BOSTON UNIVERSITY

GRADUATE SCHOOL OF ARTS AND SCIENCES

Thesis

LEARNING TEMPORAL VARIATIONS FOR ACTION RECOGNITION

by

QILI ZENG

B.Eng., Southeast University, 2018

Submitted in partial fulfillment of the

requirements for the degree of

Master of Science

2020

Approved by

First Reader

Janusz Konrad, Ph.D.
Professor of Electrical and Computer Engineering

Second Reader

Margrit Betke, Ph.D.
Professor of Computer Science

Third Reader

Brian Kulis, Ph.D.
Associate Professor of Electrical and Computer Engineering

Nam si ratio et prudentia curas,
non locus effusi late maris arbiter aufert,
caelum, non animum, mutant,
qui trans mare currunt. Horace, Epistles I. XI. 25

Acknowledgments

This thesis could not have been completed without the support from my advisors, mentors, friends, and families. I take this opportunity to sincerely appreciate them.

I would like to firstly express my highest gratitude to my advisor, Professor Janusz Konrad, for providing the opportunity for me to finish this thesis in his research group. His enthusiasm, concentration, and rigor on research guided me to be a junior researcher and inspired me with the confidence in my future pursuit.

I would also like to thank Ozan Tezcan, a PhD candidate in the Visual Information Processing Lab in the Department of Electrical and Computer Engineering, for countless, constructive discussions and teaching me various research skills. His solid foundation in machine learning helped me get through many bottlenecks and setbacks in this project. It is my fortune to work with a responsive collaborator like him.

During my first year, Vasili Ramanishka, from Image and Video Computing Group, enlightened me with his rich experience and deep insights in video analysis. I was exposed to brand new and inspiring perspectives towards video, motion, and action every time I talked to him. He also deserves my gratitude and respect.

It is also my great honor to have Professor Margrit Betke (supported by NSF 1838193) and Professor Brian Kulis as the members of the thesis committee.

Finally, I would like to dedicate this thesis to my parents, who always give the endless support and generous understanding for any decision I make. I feel lucky and warmed every moment I have them in my mind.

Qili Zeng

LEARNING TEMPORAL VARIATIONS FOR ACTION RECOGNITION

QILI ZENG

ABSTRACT

As a core problem in video analysis, action recognition is of great significance for many higher-level tasks, both in research and industrial applications. With more and more video data being produced and shared daily, effective automatic action recognition methods are needed. Although, many deep-learning methods have been proposed to solve the problem, recent research reveals that single-stream, RGB-based networks are always outperformed by two-stream networks using both RGB and optical flow as inputs. This dependence on optical flow, which indicates a deficiency in learning motion, is present not only in 2D networks but also in 3D networks. This is somewhat surprising since 3D networks are explicitly designed for spatio-temporal learning.

In this thesis, we assume that this deficiency is caused by difficulties associated with learning from videos exhibiting strong temporal variations, such as sudden motion, occlusions, acceleration, or deceleration. Temporal variations occur commonly in real-world videos and force a neural network to account for them, but often are not useful for recognizing actions at coarse granularity. We propose a Dynamic Equilibrium Module (DEM) for spatio-temporal learning through adaptive Eulerian motion manipulation. The proposed module can be inserted into existing networks with separate spatial and temporal convolutions, like the R(2+1)D model, to effectively handle temporal video variations and learn more robust spatio-temporal features. We demonstrate performance gains due to the use of DEM in the R(2+1)D model on miniKinetics, UCF-101, and HMDB-51 datasets.

