network. While 30 frames are inadequate for recognizing “Nunchucks”, 60-framed network increases its accuracy to 14.29% by distinguishing it from “TaiChi”. “HandStandWalking”s accuracy is increased by only 2.94%; hence, it can be said that increasing temporal length is irrelevant for recognizing this action.

The actions which have their accuracy levels increased more than 30% are provided for 30-framed and 60-framed networks in Table 4.2 and Table 4.3 respectively. The corresponding accuracy levels, the increase in the accuracy, the number of videos that each class has, the number of videos that are correctly classified and the action classes causing confusion the most are also provided.

Table 4.2. *Class-wise analysis for 30-framed network*

Class	*Top-1 (%) for 16 frames*	*Top-1 (%) for 30 frames*	*Amount of increase*	*Correctly labeled (16 frames)*	*Correctly labeled (30 frames)*	*Mostly confused with (16 frames)*
RockClimbing Indoor	46.34	82.93	**36.59**	19 / 41	34 / 41	CliffDiving
FloorGymnastics	44.44	75.00	**30.56**	16 / 36	27 / 36	BalanceBeam

Table 4.3. *Class-wise analysis for 60-framed network*

Class	*Top-1 (%) for 16 frames*	*Top-1 (%) for 60 frames*	*Amount of increase*	*Correctly labeled (16 frames)*	*Correctly labeled (60 frames)*	*Mostly confused with (16 frames)*
PushUps	16.67	80.00	**63.33**	5 / 30	24 / 30	CleanAndJerk
JumpRope	13.16	71.05	**57.89**	5 / 38	27 / 38	MoppingFloor/ VolleyballSpiking
PlayingGuitar	13.95	62.79	**48.84**	6 / 43	27 / 43	PlayingFlute
FrontCrawl	35.14	78.38	**43.24**	13 / 37	29 / 37	BreastStroke
TaiChi	21.43	60.71	**39.29**	6 / 28	17 / 28	Fencing
PlayingTabla	51.61	90.32	**38.71**	16 / 31	28 / 31	PlayingViolin
SalsaSpin	0.00	37.21	**37.21**	0 / 43	16 / 43	SumoWrestling
PlayingDaf	0.00	31.71	**31.71**	0 / 41	13 / 41	PlayingFlute
FloorGymnastics	44.44	75.00	**30.56**	16 / 36	27 / 36	BalanceBeam
ShavingBeard	9.30	39.53	**30.23**	4 / 43	17 / 43	ApplyEyeMakeup

Using 30 frames instead of 16 frames increases 77 of the classes' recognition accuracy while decreasing 24 of them. Utilizing 60 frames increases the recognition of 79 classes and decreases 22 of them. Since the number of classes with increased accuracy and their increment amounts are larger than the number of classes with decreased accuracy and their decrement amounts, the overall accuracy rate increases when longer temporal extents are focused on. Overall accuracy gains of 4.55% and 10.18% are observed when 30 and 60 frames are used respectively.

The computation times corresponding to 16, 30 and 60 framed networks are provided in Table 4.4 with frame lengths, number of epochs, number of channels and batch sizes used during training. The training computation times are obtained after all the epochs are completed. As it can be interpreted from the results, there's a linear relation between the training time and the frame length. As frame length is doubled, the training time is also doubled. For 60 frames, halving the batch size and the number of epochs cancel each other; therefore, it takes nearly two times to use 60 frames instead of 30. For the testing time, the time per sample is provided. Unlike the training times, the relation is not linear. However, as number of frames increase, the test time per sample clip also increases as expected. The computation times are not affected by the number of parameters since the number of channels are same for all networks.

Table 4.4. *Training and testing computation times for varying input frame lengths*

Network	*Frame length*	*# epochs*	*# channels*	*Training batch size*	*Training time (hours)*	*Testing time per sample (msec)*
16-framed	16	150	3	16	112.5	210.5
30-framed	30	150	3	16	225	330
60-framed	60	80	3	8	480	739.25

Argument of EG 1: Increasing frame length of the input videos enhance the human action recognition rate by great amounts since the patterns characterizing the action are interpreted more effectively when longer temporal extents are utilized. The best result is achieved by 60-framed network.

4.1.2. EG 2: Color Information

The second experiment group investigates the effect of removing the color information from the input and evaluates whether the decrease in accuracy level is negligible since grayscale input decreases the computation time by reducing the number of parameters. In other words, there is a tradeoff between a decrease in accuracy in small amounts and the number of parameters the network uses.

It is foreseeable that the recognition accuracy decreases when grayscale input is used instead of RGB because the information is reduced. However, since actions are characterized mostly by the temporal relations between frames, the foresight is that the accuracy will not be affected in great amounts if the color information is disregarded. In the meantime, since grayscale input requires 1 channel instead of 3 channels as RGB input does, the number of parameters would be less when compared to the case where RGB input is used. Less number of parameters mean less computational complexity and less computation time. Since processing videos is expensive by means of time, decreasing computation time is useful.

Two 16-framed networks having RGB and grayscale inputs are trained for 150 epochs. The computation times corresponding to RGB and grayscale networks are provided in Table 4.5 with number of parameters, number of channels and batch sizes used during training. The training computation times are obtained after all the epochs are completed. Since grayscale input uses 1 channel instead of 3, using grayscale input decreases the number of parameters by 43904 while the training computation time decreases by 17.78%. The testing time per sample is also decreased by 14.73%. It can be clearly stated that using grayscale input decreases the computation times for training and testing.

Table 4.5. *Training and testing computation times for RGB and grayscale networks*

Network	*# parameters*	*# channels*	*Training batch size*	*Training time (hours)*	*Testing time per sample (msec)*
RGB	63,565,349	3	16	112.5	210.5
Grayscale	63,521,445	1	16	92.5	179.5

Grayscale input requires 1 channel instead of 3 only at the first layer. Since it is aimed to preserve the 3-dimensional ResNet architecture throughout the work, number of channels in higher layers are kept constant. Hence, the decrease in number of parameters is limited by the first layer only. It is possible to decrease the number of parameters even more; however, this experiment group only modifies the input layer.

The graph showing the validation accuracies and comparison of test accuracies of networks are provided in Figure 4.2 and Table 4.6 respectively.

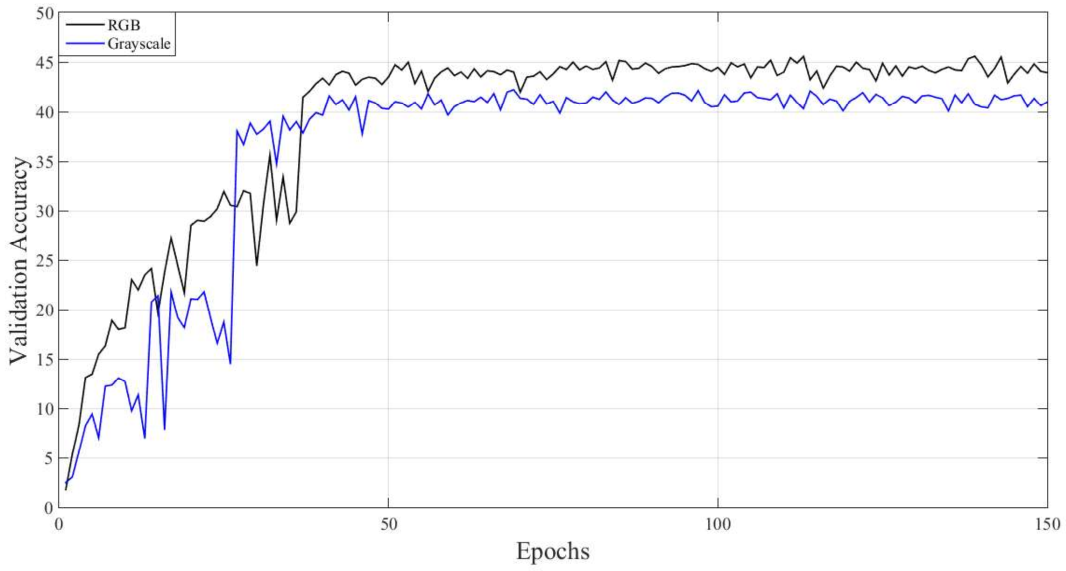

Figure 4.2. Validation accuracies for varying color information

Table 4.6. *Top-1 and top-5 test accuracies for varying color information*

Network	*Top-1 (%)*	*Top-5 (%)*
RGB	**47.58**	72.51
Grayscale	46.23	**73.33**

The maximum validation accuracy levels reached by RGB and grayscale networks are 45.57% and 42.23% respectively; so, there's a 3% difference in favor of RGB input as expected. The top-1 test accuracy also states that the network is more successful when recognizing actions from RGB input instead of grayscale input. However, for top-5 test accuracy, grayscale input achieves higher than RGB input. This result indicates that although RGB input contributes to the recognition more, accuracy rates obtained by utilizing grayscale input are not that bad since they are close.

When the overall accuracy and the computation time are evaluated together, it is preferable to use grayscale input. Yet, the main concern of the experiments is top-1 test accuracy rather than computation time. Therefore, for the rest of the experiments, RGB input is used instead of grayscale.