Contents

List of Tables

List of Figures

List of Abbreviations

CNN	.....	Convolutional Neural Network
C3D	.....	Convolutional 3D Neural Network
DEM	.....	Dynamic Equilibrium Module
DT	.....	Dense Trajectory
DTPN	.....	Dynamic Temporal Pyramid Network
GRU	.....	Gated Recurrent Unit
HOG	.....	Histogram of Gradient
iDT	.....	improved Dense Trajectory
I3D	.....	Inflated 3D Convolutional Network
LSTM	.....	Long Short-term Memory
MLP	.....	Multilayer Perceptrons
P3D	.....	Pseudo-3D Residual Networks
RNN	.....	Recurrent Neural Network
R(2+1)D	.....	Residual (2+1)D Neural Network
R3D	.....	Residual 3D Neural Network
S3D	.....	Separable 3D Neural Network
SGD	.....	Stochastic Gradient Descent
SIFT	.....	Scale Invariant Feature Transform
STIP	.....	Space-Time Interest Points
SURF	.....	Speeded Up Robust Feature
TDN	.....	Temporal Difference Networks
TRN	.....	Temporal Relation Networks
TSN	.....	Temporal Segment Networks

Chapter 1

Introduction

Video is a ubiquitous information medium in the modern society, with a range of applications in television, surveillance and social media. Compared to static images, video provides a more natural and comprehensive way to record scenes and capture events, which is much closer to the way humans observe and analyze objects, activities and emotions.

In the last two decades, thanks to the rapid growth of internet bandwidth and mobile devices, there has been an explosive increase in the number of videos uploaded every day to online platforms, such as Facebook, Instagram, Youtube and many others. While such videos are mostly intended for entertainment, they can also be used for personalized recommendations, targeted advertising, censorship, etc. This, however, necessitates an analysis of video streams to understand their content. Obviously, manual inspection is time-consuming, expensive and hence not scalable. Therefore, automatic computer algorithms for video understanding are needed.

In the field of computer vision, video modeling and understanding has been studied for decades leading to various algorithms and applications, such as video inpainting, video captioning, action recognition, etc. Due to the fact that most videos are human-related, understanding human actions is not only an important goal in itself, but also serves as the foundation for other higher-level tasks, such as action anticipation, video generation, etc. Therefore, human action recognition has been one of core topics in computer vision for decades. Although largely solved in simple cases, human action

recognition still faces significant challenges due the complexity of natural human motion, variations in body build and pose, and interactions between actors and their environment.

Although traditional methods based on hand-crafted feature extraction perform well in many cases, they do not generalize to large-scale, real-world video datasets and would not perform well on massive daily uploads of, often, complex videos. Since deep convolutional neural networks (CNN) became a dominant solution to universal image understanding, people have attempted to replicate their success on video, which is often considered as a sequence of still images.

3D Convolutional Networks (3D ConvNets) were proposed based on a natural extension from 2D pixels to 3D voxels in order to learn from video data in an elegant way. However, this simple transformation from spatial convolution to spatio-temporal convolution failed to achieve desirable performance with regard to the accuracy of prediction. It has been observed that 3D ConvNets with only RGB as an input do not perform as well as their two-stream counterparts which take RGB and optical flow as two independent inputs. Since optical flow is computed deterministically from video frames, the introduction of optical flow can be considered as feature engineering which, in the context of deep learning, is a limitation and suggests that current 3D ConvNets could be improved in terms of learning spatio-temporal information.

A lot of research has been conducted in order to understand this deficiency and seek ways to improve recognition accuracy. Some of the developed approaches have focused on exploring the possibilities of training single RGB stream with extra supervision (Stroud et al., 2018; Crasto et al., 2019). Other approaches have attempted to analyze the reasons for optical flow's impact and to incorporate the learning of optical flow into the networks (Sevilla-Lara et al., 2017; Ng et al., 2018; Zhu et al., 2018a; Zhao and Snoek, 2019). Still other solutions facilitated indirect motion modeling

based on the nature of spatio-temporal data (Zhao et al., 2018a; Zhao et al., 2018b; Wang et al., 2018; Feichtenhofer et al., 2019; Wang et al., 2019a; Liu et al., 2019). The proposed method falls under the umbrella of the last category of methods.

In this thesis, we propose and develop a Dynamic Equilibrium Module (DEM) designed to explicitly account for the temporal variability of people and objects' dynamics. This module can be used as an enhancement to various CNN architectures. It has been developed based on one observation and one hypothesis.

First, we noticed that motion in real-world videos can dramatically change in just a few frames and, therefore, is difficult to predict. Since the human visual system is endowed with bi-directional attention, i.e., top-down and bottom-up (Lu and Sperling, 1995; Buschman and Miller, 2007), and with prior knowledge of object structure, maintaining visual coherence with respect to moving objects over time is not difficult, which contributes to handling motion variations. However, in the case of neural networks trained from scratch, learning high temporal variations of motion leads to high computational cost. Considering the mechanism for motion interpretation inside a CNN, which was demonstrated in early CNN-based optical flow estimators (Dosovitskiy et al., 2015; Ilg et al., 2017), a large displacement between adjacent frames cannot be handled by local convolutional operators. When recognizing actions from several frames (a video segment), the network must include different groups of filters across layers in order to adapt to time-varying motion. This, on the other hand, brings about extra difficulties in making accurate prediction with given supervision and limited-scale dataset.

Nevertheless, for general action recognition, where each trimmed short video is associated with one coarse label (e.g., "playing soccer" rather than specific actions of "dribbling", "passing" or "shooting"), fine-grained motion information including acceleration and deceleration may be unnecessary. For instance, humans would not

need to know whether the speed of a swimmer increases or decreases before understanding the person is swimming. In most cases, a coarse description of objects and their dynamics is sufficient to generate reliable prediction for action category.

Having argued that a detailed motion description is not required and even could lead to extra computational cost during learning, we propose to develop a specialized network structure to deal with time-varying movement. We expect this structure could be helpful in reducing the sensitivity of the network to temporal variations and enhancing network's robustness to different motion types. DEM, that we propose in this thesis, encapsulates this idea.

DEM can be inserted into existing models such as R(2+1)D (Tran et al., 2018), S3D (Xie et al., 2018) and P3D (Qiu et al., 2017), where a 3D convolution is decomposed into consecutive convolutions, a 2D spatial one and a 1D temporal one. More specifically, DEM inserted between a spatial convolution and a temporal one is expected to stabilize the spatio-temporal learning by extracting Eularian motion representation from adjacent spatial feature maps and merging this information back into the backbone network before temporal convolution. Compared to SlowFast Networks (Feichtenhofer et al., 2019) and Random Temporal Skipping (Zhu and Newsam, 2018), where temporal variations are handled by explicit multi-rate sampling, DEM shares the same motivation but provides a more flexible solution to motion modeling.

The subsequent chapters are organized as follows. Chapter 2 gives a review of related work, including major milestones and recent progress in action recognition, motion representation and video sequence modeling. Chapter 3 describes the proposed Dynamic Equilibrium Module (DEM) and intuitively explains how it works. Chapter 4 quantitatively demonstrates the effectiveness of DEM through experimental results and an ablation study. Chapter 5 summaries the thesis and discusses possible extensions in the future.

Chapter 2

Related Work

2.1 Action Recognition

Different from image classification, video-based action recognition requires reliable motion features to reflect the dynamic changes occurring in videos. Laptev *et al.* proposed a spatio-temporal interest points (STIPs) method (Laptev and Lindeberg, 2003) by extending Harris corner detector to 3-dimensional space to capture motion. Similarly, 3D extensions of Scale-Invariant Feature Transform (SIFT) (Klaeser et al., 2008), Speeded Up Robust Features (SURF) (Willems et al., 2008) and Histogram of Oriented Gradients (HOG) (Laptev et al., 2008) have also been introduced. Guo *et al.* proposed to efficiently generate low-dimensional representation for pre-computed motion descriptors, such as optical flow and silhouette tunnel, *via* log-covariance matrices (Guo et al., 2010; Guo et al., 2013). Dense Trajectories (DT) (Wang et al., 2011) and the method's successor, Improved Dense Trajectories (iDT) (Wang et al., 2013) were the best performing solutions before deep learning's remarkable success. However, iDT is computationally expensive and becomes intractable on large-scale video datasets.