The class-wise analysis performed on top-1 test accuracy results indicates that 50 out of 101 action classes are more recognizable with grayscale input and the highest increase in accuracy is 37.21%. Being an unexpected success, the recognition accuracies of 5 classes increase more than 30%. These are: "ApplyEyeMakeup", "CricketShot", "SalsaSpin", "TableTennisShot", "Typing". For RGB input, "ApplyEyeMakeup" videos are mostly confused with "ApplyLipstick" videos. The recognition is increased by 34.09%. The reason for the confusion is that when the lips of the human subject in "ApplyEyeMakeup" videos appear in color, the network focuses on the lips even though the on-going action takes place in the eye area. Since network does not consider the lips as background, it classifies the video as "ApplyLipstick". Moreover, the motion flows of applying lipstick and eye make-up are similar. By eliminating the color information, the difference between the actions becomes clear to the network. 4 of the “TableTennisShot” videos are labeled as “Billiards” by the network using RGB input. Since two actions have green background as billiard table and tennis table, those actions get mistaken with each other. Lack of color in the input frames helps the network to correctly classify them. Another example is that 10 of “Typing” actions are classified as “Haircut” while only 4 of them are classified correctly. In videos of both actions, the hands are outstanding causing the network to focus on that area. When the hands become uncolored by the grayscale input, the network discriminates “Typing” action by increasing its accuracy by 37.21%.

Using grayscale input increases 3 out of 4 action classes that have 0 accuracy for RGB input. “SalsaSpin” has the highest amount of increase (30.23%) whereas “HandStandWalking” does not benefit from the grayscale input in any amounts. “Nunchucks” class has its recognition rate up to 11.43%, which is close to 60-framed network achieves in Section 4.1.1. “PlayingDaf” is also recognized and has an accuracy of 12.20%.

The actions which have their accuracy levels increased more than 30% are provided in Table 4.7 for the network using grayscale videos as. The accuracy levels for RGB and grayscale inputs, the increase in the accuracy, the number of videos that each class has, the number of videos that are correctly classified and the action classes causing confusion the most are also provided.

Table 4.7. *Class-wise analysis for grayscale input*

Class	*Top-1 (%) for RGB input*	*Top-1 (%) for gray input*	*Amount of increase*	*Correctly labeled (RGB input)*	*Correctly labeled (gray input)*	*Mostly confused with (RGB input)*
Typing	9.30	46.51	**37.21**	4 / 43	20 / 43	CleanAndJerk / Haircut
ApplyEyeMakeup	47.73	81.82	**34.09**	21 / 44	36 / 44	ApplyLipstick
TableTennisShot	51.28	82.05	**30.77**	20 / 39	32 / 39	Billiards / Hammering
CricketShot	4.08	34.69	**30.61**	2 / 49	17 / 49	SoccerPenalty
SalsaSpin	0.00	30.23	**30.23**	0 / 43	13 / 43	SumoWrestling

While 50 of 101 classes have their accuracies increased, 51 of them have their accuracies decreased. The advantages and disadvantages of grayscale input are nip and tuck as it can be interpreted from the overall accuracy since the recognition rate remains almost the same.

Argument of EG 2: Using grayscale inputs instead of RGB inputs decreases the computation time by decreasing the number of parameters that the network approximates while causing a decrease in accuracy by small amounts. As the main concern is the recognition accuracy rather than the computation time, RGB inputs are used for the rest of the experiments.

4.1.3. EG 3: Normalization of the Frames

The pixel values of the frames constituting the videos are in the range of 0 – 255 for Red, Green and Blue channels. By their nature, convolutional neural networks have their parameters as convolutional kernels which are multiplied by the pixel values of the input. Hence, standardizing the input values is a good effort to increase the accuracy. The explanation for applying such method is that when the inputs have varying ranges of values, the network behaves biased in the favor of the inputs with large values. In other words, as the convolutional kernels are multiplied with the input values, they are approximated towards the inputs with larger values. This effect may be advantageous or disadvantageous. If the network is not on the right track for inputs with wide range of values, the learning process gets complicated. Hence, the effect is more intense than expected. Therefore, normalization aims to increase the accuracy and speed up the training process by bringing the values of the input on the same scale and decreasing variance. Even though standardizing the input is useful, it comes with a risk of disregarding useful information. If the disregarded information is irrelevant for the task, it is effective to get rid of such information. However, if it is crucial for the task, then the learning process suffers from such modification.

Even though the input values (the range of the pixel values) are not different from each other in great amounts, normalization may still be useful for increasing the recognition capacity of the network. For this experiment, 3 input types are utilized. First one has no normalization for the input values; hence, the pixel values are in range of (0, 255). Second input type aims a Gaussian distribution for the pixel values. In other words, the pixel values are distributed by following a zero mean, unit variance fashion. To do so, for every channel, mean and deviation values are calculated for UCF101 dataset. These are provided in Table 4.8. The mean values are subtracted from corresponding channels and the resultant values are divided by the relevant deviation values. The new pixel value ranges are also provided in Table 4.8. For the last input type, the pixel values are divided by 255 in order to specify the range as (0, 1).

Table 4.8. *Mean and standard deviation values for every channel for the videos in UCF101, new ranges of pixel values after the normalization*

Channels	*Mean values*	*Standard deviation values*	*New ranges of pixels*
Red	90.68	59.02	-1.54, 2.78
Green	98.04	59.91	-1.63, 2.62
Blue	101.91	61.42	-1.65, 2.49

The networks are trained with the three input types for 150 epochs and 16 frames are used as frame lengths of the inputs. The graph of validation accuracies and comparison of test accuracies of three networks are provided in Figure 4.3 and Table 4.9 respectively.

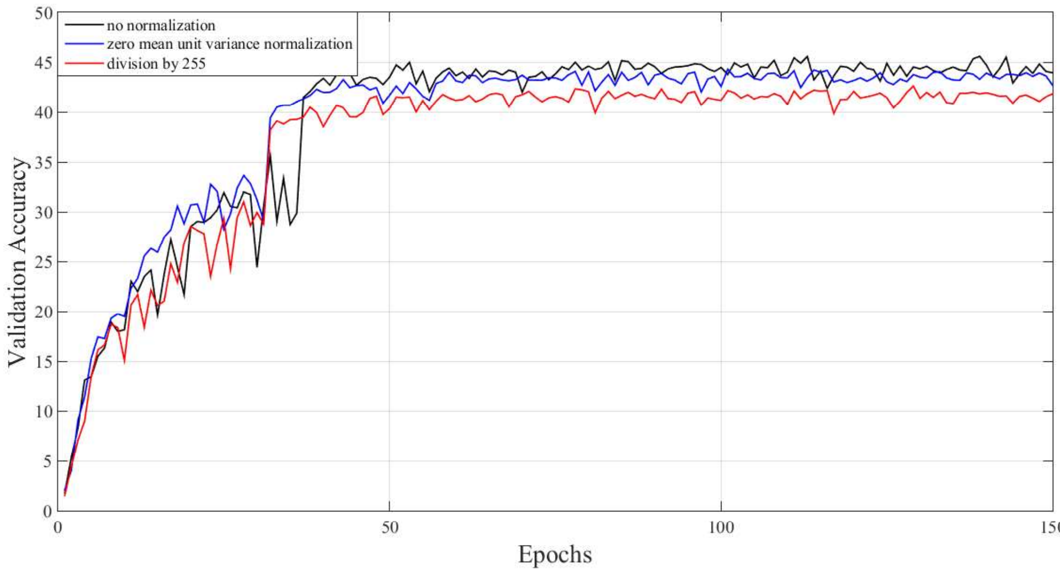

Figure 4.3. Validation accuracies for varying normalization methods

Table 4.9. *Top-1 and top-5 test accuracies for varying normalization methods*

Network	*Top-1 (%)*	*Top-5 (%)*
No normalization	**47.58**	**72.51**
Zero mean unit variance normalization	47.05	71.40
Division by 255	46.00	70.71

As it can be seen in Figure 4.3, normalizing the pixel values do not enhance the validation accuracy for neither obtaining a Gaussian distribution nor dividing the values by 255. In fact, the accuracy rate of Gaussian distributed pixels competes with not normalized inputs' while division by 255 even drops the validation accuracy rate. The reason behind these results is that the pixel values of the frames are already comparable with each other; hence, normalizing these values do not have any boosting effect on the network. Though standardizing betters the operation of the networks for most input types, it is unnecessary for such tasks that use images as inputs as it can be interpreted from the test results provided in Table 4.9. The accuracy rates are close to each other; but the highest rates are achieved without performing normalization.

The class-wise analysis shows that subtracting the mean and dividing by deviation increases the test accuracies of 56 of the 101 action classes while the maximum increase is 28.57%. Dividing the pixel values by 255 causes only 50 of 101 classes to increase the accuracies and maximum change is 23.53%. None of the classes have their accuracies increased more than 30% as it can be predicted the overall test accuracies.

For the classes with 0 recognition accuracy when not normalized inputs are used, none of the normalization techniques result in a significant increase except for "SalsaSpin" class when division by 255 method is used. Its accuracy reaches to 16.28%.

Using Gaussian normalization increases the accuracy rates of 56 classes and decreases 45 of them while division by 255 increases the accuracy rates of 50 classes and decreases 51 of them. As the class-wise analysis suggests, normalization does not contribute to the network's learning process. Therefore, the increment and decrement rates cancel each other, resulting in 0.49% and 1.63% decrease in recognition levels.

The networks are trained for 150 epochs and the training computation times are obtained after all the epochs are completed. Since the number of epochs, frame lengths, numbers of channels, batch sizes and numbers of parameters are same for the three networks, the computation times for training and testing are almost the same. These values are provided in Table 4.10.

Table 4.10. *Training and testing computation times for varying normalization methods*

Network	*Training batch size*	*Training time (hours)*	*Testing time per sample (msec)*
No normalization	16	112.5	210.5
Zero mean unit variance normalization	16	112.5	219.8
Division by 255	16	112.5	208.6

Argument of EG 3: Normalizing the pixel values of the input videos do not have any boosting effect on the recognition accuracy rates since the pixel values are already in close ranges to each other, none of the action classes benefit from the application of normalization. As standardizing the pixel values turns out to be unnecessary, normalization is not performed for the rest of the trainings.

4.1.4. EG 4: Combination of Different Input Types

Computing optical flow between consecutive input frames and making use of such information is useful for action recognition tasks since most of the information defining an action is contained in the motion flow rather than the stationary background information. Even though 3-dimensional deep networks reveal the flow between patterns during 3-dimensional convolutions, offering the optical flow to the network as input boosts the recognition accuracy.

Originated from this idea, optical flow between consecutive frames are calculated as covered in Section 3.5.4. The aim is to provide the calculated flow together with the RGB information instead of having two streams with RGB and flow information. To do so, RGB frames are masked with values altering between 0 and 1, proportional to the magnitude of optical flow in x and y directions.

Before having an input sequence with flow masked RGB-only frames, this experiment group evaluates having a combined sequence as input. By this way, both effects of utilizing a combined sequence (having RGB and flow masked RGB frames in turns) and adopting an input characteristic with alterations are observed.

The networks are trained for 150 epochs with two input sequences having 30 frames. The graph of validation accuracies and comparison of test accuracies of RGB-framed and combined networks are provided in Figure 4.4 and Table 4.11 respectively.

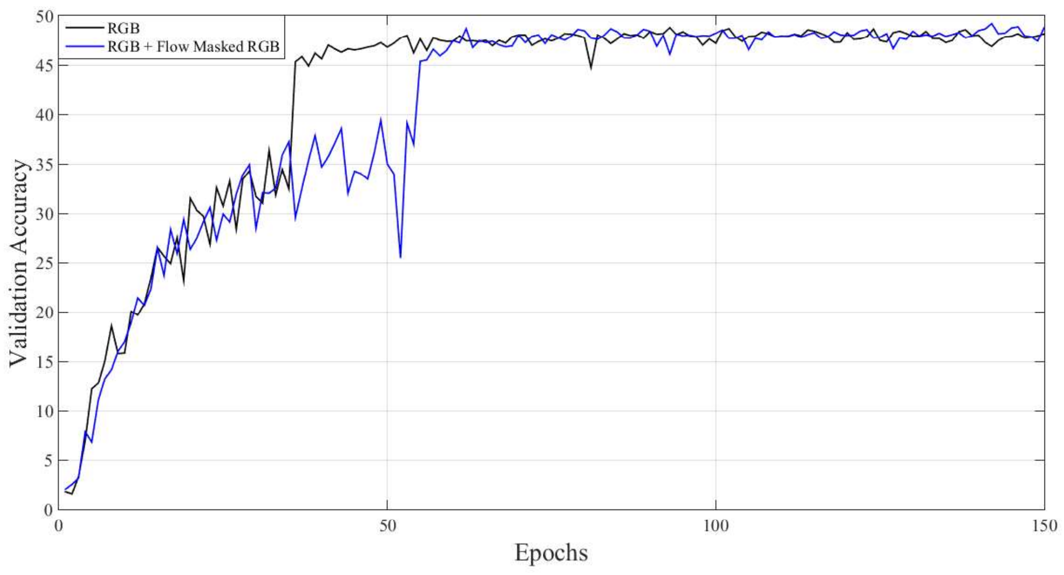

Figure 4.4. Validation accuracies for combining different input types

Table 4.11. *Top-1 and top-5 test accuracies for combining different input types*

Network	*Top-1 (%)*	*Top-5 (%)*
RGB	52.18	**76.69**
RGB + flow masked RGB	**53.24**	75.84

When the validation accuracies of two networks are examined in detail, although they seem to have very close values in Figure 4.4, the combined sequence has its maximum value 1% higher than the RGB-only sequence has. The same case is valid for the top-1 test accuracies provided in Table 4.11. However, the results are opposite for the top-5 test accuracies.

One might argue that using flow information has almost no effect on the action recognition accuracy; still, the key-point here is that the flow information is appearing in every other frame. In other words, the motion flow is present in an input frame, and disappears in the next one. It may be said that having an input characteristic having such alterations cause neither an increase nor a decrease in the resultant recognition accuracy. However, when the operation of the network is focused on, the outcomes are different.

Providing the optical flow information together with the RGB frames should enhance the accuracy rate since the most useful information about an action is contained in the flow. The results for combined sequence are actually slightly higher than the original one. In this case, the final verdict would be: even though the optical flow information present in the RGB frames attempts to increase the learning capacity of the network and the recognition accuracy, the network suffers from processing an input characteristic with high alterations such that the recognition accuracy remains almost the same.

This inference is supported by the class-wise analysis results. 64 of 101 classes have their accuracies increased but none of them have reached an increase more than 30% when combined sequence for 30 frames are used instead of RGB-only sequence for

30 frames. Most of the classes are more recognizable by the network while suffering from the penalty caused by having an altering input characteristic. The maximum increase in accuracy is 24.24%, which is not outstanding. The varying presence of the optical flow information between consecutive frames has a bad effect on the recognition rate since this situation prevents the network to make use the flow information. Therefore, utilizing such combined input have no contribution to the overall accuracy.

For the 30-framed RGB-only network, there are two classes with 0 recognition accuracy. These are "HandStandWalking" and "Nunchucks". For "HandStandWalking", the combined network has no increase in the accuracy; but it benefits the half-time present optical flow information and increases the accuracy rate of "Nunchucks" up to 22.86%.

Using an altering input with flow information present in every other frame increases the recognition of 64 classes while decreasing 37 of them. Since utilizing an altering input is not an effective way to recognize the classes accurately, the overall accuracy remains nearly the same.

The networks are trained for 150 epochs and the training computation times are obtained after all the epochs are completed. Since the number of epochs, frame lengths, numbers of channels, batch sizes and numbers of parameters are same for both networks, the computation times for training and testing are almost the same. These values are provided in Table 4.12.

Table 4.12. *Training and testing computation times for combining different input types*

Network	*Training batch size*	*Training time (hours)*	*Testing time per sample (msec)*
RGB	16	225	330
RGB + flow masked RGB	16	225	348.9

Argument of EG 4: Having a combined input of RGB-only and flow masked RGB-only frames tend to increase the recognition capacity of the network by making use of the occasionally present optical flow information while suffering from the altering input characteristic, which results in no improvement for the overall recognition accuracy. In order to benefit from the optical flow information, the flow should be present for all frames and/or in different forms.

4.1.5. EG 5: Content of the Input

As stated in Section 4.1.4, motion flow information is useful for the network when it's present for every frame in the video. Experiment group 5 aims to evaluate the effect of flow information together with RGB information on the accuracy rate. For this purpose, optical flow is adopted in two different ways. First, the calculated Brox flow between consecutive frames is added as a fourth channel to red, green and blue channels for all frames. The constructed input sequence is called as RGBF. This method modifies the RGB frames such that the pixels without motion becomes transparent whereas the pixels having the maximum amount of motion are preserved. The pixels with motion values in between gain transparency inversely proportional to the flow values. Hence, the outstanding patterns are the ones which have the highest flow. The other method to include flow information is to mask all of the RGB frames with the calculated flow values. By this way, when the magnitude of the flow value is 0, the corresponding pixel values are multiplied with 0 and appear in black. If the flow is maximum at a pixel by having a value of 1, the corresponding pixel value remains the same. All pixels are masked with a value between 0 and 1 proportional to the magnitude of the flow. Again, this method makes sure that the attention of the network is on the areas where the action takes place rather than the background with irrelevant information.

In order to perform a comparison, RGB input is used. 16 frames are used to evaluate the impact of the different contents of the input within the scope of flow information. Three networks are trained for 150 epochs. The graph of validation accuracies and comparison of test accuracies of networks using RGB, RGBF and flow masked RGB frames are provided in Figure 4.5 and Table 4.13 respectively.