Since AlexNet's (Krizhevsky et al., 2012) breakthrough in image classification, there have been active explorations into action recognition using neural networks. In early attempts, features were extracted in each frame through Convolutional Neural Networks (CNNs) pretrained on an image dataset and these frame-level representations were then fused *via* feature pooling (Joe Yue-Hei Ng et al., 2015), high-

dimensional feature encoding (Girdhar et al., 2017; Diba et al., 2017; Lin et al., 2018) or recurrent neural networks (RNNs)(Donahue et al., 2015; Joe Yue-Hei Ng et al., 2015; Srivastava et al., 2015) to generate a video's global description.

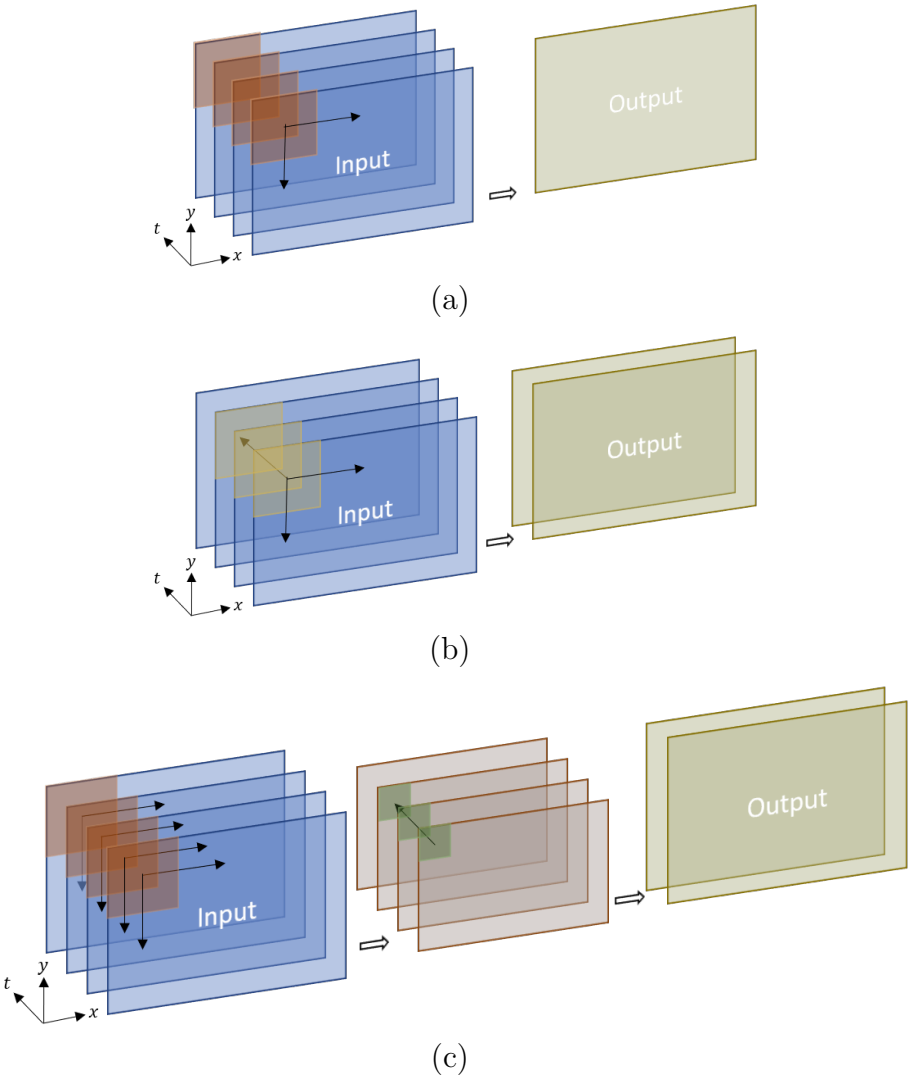


Figure 2·1: Comparison between different spatio-temporal convolutions, where video frames are represented by blue rectangles: (a) 2D convolution for video modeling; (b) 3D convolution; (c) (2+1)D convolution.