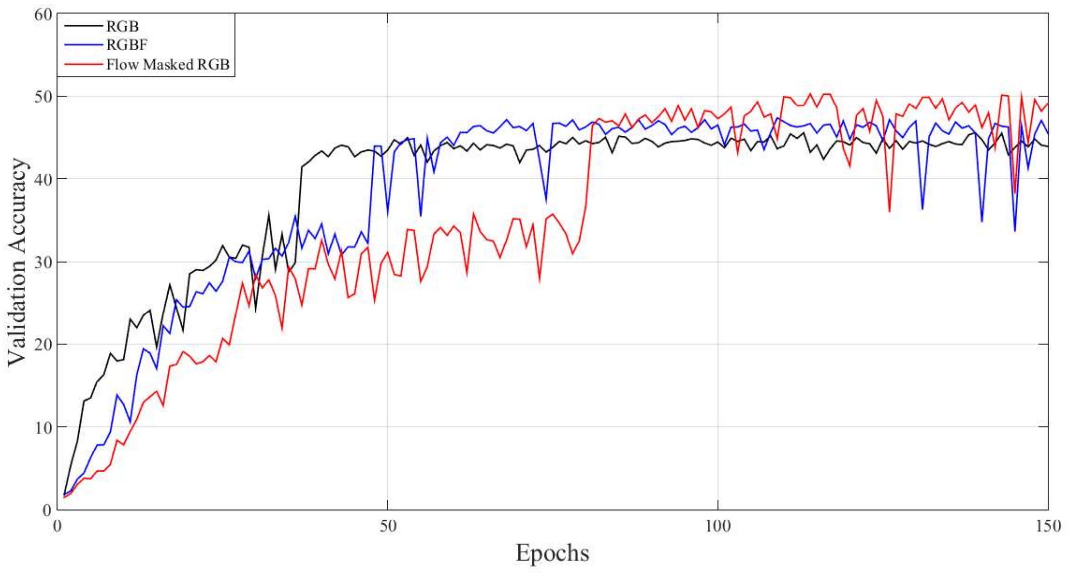

Figure 4.5. Validation accuracies for different usages of motion flow

Table 4.13. *Top-1 and top-5 test accuracies for different usages of motion flow*

Network	*Top-1 (%)*	*Top-5 (%)*
RGB	47.58	72.51
RGBF	53.26	79.04
Flow masked RGB	**59.90**	**84.01**

As it can be seen in Figure 4.5, when the flow information is presented to the network, the learning capacity and the validation accuracy levels increase. For RGB input, the validation accuracy is 45.57%, RGBF and flow masked RGB inputs reach 47.34% and 50.29% respectively. The boosting effect of flow information is more observable in the test accuracies. Including flow as RGBF format provides 5.68% accuracy gain whereas utilizing masked frames provides 12.32% accuracy gain by reaching to an overall accuracy level of 59.90%. Among all the experiment groups, flow masked RGB input results in the maximum increase of the recognition accuracy. This observation is not unexpected because actions are formed by temporal relations and performing a satisfactory recognition is possible by interpreting the flow information with the patterns present in the frames. Integrating the flow information to the RGB frames and presenting the modified frames to the network as input is helpful for the recognition task more than expected.

Even though the outstanding performance is achieved by masked inputs, the results belonging to RGBF inputs are also promising. However, as the values in Table 4.13 suggests, RGBF inputs do not enhance the usage of flow information as effective as flow masked RGB inputs do. Moreover, as using RGBF inputs increases the channels from 3 to 4, it causes increase in number of parameters and computation time. Hence, masked inputs are obviously preferable to RGBF inputs.

The class-wise analysis results are as follows: when RGBF frames are used instead of RGB frames, 60 of 101 action classes have their accuracies increased and the maximum increase is 55.81%. Masking the RGB frames increases 73 of 101 classes' accuracy levels while the highest increase is 68.29%. This is also the highest amount of increase for a class when all modifications in 5 experiment groups are considered. Moreover, the numbers of classes with accuracy gain more than 30% are 11 and 18 for the RGBF and masked inputs respectively. Again, by increasing the accuracy levels of 18 classes more than 30%, masking the RGB frames by corresponding flow values achieves the best performance among all the applied modifications.

When the effect of using RGBF frames is examined in detail for the classes with more than 30% gain in accuracy, "CricketShot" has an increase of 32.65%. Among 49 videos of that class, only 2 of them are correctly labeled by the network with RGB frames. 13 of them are confused with "SoccerPenalty". In the videos of both actions, a human subject throws a ball across a field. However, "CricketShot" requires a bat and the human subject hits the ball using that bat whereas in “SoccerPenalty” the hit is carried out by the foot of the human subject. Since the camera angle is wide, two actions resemble each other for the network. When the flow information is present, the network succeeds to label the “CricketShot” videos accurately. The highest improvement is for the “PlayingGuitar” action, which is confused with “PlayingFlute”. Both actions have two hands in the foreground and the fingers are moving. But the flow information helps the network to realize that in “PlayingGuitar” videos one hand is located at the body of the guitar with fingers moving and the other hand slides through the neck. These movements of hands differ the action from “PlayingFlute”. Another outstanding increase is for “TaiChi” with a value of 53.57%. The videos are mislabeled with mostly “Fencing”. It is understandable since both actions are examples of martial arts. While “Fencing” is based on defending oneself with swords and there’s a specified outfit, “TaiChi” appears as a form of exercise with gentle and demanding movements. The different characteristics of these actions become clear to the network when flow between consecutive frames are introduced.

All of the actions which have 0 recognition accuracy for RGB input have their accuracies increased when RGBF input is used. “PlayingDaf” has 46.34% gain and the presence of flow information helps the network distinguish the action from “PlayingFlute”. Even though the fingers are moving in both actions, the path they follow are different and this difference is realized when RGBF input is used. “Nunchucks” and “SalsaSpin” benefit the modification by 11.43% and 37.21%. “HandStandWalking” action class has an accuracy of 11.76% with the RGBF inputs. This is important because none of the modifications applied to the input characteristic yield an accuracy gain more than 2.94%. The recognition accuracy is either 0 or 2.94 for all modifications since they are all unsuccessful in characterizing “HandStandWalking”. Yet, RGBF frames are more convenient for this task. Even though the recognition rate is still low, one might argue that flow information is required in order to recognize a challenging action class such as “HandStandWalking”.

The class-wise analysis of the recognition rates for the networks with RGB and flow masked RGB frames shows that for 18 classes, the accuracies increase more than 30%. This result states that masked frames have the highest performance in enhancing the recognition capacity of the network. Introducing the flow information to the network in the form of masks applied to RGB frames is the most efficient modification for both the overall recognition and class-wise recognition. When "BenchPress" action is considered, it is mostly confused with "ParallelBars" by the network with RGB frames. Actually, these actions are different in many aspects. The only resembling features are the bars present in the videos and the action taking place indoors. However, human subject in "ParallelBars" is flipping on two parallel bars while a lying human subject is lifting bars with weights in "BenchPress". Everything about the movements of two actions is different. In order the network to perform a successful recognition, flow information is used and the accuracy is increased up to 93.75%. Another example is that "JumpRope" having a recognition gain of 55.26%. It is mostly confused with "VolleyballSpiking" by the RGB frames since they both contain jumping. Masking the RGB frames helps the network to understand the repetitive jumps in "JumpRope" are located in a specific point whereas "VolleyballSpiking" consists of a human subject who jumps once and serves the ball. "PlayingCello" is an interesting action class since almost all modified inputs which do not use motion flow cause decrease in the accuracy rate. On the contrary, masked inputs yield an accuracy gain of 59.09% which is, again, emphasizing the positive influence of making use of flow information. Another outstanding class-wise performance is for "YoYo" class. Almost none of the modifications, except masked input, increase the recognition of this class more than 2.86%. Among 35 videos, only 1 of them is correctly classified by RGB frames. The remaining 34 videos are labeled among 28 classes, which indicates that neither the modifications are successful in recognizing "YoYo" videos nor they confuse the action with another one in an obvious way. Increasing the frame length does not contribute to the recognition by any amounts while including flow information in RGBF format even decreases the recognition to 0. In order to characterize the action, flow information appearing as masks is required because of its accuracy gain of 34.29%. This result qualifies masking the frames with flow values as the best method to achieve higher recognition rates.

For the classes with 0 recognition for RGB frames; "HandStandWalking" and "SalsaSpin" achieve recognition rates of 11.76% and 37.21%, which are the same rates RGBF frames result in. This supports the idea that flow information is beneficial for recognizing such challenging actions. "Nunchucks" reaches an accuracy rate of 17.14%, which is one of the highest improvements for this class. The videos of "PlayingDaf" are recognized with a rate of 68.29%, which is the maximum amount of increase that using masked input achieves.

Classes gain accuracy more than 30% for RGBF and masked frames are provided in Table 4.14 and Table 4.15 respectively with corresponding accuracy levels, the increase in the accuracy, the number of videos that each class has, the number of videos that are correctly classified and the action classes causing confusion the most.