Simonyan *et al.* proposed a biologically-inspired Two-Stream Network (Simonyan and Zisserman, 2014) which computes the representation of appearance and motion

separately by taking video frames and optical flow as inputs to two different branches of networks, respectively. Two-stream networks successfully alleviate the problem of weak temporal modeling ability of 2D convolutions by using motion features explicitly. A number of methods followed this design, such as Temporal Segment Networks (TSN) (Wang et al., 2016), Hidden Two-Stream Networks (Zhu et al., 2018b), Spatiotemporal Pyramid Network (Wang et al., 2017). Feichtenhofer *et al.* proposed a fusion scheme for appearance and motion features (Feichtenhofer et al., 2016). Two-Stream networks also inspired research with other modalities including audio (Li et al., 2017), motion vectors (Wu et al., 2018), and estimated motion representations (Zhao et al., 2018a). However, computing optical flow is expensive and, more importantly, could be considered as feature engineering in the context of deep networks since it is pre-computed from video frames.

3D convolution is another family of solutions, which learns spatio-temporal representation in a unified way. An early version of 3D ConvNets was proposed by Ji *et al.* (Ji et al., 2013), while Tran *et al.* proposed a modern 3D convolutional network (C3D) (Tran et al., 2015) with more mature deep learning configurations. Subsequent work with 3D Convolution includes I3D (Carreira and Zisserman, 2017) and R3D (Hara et al., 2018), which inflated a 2D Inception Network and a Residual Network into corresponding 3D versions along the temporal dimension. X3D (Feichtenhofer, 2020) further explored feasible solutions to expand 2D networks along other axes, such as bottleneck width and depth. Considering the heavy computational complexity of 3D convolutions and the inexact symmetry of 3D kernels (spatial and temporal information is intermixed and considered jointly), researchers made an effort to employ 2D and 3D operators together in proper order (Xie et al., 2018; Zhou et al., 2018b), decompose 3D convolution into consecutive 2D and 1D convolutions (Qiu et al., 2017; Tran et al., 2018), or apply 2D operations along 3 planes of spatio-temporal tensors

(horizontal, vertical and temporal) whose outputs are fused by a weighted summation (Li et al., 2019).

Although 3D ConvNets are designed to jointly learn spatio-temporal features, almost all of them achieve improved performance with an extra flow stream, which means that complementary motion information is still beneficial and thus current 3D networks are not sufficiently capable of modeling motion as expected. Some work has been undertaken to understand how optical flow helps spatio-temporal learning in action recognition (Sevilla-Lara et al., 2017; Güney et al., 2019). D3D (Stroud et al., 2018) and MARS (Crasto et al., 2019) methods transfer knowledge from an optical-flow stream to an RGB stream *via* distillation and demonstrate that, under proper supervision, 3D networks could perform similarly to their two-stream counterparts. Researchers also investigated possible approaches to learn from dynamics without explicitly using or estimating motion features. Most of this work was proposed based on the temporal structure and internal relations within a sequence of frames. Non-local Networks (Wang et al., 2018) established pixel-to-pixel relations across all feature maps, implicitly learning motion through generalized self-attention. Correlation Networks (Wang et al., 2019a) established frame-to-frame matches over convolutional feature maps through learnable correlation operators. Temporal Shift Module (Lin et al., 2019) performs efficient temporal modeling by moving the feature map along the temporal dimension, which enables high-speed online action recognition with 2D networks.

2.2 Motion Representation

Motion in a video sequence implies a relationship between video frames and reflects important properties of moving objects, such as shape, texture, and 3D structure, from which distinctive patterns can be captured and used for many tasks. Various

motion descriptors have been developed in the past, however most of them can be classified as either a Lagrangian or Eulerian approach.

The Lagrangian perspective on motion considers it as the movement of particles in a medium. Among the most successful hand-crafted Lagrangian approaches are dense optical flow (Horn and Schunck, 1981) and improved dense trajectory (Wang and Schmid, 2013) methods. Since accurate optical flow computation using variational approaches requires hundreds of iterations (Zach et al., 2007), CNNs were explored for optical flow estimation as well (Dosovitskiy et al., 2015; Ilg et al., 2017; Ranjan and Black, 2017; Sun et al., 2018; Fan et al., 2018). Although FlowNetS (Dosovitskiy et al., 2015) demonstrates the ability of directly converting image pairs into optical flow, special structures or intermediate representations including correlation layers (Dosovitskiy et al., 2015), cost volume (Xu et al., 2017) and image pyramids (Ranjan and Black, 2017) have been adopted for more robust estimation in highly-dynamic scenes.

The Eulerian perspective on motion, on the other hand, considers motion as a variation of pixel values at fixed positions over time. Without explicitly capturing pixel correspondences, Eulerian motion features are more sensitive to occlusions, blur, and large displacements, and thus only provide a rough motion description. Previous explorations have successfully employed Eulerian motion in video motion magnification (Wu et al., 2012; Wadhwa et al., 2013; Oh et al., 2018a) and video frame interpolation (Meyer et al., 2018).

In the area of action recognition, RGB differences, the simplest Eulerian motion representation, have been used in Temporal Segment Network (Wang et al., 2016). However, its experimental results are inferior to those of the same network operating on optical flow as the input. Similar to (Wadhwa et al., 2013), phase-based motion, where movement's state is represented by the phase of pixels in complex domain,

was also applied to action recognition (Hommos et al., 2018). Temporal Difference Networks (TDN) (Ng and Davis, 2018) made an attempt to extract learning-based Eulerian motion as an independent stream for subsequent classification. All these prior methods consider Eulerian motion as a replacement for or complement of optical flow, while in the proposed DEM module, Eulerian motion is learned to manipulate spatio-temporal representations flowing through the backbone network.

2.3 Sequential and Temporal Modeling

Recurrent Neural Networks (RNN), especially with gated cells such as Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Units (GRU) (Cho et al., 2014) have been widely used for decades in order to extract and learn information from a sequence, including natural language (Sutskever et al., 2014) and sound (Eck and Schmidhuber, 2002). There is also a long history of using convolutional networks for effective and parallelizable sequence modeling (Waibel et al., 1989; van den Oord et al., 2016; Bai et al., 2018). The combinations of convolutional networks and recurrent models have been also explored, such as ConvLSTMs (Donahue et al., 2015) and VideoLSTMs (Li et al., 2018). Recent work also proposed effective long-range sequence modeling solutions solely based on attention mechanism (Vaswani et al., 2017).

Since frames are organized in a video in temporal order, video modeling often leverages time-related characteristics for specific tasks. Temporal Relation Network (TRN) (Zhou et al., 2018a) samples video frames sparsely in different temporal increments and fuses the features through multi-scale multilayer perceptrons (MLP) to explore temporal dependencies between frames at multiple time scales. Considering real-world movement is continuous and smoothly-varying, slow feature analysis (Zou et al., 2012; Jayaraman and Grauman, 2016) was proposed to utilize temporal

continuity in video for unsupervised representation learning. Wei *et al.* proposed to learn and visualize the "arrow of time", i.e., the natural temporal order of video sequence, and demonstrated its effectiveness as a self-supervised pretraining for action recognition (Wei et al., 2018). Wang *et al.* and Dwibedi *et al.* employed temporal cycle-consistency for self-supervised representation learning and achieved desirable results on several fine-grained tasks (Dwibedi et al., 2019; Wang et al., 2019b).

The model proposed in this thesis has been also inspired by prior work on video modeling using multivariate or temporal multi-scale sampling, which emphasizes learning representations for actions occurring at various speeds. Multirate Gated Recurrent Unit (mGRU) (Zhu et al., 2017) followed the idea of Clockwork RNN (Koutnik et al., 2014) and encoded video frames with different intervals. Random Temporal Skipping (Zhu and Newsam, 2018) attempted to cover all motion speed variations by randomizing the sampling rate during training in an exhaustive way. Similarly, Dynamic Temporal Pyramid Network (DTPN) (Zhang et al., 2018) also sampled frames with different frame rate to construct a natural pyramidal representation for arbitrary-length input videos. SlowFast Networks (Feichtenhofer et al., 2019) included two different network streams for both high frame-rate inputs and low frame-rate inputs, modeling motion at fine and coarse temporal resolutions separately. Temporal Pyramid Network (TPN) (Yang et al., 2020) aggregated the information of temporal variations at multiple feature levels in the backbone network in a plug-and-play manner and fused them to make the final prediction.