Table 4.14. *Class-wise analysis for RGBF input*

Class	*Top-1 (%) for RGB input*	*Top-1 (%) for RGBF input*	*Amount of increase*	*Correctly labeled (RGB input)*	*Correctly labeled (RGBF input)*	*Mostly confused with (RGB input)*
PlayingGuitar	13.95	69.77	**55.81**	6 / 43	24 / 43	PlayingFlute
TaiChi	21.43	75.00	**53.57**	6 / 28	21 / 28	Fencing
PlayingDaf	0.00	46.34	**46.34**	0 / 41	19 / 41	PlayingFlute
ShavingBeard	9.30	51.16	**41.86**	4 / 43	22 / 43	ApplyEyeMakeup
PlayingCello	18.18	56.82	**38.64**	8 / 44	25 / 44	PlayingFlute
SalsaSpin	0.00	37.21	**37.21**	0 / 43	16 / 43	SumoWrestling
PommelHorse	34.29	68.57	**34.29**	12 / 35	24 / 35	ParallelBars
CricketShot	4.08	36.73	**32.65**	2 / 49	16 / 49	SoccerPenalty

Table 4.14. (continued)

PlayingPiano	64.29	96.43	**32.14**	18 / 28	27 / 28	Drumming / PlayingFlute
PlayingDhol	4.08	34.69	**30.61**	2 / 49	17 / 49	HandStand Walking
CleanAndJerk	39.39	69.70	**30.30**	13 / 33	23 / 33	BaseballPitch / ParallelBars

Table 4.15. *Class-wise analysis for flow masked RGB input*

Class	*Top-1 (%) for RGB input*	*Top-1 (%) for masked input*	*Amount of increase*	*Correctly labeled (RGB input)*	*Correctly labeled (masked input)*	*Mostly confused with (RGB input)*
PlayingDaf	0.00	68.29	**68.29**	0 / 41	28 / 41	PlayingFlute
PlayingCello	18.18	77.27	**59.09**	8 / 44	34 / 44	PlayingFlute
PlayingGuitar	13.95	72.09	**58.14**	6 / 43	31 / 43	PlayingFlute
SoccerJuggling	30.77	87.18	**56.41**	12 / 39	34 / 39	ThrowDiscus
JumpRope	13.16	68.42	**55.26**	5 / 38	26 / 38	MoppingFloor / VolleyballSpiking
ShavingBeard	9.30	55.81	**46.51**	4 / 43	24 / 43	ApplyEyeMakeup
JumpingJack	35.14	81.08	**45.95**	13 / 37	30 / 37	HandStandPushups
PommelHorse	34.29	80.00	**45.71**	12 / 35	28 / 35	ParallelBars
PushUps	16.67	60.00	**43.33**	5 / 30	18 / 30	CleanAndJerk
TaiChi	21.43	64.29	**42.86**	6 / 28	18 / 28	Fencing
PlayingDhol	4.08	46.94	**42.86**	2 / 49	23 / 49	HandStand Walking
BodyWeightSquats	26.67	66.67	**40.00**	8 / 30	20 / 30	JumpRope / TrampolineJumping

Table 4.15. (continued)

CleanAndJerk	39.39	78.79	**39.39**	13 / 33	26 / 33	BaseballPitch / ParallelBars
SalsaSpin	0.00	37.21	**37.21**	0 / 43	16 / 43	SumoWrestling
BenchPress	58.33	93.75	**35.42**	28 / 48	45 / 48	ParallelBars
YoYo	2.86	37.14	**34.29**	1 / 35	13 / 35	JugglingBalls
JugglingBalls	30.00	62.50	**32.50**	12 / 40	25 / 40	SkyDiving
WritingOnBoard	57.78	88.89	**31.11**	26 / 45	40 / 45	CleanAndJerk / TennisSwing

RGBF input increases the accuracy rates of 60 classes while decreasing 41 of them. Since including the flow information is useful for recognizing actions, the overall accuracy is not affected much by the decreasing class accuracies and increases by 5.40%. Flow masked RGB input provides improved recognition for 73 classes while decreasing the accuracies of 28 classes. However, masking the frames according to the motion flow contributes to the operation of the network in high amounts such that the overall accuracy increases by 11.67%. The observed increase by masking the RGB values with magnitudes of motion flow is consistent with the results of [56][58][59].

The networks are trained for 150 epochs and the training computation times are obtained after all the epochs are completed. The computation times corresponding to RGB, RGBF and flow masked RGB networks are provided in Table 4.16 with number of parameters, number of channels and batch sizes used during training. As the results suggest, since the number of channels and parameters are higher for RGBF network, its training and testing times are higher than the two networks. RGB and flow masked RGB networks have approximately same computation times as expected. Because the frame lengths and number of epochs are same for all three networks, they do not affect the computation times.

Table 4.16. *Training and testing computation times for different usages of motion flow*

Network	*# parameters*	*# channels*	*Training batch size*	*Training time (hours)*	*Testing time per sample (msec)*
RGB	63,565,349	3	16	112.5	210.5
RGBF	63,587,301	4	16	122.5	319.2
Flow masked RGB	63,565,349	3	16	112.5	226.3

Argument of EG 5: Using flow information between consecutive frames together with the RGB frames results in remarkable increase in both the overall and class-wise recognition accuracies since the actions are defined mostly by the flow relations present through time rather than the stationary background information. Introducing the flow information as masks applied on RGB frames achieves better recognition rates and higher learning capacity when compared to the RGBF frame format. For recognizing challenging action classes and distinguishing similar actions from each other, flow information is required as the gain amounts acquired by utilizing masked frames are outstanding.

4.1.6. Overall Comparison of the Networks

In order to observe the effects of the modifications applied to the input characteristics of the networks, 9 different network settings are formed. Each of them aims to evaluate different kinds of modifications such as frame length of the input, or normalization of the values. The aim is to separate the modifications such that the alterations in the output of the networks can be directly associated to the applied characteristic.

To do so, 9 networks are shared by 5 experiment groups. Within each group, only one of the properties are changed to obtain a controlled environment. In other words, the properties which is irrelevant with the purpose of the experiment are kept constant. The only exception is that for network using 60 frames as input, the number of epochs and the batch size differ because of the computation time and the capacity of the training environment.

In Table 4.17, the properties for the networks and the observed values are provided all together. The networks and the corresponding experiment groups are as follows:

EG 1 → {RGB_16, RGB_30, RGB_60},

EG 2 → {RGB_16, GRAY_16},

EG 3 → {RGB_16, GAUSSIAN_16, NORM_16},

EG 4 → {RGB_30, COMBINED_30},

EG 5 → {RGB_16, RGBF_16, MASKED_16}

The network properties are given as input type, normalization, batch size, # of epochs, and sample duration. The observed values are # of parameters the networks use for learning, training times (in hours), testing times per sample (in msec), the accuracy values after training, validation and test processes.

The least number of parameters and computation time is observed for grayscale frames; however, the accuracy rates are not promising. The normalized networks have the same computation time as most of the networks using 16 frames; yet, their accuracy values are also low. Even though the approximate computation times increase going from RGB_16 to RGB_30 and RGB_60, the recognition rates also increase by 5% and 10%. Therefore, the increasing computation time acts as a penalty in exchange for accuracy gain. Combined input does not yield a significant increase in accuracy when compared to RGB_30. RGBF_16 has a remarkable amount of increase in the accuracy while the corresponding computation time is slightly higher. Among all the networks, MASKED_16 has the maximum amount of recognition gain (12%) while having the same number of parameters as the basis network, RGB_16.

Comparing all the results listed in Table 4.17, MASKED_16 achieves the highest recognition accuracy while keeping the complexity as it is.

Table 4.17. *Properties and results of all networks*

Network	*Input Type*	*Nor maliz ation*	*Batch Size*	*# Epochs*	*Sample Duration*	*# Parameters (Difference from 63,565,349)*	*Training Time (hours)*	*Testing time per sample (msec)*	*Top-1 Test Acc*	*Top-5 Test Acc*	*Max Val Acc*	*Max Train Acc*
RGB_16	RGB	NO	16	150	16	0	112.5	210.5	47.58	72.51	45.57	92.60
RGB_30	RGB	NO	16	150	30	0	225	330	52.18	76.69	48.79	86.78
RGB_60	RGB	NO	8	80	60	0	480	739.25	57.92	80.68	54.25	84.77
GRAY_16	GRAY	NO	16	150	16	- 43,904	92.5	179.5	46.23	73.33	42.23	69.38
GAUSSIAN_16	RGB	YES	16	150	16	0	112.5	219.8	47.05	71.40	44.30	87.76
NORM_16	RGB	YES	16	150	16	0	112.5	208.6	46.00	70.71	42.61	83.26
COMBINED_30	RGB+ FLOW	NO	16	150	30	0	225	348.9	53.24	75.84	49.20	96.46
RGBF_16	RGBF	NO	16	150	16	+ 21,952	122.5	319.2	53.27	79.04	47.34	83.98
MASKED_16	FLOW	NO	16	150	16	0	112.5	226.3	59.90	84.01	50.30	80.40

4.1.7. Proposed Input Characteristics Set

When the arguments of the experiment groups are evaluated, the results can be concluded as follows:

- EG 1: Increasing frame length of input results in remarkable increase in the accuracy while causing high computation times. Since the main concern is the recognition accuracy, 60 frames should be used as input.
- EG 2: The lack of color information causes a slight decrease in accuracy while lowering the computation time. As the computation time is irrelevant for this thesis, RGB frames should be used.
- EG 3: Since the pixel values of the frames are already comparable with each other, normalization has no contribution for the accuracy level. Therefore, there is no need to perform normalization.
- EG 4: Occasionally present flow information prevents the network to increase its learning capacity. The flow information should be present for all frames.
- EG 5: Both RGBF and masked frames contribute to the accuracy; however, masked frames are more successful in recognizing actions. The flow information should appear as mask values applied on the RGB frames.

As a final argument; using flow masked RGB frames while setting the sample durations to 60 frames would result in a higher recognition accuracy level. Both characteristics yield increases between 10-12%. When these modifications are applied together, they both make up the other's deficiencies and increase the learning capacity of the network.

4.2. The Effect of Pre-training

When training a network from scratch, the weights are initialized randomly. Throughout the epochs, a loss term is calculated. The aim is to minimize the loss and while doing so, the weights are optimized. By this way the errors and misclassifications are lessened because the weights are said to be learning the patterns.

When the network with trained weights is utilized for a different task or on a different dataset, instead of training the network from scratch one more time, it is preferred to use the trained weights. In other words, the weights of second training are initialized by the weights of first training. This type of initialization helps the network converge to its minimum loss value fast. Because the relations between patterns are already realized by the first network's weights. Here, the first training is called as pre-training. Moreover, during the second training, since the main features are learned, the weights of earlier layers can be freezed and optimization can be performed on only the later layers' weights. This operation on the second network is called as fine-tuning. Most studies prefer pre-training their proposed networks because it increases the accuracy in large amounts. Furthermore, small datasets benefit from pre-training on larger datasets to a great extent as the effect of over-fitting caused by the insufficient size of the dataset is eliminated by initializing weights according to pre-trained values.

The main dataset used in this work, UCF101, is a small dataset for a deep network such as ResNet. It causes over-fitting because of the insufficient numbers of samples for each class. However, it is still possible to perform experiments on such small dataset. As the results of the experiment groups are concluded, the remaining aim is to increase the accuracy even further. Therefore, the pre-trained weights on Kinetics Dataset using 3-dimensional ResNet-34 provided by [21] are used to fine-tune the networks with suggested input characteristics in this thesis. The pre-trained weights are obtained by using RGB-only input.

To observe the boosting effect of pre-training; firstly, the reference network which uses 16 RGB frames is used. The fine-tuning is performed on only the last convolutional layer and the fully-connected layer. The corresponding accuracy values are provided in Table 4.18. The top-1 test accuracy rate when the network is trained from scratch is 47.58% while the value increases up to 86.92% after fine-tuning. The increase is obvious and the advantage of using pre-trained weights is in large amounts as stated.

Table 4.18. *Test accuracy results for fine-tuned networks*

Network	*Fine-tuned layers*	*Top-1 test acc before FT (%)*	*Top-1 test acc after FT (%)*	*Top-5 test acc after FT (%)*
RGB_16	Last convolutional + fully-connected	47.58	86.92	97.78
MASKED_16	Last convolutional + fully-connected	59.90	62.12	84.69
MASKED_16	All layers	59.90	77.29	93.47
MASKED_60	All layers	-	79.01	93.52

The same fine-tuning is applied to 16 frames of flow masked RGB frames. However, the increase is not as large as it is obtained for RGB input. Because, the pre-trained weights are optimized using RGB inputs. Hence, fine-tuning only the last layers does not have a significant effect on accuracy when the input characteristic is under such change. Therefore, 16 frames of flow masked RGB inputs are fine-tuned by optimizing the weights of all layers. The increase in the recognition rate is remarkable. When the network is trained from scratch, it reaches an accuracy level of 59.90%. However, by making use of pre-trained weights, the accuracy increases up to 77.29%, which constitutes an increase of 17.39%. The gain introduced by the pre-trained weights is inevitable.

Finally, since the suggested input characteristic in this work is using 60 frames of flow masked RGB, fine-tuning all layers of such network is performed. An accuracy rate of 79.01% is achieved, which is the highest among them. As it can be interpreted from the results, pre-training increases the performance in large amounts and should be performed as almost all of the previous studies do.

For all the fine-tunings, an initial learning rate of 0.001 and a weight decay of 0.00001 are used. The networks are fine-tuned for 150 epochs except for the last one because of its high computation time. The last network is trained for 110 epochs.

4.3. Comparison with the State-of-the-art

In order to be legitimate and provide fair comparisons, two types of comparisons with the state-of-the-are results are performed. First, the accuracy results obtained from the experiment groups which result in remarkable accuracy gains are compared with the results of the networks from previous studies which are trained from scratch on UCF101 dataset. In the second comparison, the fine-tuned result of flow masked RGB input for 60 frames is compared with the state-of-the-art results.

The comparison regarding networks that are trained from scratch is provided in Table 4.19. After the modifications are applied on the input characteristics, the accuracy rates of the networks are higher than most of the methods. However, LTC network using flow as input and TSN network have higher recognition values than the ones suggested in this work. LTC consist of space-time convolutions that can interpret the patterns successfully. TSN network owes its high recognition accuracy to the idea of focusing on the whole video. Among the networks suggested in this work, using flow masked RGB frames achieves the best recognition accuracy. Its value is fairly well to the state-of-the-art results.

Table 4.19. *Comparison of state-of-the-art methods on UCF101 dataset (scratch)*

Method	*Test accuracy on UCF101*
Slow Fusion Network [11]	41.30 %
ResNet-18 [21]	42.40 %
C3D + SVM [16]	44.00 %
Res3D [54] *(baseline)*	45.90 %
Two-stream CNN [12]	56.40 %
Flow masked RGB input (16 frames)	**59.90 %**
LTC_{Flow} (16 frames) [18]	78.70 %
LTC_{Flow} (60 frames) [18]	80.50 %
TSN [14]	82.90 %

The ResNet-34 network used in this thesis is based on Res3D [54]; hence, the ResNet-34 architectures are compared with each other and the original Res3D [54] in Table 4.20. The baseline network, Res3D [54], uses 16 frames as input. The results show that increasing the frame length to 30 and 60 provides accuracy gains of 6.28% and 12%. Modifying the input by increasing the temporal extent achieves better results from the baseline network. RGBF frames achieve 7.36% higher accuracy level than the baseline network does. The network with the highest accuracy rate is the one using flow masked RGB frames which overcomes the baseline network by 14%.

Table 4.20. *Comparison with the baseline network (Res3D [54])*

Method	*Test accuracy on UCF101*
Res3D [54] (16 frames) *(baseline)*	45.90 %
RGB input (30 frames)	52.18 %
RGB input (60 frames)	57.92 %
RGBF input (16 frames)	53.26 %
Flow masked RGB input (16 frames)	**59.90 %**

Since most of the networks in the literature adopt the idea of pre-training, the comparison in Table 4.21 is necessary. The effect of pre-training is obvious since the accuracies are increased by at least 17 %. Still, the recognition results are lower when compared to the state-of-the-art methods. Especially, two-stream I3D reaches the highest performance of 98% after pre-training on both ImageNet and Kinetics. Even though the results of flow masked RGB network pre-trained on Kinetics are not as high as the state-of-the-art methods, the boosting effects of including flow information by masking the RGB frames and increasing the temporal extent to 60 frames are seen in Table 4.21. Pre-training on a large scale dataset such as Kinetics increases the accuracy as it does to other methods. However, since during pre-training the frames are RGB and during fine-tuning the frames are masked with flow values, the gain in the accuracy is not as high as other methods. If the pre-training is performed on flow masked RGB frames, the accuracy is expected to increase even more.

Table 4.21. *Comparison of state-of-the-art methods on UCF101 dataset*

Method	*Test accuracy on UCF101*
Slow Fusion Network (top 3 layers fine-tuned) [11]	65.40 %
C3D (3 nets) + linear SVM (pre-trained on I380K and fine-tuned on Sports-1M) [16]	85.20 %
Res3D (pre-trained on Sports-1M) [54]	85.80 %
ResNet-34 (pre-trained on Kinetics) [21]	87.70 %
Two-stream CNN (pre-trained on ILSVRC, SVM fusion) [12]	88.00 %
P3D ResNet (152 layers pre-trained on Sports-1M) [17]	88.60 %
ResNeXt-101 (pre-trained on Kinetics) [21]	90.70 %
LTC$_{\text{Flow+RGB}}$ (RGB pre-trained on Sports-1M) [18]	91.70 %
Asymmetric 3D-CNN (RGB+RGBF+IDT) (pre-trained on FCVID) [58]	92.60 %
TSN (BN-Inception with 3 modalities) [14]	94.20 %
Two-stream I3D (pre-trained on ImageNet and Kinetics) [9]	98.00 %
ResNet-34 architectures in this thesis	
Flow masked RGB input (16 frames, pre-trained on Kinetics with RGB inputs)	77.29 %
Flow masked RGB input (60 frames, pre-trained on Kinetics with RGB inputs)	79.01 %

4.4. Application on SqueezeNet

The results of the experiment groups show that using 60 frames instead of 16 and masking the RGB frames with the magnitude of motion flow between consecutive frames boost the recognition performance of 3-dimensional Residual Networks. They contribute to the learning process of the network by guiding the network to focus on the areas where the actual motion takes place. Moreover, the actions are characterized in a more efficient way because of the longer temporal extents. In order to emphasize the beneficial work of such input characteristics, the same logic is applied on a different network.

SqueezeNet [63] is a model which has less parameters than AlexNet has while achieving the same performance on images. In [64], 3-dimensional version of SqueezeNet, 3D-SqueezeNet, is implemented such that it can be used on videos in a spatio-temporal manner.

In order to examine the effects of using an input characteristic as proposed in this thesis, two networks with same parameters are trained. The first network is trained with RGB frames with a sample duration of 16 and the second network is trained with flow masked RGB frames with a sample duration of 60. Both networks are initialized with a learning rate of 0.1 and trained for 80 epochs. The top-1 test accuracy for the network with 16 RGB frames is 53.56% while top-1 test accuracy for the network with 60 flow masked RGB frames is 61.86%. As it can be interpreted from these accuracy rates, utilizing masked frames and increasing the temporal extent have significant improvement on the operation of the network.

CHAPTER 5

FUTURE WORK

The aim of this thesis is to observe the effects of modification applied on the input characteristics of the 3-dimensional Residual Network on the recognition accuracy level for the task of human action recognition. As the results of the experiment groups suggest, there are two main ideas that result in high amounts of increase in the accuracy. One of them is increasing the temporal extent of the input whereas the other one adopts using the flow information between consecutive frames as mask values to RGB frames. These modification yield accuracy gain. Furthermore, by pre-training the network on a large scale dataset, the accuracy is increased even more.

There are some possible future works in order to contribute the recognition further. During the pre-training and fine-tuning processes, different input characteristics are used. Pre-training the network with RGB inputs on Kinetics dataset and fine-tuning the weights with flow masked RGB inputs on UCF101 dataset prevent the process to reach its full potential. Although the accuracy increases by great amounts, it may reach higher values when same input characteristics are used. Because of the high computation time of the network trained on Kinetics, the training is not performed in this work. Instead, the trained weights shared by [21] are used. By this way, the time is spent for focusing on observing the effects of different input characteristics. Hence, as a possible improvement, the network should be trained on Kinetics dataset using 60 frames of flow masked RGB inputs, and then fine-tuned on UCF101 dataset.

The experiments and observations are performed on UCF101 dataset using 3-dimensional ResNets. These settings are adequate to evaluate the characteristics. However, the arguments of the experiments are applicable for different networks, different datasets, and even different tasks. As a future work, the ideas adopted in this thesis may be applied to the networks suggested in previous studies. The networks reaching state-of-the-art results may even increase their recognition accuracy rates when they utilize the suggested input characteristics in this thesis. Therefore, higher accuracy levels may be obtained when the modifications are applied to input frames while using state-of-the-art methods.

CHAPTER 6

CONCLUSION

Action recognition is a far-reaching and challenging task which is used in various fields from collecting statistics to crime detection. Throughout the years, action recognition has taken place in daily lives, technological studies and mostly become the target of machine learning researches. In previous studies, countless algorithms were suggested in order to recognize patterns to interpret the on-going actions. They have aimed to represent the hidden relations defining the actions and classify the representations using classifiers. However, these methods were hand-crafted; therefore, these kinds of algorithms have shown success within certain limits. Also, these algorithms were not generic since they were optimized for specific kinds of data. Therefore, studies were in search for more generic, fast computing and effective methods. After the outbreak of deep neural networks, they have become widely-used for action recognition tasks. Their success has been remarkable since they have required less amounts of effort while reached higher levels of accuracy.

Deep networks are preferred because of their high learning capacities and interpretation of hidden relations between patterns in an implicit way. Over time, there are different kinds of networks suggested for reaching the state-of-the-art results such as two-stream networks using spatial and temporal information separately, 3-dimensional convolutional neural networks which process the input in a spatio-temporal manner. As computational power improves and networks go deeper, higher levels of recognition are achieved. At this point, there are a great number of networks with remarkable performances such that suggesting a new network increases the state-of-the-art results in small amounts. Therefore, this thesis aims to explore different input characteristics yielding accuracy gain instead of suggesting a different network architecture with a slight increase in accuracy.

For this purpose, modifications regarding different characteristics are applied on inputs and the effects are observed within experiment groups. While doing so, 3-dimensional residual networks are used because of their increased learning capacity. With the experiment groups, the following modifications are studied: different frame lengths (16, 30, 60), presence of color information (RGB, grayscale), different normalization methods (no normalization, zero mean unit variance normalization, division by 255 normalization), different combinations of inputs (RGB-only, RGB and flow masked RGB combination), different contents of inputs (RGB, RGBF, flow masked RGB). All modifications are evaluated separately and analyzed in a class-wise sense. The input characteristics yielding significant accuracy increase are studied in detail by investigating which kinds of actions benefit the applied modification. Also, there were some classes with 0 recognition accuracy. During class-wise analysis, these are also examined.

The results of the experiments for observing the effects on top-1 test accuracy levels are as follows. As the frame lengths increase, the network's capacity of interpreting the patterns also increases. Even though the computation times are higher, the accuracy gains are remarkable such that 5% and 10% for 30 and 60-framed networks when compared to 16-framed network. The main reason of such increase is that actions are formed by patterns which are highly related to each other temporally. Focusing on longer temporal extents contributes the network to reveal the actions more effectively. Using grayscale frames instead of RGB results in a slight decrease in recognition level while decreasing the computation time. However, the main concern of the thesis is the recognition performance rather than the computation time, it is unnecessary to use grayscale frames. When it comes to normalization, the best result is achieved without normalization. Because the frame values of the videos are already in the same scale (0 – 255), normalization does not contribute to the operation of the network. Combining RGB frames with flow masked RGB frames is expected to increase the accuracy because the flow information is introduced. However, this is not the case because varying the presence of flow information complicates the learning process. Occasionally present motion flow prevents the network to reach high accuracy levels by making use of the flow information. This experiment concludes with the idea that either the flow information should be present in all frames or it should appear in a different form. Therefore, the last experiment introduces the flow as a fourth dimension to RGB frames (RGBF). This kind of input results in 6% increase in accuracy. This amount is fairly well; still, it is not sufficient. As a last modification, RGB values of each frame in videos are masked with optical flow values between consecutive frames. This modification achieves the highest accuracy gain. The increase is 12%, resulting in a recognition level of 59.90%. Since optical flow carries the main information used to describe the actions and disregard irrelevant information, masking the RGB pixel values with magnitudes of flow values is an effective method for action recognition. Making use of flow masked RGB frames increases the network's capacity of learning challenging classes. Moreover, the classes with 0 recognition accuracy benefit the masked inputs since they have the most accuracy gain with the presence of motion flow. As a final argument regarding all the experiments performed for investigating the input characteristics, using flow masked RGB inputs and setting the frame length to 60 result in the highest accuracy.

As the studies in the literature improve their networks' accuracy by adopting the idea of pre-training the networks on large-scale datasets, the same concept is applied. However, since training the network on such dataset causes really high computation time, the pre-trained values on ResNet-34 which are provided by [21] are used. These values are obtained for RGB input. When the pre-trained values are fine-tuned for flow masked RGB input using 16 frames, the accuracy rate increases from 59.90% to 77.29%. Moreover, if the fine-tuning is applied for flow masked RGB input using 60 frames, the recognition reaches to 79.01%. Because the optimization of the weights is initialized from learned parameters instead of random ones, the accuracies are increased in great amounts. The networks benefit from pre-training and reach higher levels of recognition.

When the networks with modified inputs, which are trained from scratch, are compared with state-of-the-art methods, they reach accuracy levels of almost 60%. This result shows improvements among most of the methods while remaining lower than some. The comparison of the pre-trained networks shows that using modified inputs for 3-dimensional ResNet-34 during fine-tuning reaches higher recognition levels while not performing as well as the state-of-the-art results. Although the accuracy increases in great amounts, the full learning capacity may not be reached since the frames used for pre-training are not masked. As a future work, the pre-training may be performed for flow masked RGB inputs, which is expected to result in higher accuracy levels.

After evaluating different input characteristics, this thesis proposes that increasing the frame lengths while masking the RGB frames with motion flow values contributes to the recognition of human actions in great amounts. Because actions are described by temporal relations, interpreting the patterns obtained from longer temporal extents and focusing on the segments which contribute the action mostly result in increased recognition capacity. As a final improvement, when the weights are initialized from pre-trained values, the recognition accuracy increases ever more.

REFERENCES

[1] D. Kumar, “A survey of activity recognition and understanding the behaviour in video surveillance,” *Vis. Comput.*, vol. 29, no. 10, pp. 983–1009, 2013.

[2] O. P. Popoola and K. Wang, “Video-Based Abnormal Human Behavior Recognition—A Review,” *IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev.*, vol. 42, no. 6, pp. 865–878, Nov. 2012.

[3] C. Lea, G. D. Hager, and R. Vidal, “An Improved Model for Segmentation and Recognition of Fine-Grained Activities with Application to Surgical Training Tasks,” in *2015 IEEE Winter Conference on Applications of Computer Vision*, 2015, pp. 1123–1129.

[4] H.-C. Shin, M. R. Orton, D. J. Collins, S. J. Doran, and M. O. Leach, “Stacked Autoencoders for Unsupervised Feature Learning and Multiple Organ Detection in a Pilot Study Using 4D Patient Data,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1930–1943, Aug. 2013.

[5] Y. Ren, W. K. G. Seah, and P. D. Teal, “Performance of pressure routing in drifting 3D underwater sensor networks for deep water monitoring,” in *Proceedings of the Seventh ACM International Conference on Underwater Networks and Systems - WUWNet ’12*, 2012, p. 1.

[6] G. Willems, T. Tuytelaars, and L. V. Gool, “An efficient dense and scale-invariant spatio-temporal interest point detector,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5303 LNCS, no. PART 2, 2008, pp. 650–663.

[7] A. Klaeser, M. Marszalek, and C. Schmid, “A Spatio-Temporal Descriptor Based on 3D-Gradients,” p. 99.1-99.10, 2012.

[8] C. Schuldt, I. Laptev, and B. Caputo, “Recognizing human actions: a local SVM approach,” in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, 2004, p. 32–36 Vol.3.

[9] J. Carreira and A. Zisserman, “Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4724–4733.

[10] K. Hara, H. Kataoka, and Y. Satoh, “Learning spatio-Temporal features with 3D residual networks for action recognition,” in *Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017*, 2018, vol. 2018–Janua, pp. 3154–3160.

[11] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-Scale Video Classification with Convolutional Neural Networks,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[12] K. Simonyan and A. Zisserman, “Two-Stream Convolutional Networks for Action Recognition in Videos,” pp. 1–9, Jun. 2014.

[13] C. Feichtenhofer, A. Pinz, and A. Zisserman, “Convolutional Two-Stream Network Fusion for Video Action Recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, no. i, pp. 1933–1941.

[14] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool, “Temporal segment networks: Towards good practices for deep action recognition,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9912 LNCS, pp. 20–36, 2016.

[15] S. Ji, W. Xu, M. Yang, and K. Yu, “3D Convolutional Neural Networks for Human Action Recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[16] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning Spatiotemporal Features with 3D Convolutional Networks,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4489–4497.

[17] Z. Qiu, T. Yao, and T. Mei, “Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 5534–5542.

[18] G. Varol, I. Laptev, and C. Schmid, “Long-Term Temporal Convolutions for Action Recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1510–1517, Jun. 2018.

[19] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell, “Long-Term Recurrent Convolutional Networks for Visual Recognition and Description,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 677–691, Apr. 2017.

[20] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, “Beyond short snippets: Deep networks for video classification,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4694–4702.

[21] K. Hara, H. Kataoka, and Y. Satoh, “Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6546–6555.

[22] S. P. Charles, B. C. Bates, and J. P. Hughes, “A spatiotemporal model for downscaling precipitation occurrence and amounts,” *J. Geophys. Res. Atmos.*,

vol. 104, no. D24, pp. 31657–31669, Dec. 1999.

[23] M. Vazirgiannis and O. Wolfson, “A Spatiotemporal Model and Language for Moving Objects on Road Networks,” pp. 20–35, 2001.

[24] D. Yue, X. Xu, Z. Li, C. Hui, W. Li, H. Yang, and J. Ge, “Spatiotemporal analysis of ecological footprint and biological capacity of Gansu, China 1991–2015: Down from the environmental cliff,” *Ecol. Econ.*, vol. 58, no. 2, pp. 393–406, Jun. 2006.

[25] G. Atluri, A. Karpatne, and V. Kumar, “Spatio-Temporal Data Mining: A Survey of Problems and Methods,” vol. 1, no. 1, pp. 1–37, Nov. 2017.

[26] B. K. P. Horn and B. G. Schunck, “Determining optical flow,” *Artif. Intell.*, vol. 17, no. 1–3, pp. 185–203, 1981.

[27] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, “High Accuracy Optical Flow Estimation Based on a Theory for Warping,” vol. 3024, 2004, pp. 25–36.

[28] H. B. Zhang, Y. X. Zhang, B. Zhong, Q. Lei, L. Yang, J. X. Du, and D. S. Chen, “A comprehensive survey of vision-based human action recognition methods,” *Sensors (Switzerland)*, vol. 19, no. 5, pp. 1–20, 2019.

[29] Y. Kong and Y. Fu, “Human Action Recognition and Prediction: A Survey,” vol. 13, no. 9, Jun. 2018.

[30] H. Bay, T. Tuytelaars, and L. V. Gool, “SURF: Speeded up robust features,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3951 LNCS, pp. 404–417, 2006.

[31] Z. Lan, S. Yu, M. Lin, B. Raj, and A. G. Hauptmann, “Handcrafted Local Features are Convolutional Neural Networks,” no. 2005, pp. 1–11, Nov. 2015.

[32] A. F. Bobick and J. W. Davis, “The recognition of human movement using temporal templates,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 3, pp. 257–267, Mar. 2001.

[33] D. Weinland, R. Ronfard, and E. Boyer, “Free viewpoint action recognition using motion history volumes,” *Comput. Vis. Image Underst.*, vol. 104, no. 2–3 SPEC. ISS., pp. 249–257, Nov. 2006.

[34] I. Laptev, “On Space-Time Interest Points,” *Int. J. Comput. Vis.*, vol. 64, no. 2–3, pp. 107–123, Sep. 2005.

[35] W. Wang, Z. Xu, W. Lu, and X. Zhang, “Determination of the spread parameter in the Gaussian kernel for classification and regression,” *Neurocomputing*, vol. 55, no. 3–4, pp. 643–663, 2003.

[36] I. Laptev and T. Lindeberg, “Space-time interest points,” in *Proceedings Ninth IEEE International Conference on Computer Vision*, 2003, pp. 432–439 vol.1.

[37] M. Bregonzio, S. Gong, and T. Xiang, “Recognising action as clouds of space-time interest points,” in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, 2009, vol. 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1948–1955.

[38] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, “Learning realistic human actions from movies,” in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[39] T. V. Duong, H. H. Bui, D. Q. Phung, and S. Venkatesh, “Activity recognition and abnormality detection with the switching hidden semi-Markov model,” in *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, 2005, vol. I, no. Cvpr, pp. 838–845.

[40] B. W. Sy, A. Quattoni, L. P. Morency, D. Demirdjian, and T. Darrell, “Hidden conditional random fields for gesture recognition,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, vol. 2, pp. 1521–1527.

[41] Q. Shi, L. Cheng, L. Wang, and A. Smola, “Human action segmentation and recognition using discriminative semi-Markov models,” *Int. J. Comput. Vis.*, vol. 93, no. 1, pp. 22–32, 2011.

[42] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, “Actions as space-time shapes,” in *Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1*, 2005, vol. 29, no. 12, p. 1395–1402 Vol. 2.

[43] H. Riemenschneider, M. Donoser, and H. Bischof, “Bag of Optical Flow Volumes for Image Sequence Recognition,” in *Procedings of the British Machine Vision Conference 2009*, 2009, p. 28.1-28.11.

[44] F. Perronnin and C. Dance, “Fisher Kernels on Visual Vocabularies for Image Categorization,” in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[45] A. Kovashka and K. Grauman, “Learning a hierarchy of discriminative space-time neighborhood features for human action recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 2046–2053, 2010.

[46] L. Marchesotti, F. Perronnin, D. Larlus, and G. Csurka, “Assessing the aesthetic quality of photographs using generic image descriptors,” *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 1784–1791, 2011.

[47] K. Soomro, A. R. Zamir, and M. Shah, “UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild,” no. November, 2012.

[48] “Large scale visual recognition challenge 2010.” [Online]. Available: www.imagenet.org/challenges.

[49] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The Kinetics Human Action Video Dataset," 2017.

[50] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[51] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *Journal. Pract.*, vol. 10, no. 6, pp. 730–743, Feb. 2015.

[52] C. Zach, T. Pock, and H. Bischof, "A Duality Based Approach for Realtime TV-L 1 Optical Flow," in *Pattern Recognition*, vol. 1, no. 1, Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 214–223.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[54] D. Tran, J. Ray, Z. Shou, S. Chang, and M. Paluri, "ConvNet Architecture Search for Spatiotemporal Feature Learning," no. section 3, Aug. 2017.

[55] J. Elman, "Finding structure in time," *Cogn. Sci.*, vol. 14, no. 2, pp. 179–211, Jun. 1990.

[56] P. Viswanath, G. Sistu, M. Ilie, S. Yogamani, and J. Horgan, "Early fusion of Dense Optical Flow with image for semantic segmentation in autonomous driving," *CEUR Workshop Proc.*, vol. 2259, pp. 126–137, 2018.

[57] G. Farneback, "Two-Frame Motion Estimation Based on," *Lect. Notes Comput. Sci.*, vol. 2749, no. 1, pp. 363–370, 2003.

[58] H. Yang, C. Yuan, B. Li, Y. Du, J. Xing, W. Hu, and S. J. Maybank, "Asymmetric 3D Convolutional Neural Networks for action recognition," *Pattern Recognit.*, vol. 85, pp. 1–12, 2019.

[59] H. Rashed, S. Yogamani, A. El-Sallab, P. Krizek, and M. El-Helw, "Optical Flow augmented Semantic Segmentation networks for Automated Driving," *VISIGRAPP 2019 - Proc. 14th Int. Jt. Conf. Comput. Vision, Imaging Comput. Graph. Theory Appl.*, vol. 5, pp. 165–172, Jan. 2019.

[60] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spatiotemporal Multiplier Networks for Video Action Recognition," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7445–7454.

[61] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spatiotemporal Residual Networks for Video Action Recognition," *Proceed- ings Adv. Neural Inf. Process. Sys- tems*, vol. 2017–Janua, no. Nips, pp. 7445–7454, Nov. 2016.

[62] "UCF101 Dataset Split 1." [Online]. Available:

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

[63] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size,” no. February, pp. 0–13, 2016.

[64] O. Köprülü, N. Kose, A. Gunduz, and G. Rigoll, “Resource Efficient 3D Convolutional Neural Networks,” Apr. 2019.