Chapter 3

Dynamic Equilibrium Module

In order to improve performance of CNN-based action recognition, in this chapter we propose a Dynamic Equilibrium Module (DEM). This module aims at discovering temporal variations in the input video and in its intermediate spatio-temporal representations within the backbone network. The module produces feedback signals that allow the backbone network to leverage motion information more accurately.

3.1 Eulerian Description of Temporal Variations

Temporal variations occurring in a video sequence capture the change in dynamics of objects in a 3-D scene. Such changes occur naturally in a real world and are usually unpredictable but are of key importance for recognizing actions. Before focusing on temporal variations let us first define general Eulerian motion representation in the context of neural networks.

A Eulerian motion description typically involves computing the difference of certain properties of an image sequence either in space-time or in spatio-temporal frequency domain. For instance, using temporal convolution $TConv$ with filters of size $t \times 1 \times 1$ (temporal $\times$ horizontal $\times$ vertical dimensions), the dynamics present in a video sequence could be described in the most general form as follows:

$$\Delta \mathbf{I} = TConv(\mathbf{I}_{t-1}, \mathbf{I}_{t-2}, \ldots, \mathbf{I}_0) \tag{3.1}$$

where $\mathbf{I}_t$ denotes a video frame at time t. As modern convolutional neural networks

learn representation in a hierarchical manner, we assume such operation is not only applicable to the input frames (low-level motion), but also to intermediate feature maps (high-level motion). In practice, motion description is usually inferred from a pair of input frames, in which case t would equal 2 in (3.1).

By observing adjacent video frames, humans can easily determine whether a particular frame contains large-amplitude motion, occlusions, acceleration, deceleration, etc. However, it is not obvious how to characterize such temporal variations mathematically. One possible quantitative description of such variations can be obtained by analyzing either three consecutive video frames (in the input layer) or three consecutive spatio-temporal representations (in subsequent layers), denoted $\mathbf{x}_{n-1}, \mathbf{x}_n, \mathbf{x}_{n+1}$, as follows:

$$\mathbf{D}_n = g(f(\mathbf{x}_{n-1}, \mathbf{x}_n), f(\mathbf{x}_n, \mathbf{x}_{n+1})) - f(\mathbf{x}_{n-1}, \mathbf{x}_{n+1}) \tag{3.2}$$

where f and g refer to $TConv$ operations with different filters. The role of $\mathbf{D}_n$ can be explained as follows. In case of an action that evolves uniformly in time (for example, linear, constant-velocity movement such as a cyclist coasting on a flat road), motion description based on the observation of $(\mathbf{x}_{n-1}, \mathbf{x}_{n+1})$ should be numerically close to the composition of motion descriptions based on observations of $(\mathbf{x}_{n-1}, \mathbf{x}_n)$ and $(\mathbf{x}_n, \mathbf{x}_{n+1})$ and, consequently, $\mathbf{D}_n$ should be small. If $\mathbf{D}_n$ is large, then $\mathbf{x}_n$ or $(\mathbf{x}_{n-1}, \mathbf{x}_{n+1})$ likely disobeys action uniformity in time (e.g., the cyclist makes a sudden turn). While a large value of $\mathbf{D}_n$ can be useful in recognizing a particular detail in an action (e.g., cyclist's turn), it is not helpful in determining a high-level action (i.e., cycling, in this example). In order to learn the fine details of an action, a more complex network (more parameters) or extra supervision would be needed. Therefore, the goal is to "discover" such fine details and help the backbone network compensate for them.

3.2 Module Formulation

An observation of time-varying appearance by human visual system leads to motion perception. Attributes of movement, such as velocity and acceleration, are closely related to the way the appearance of objects changes in time. In other words, what is presented in consecutive video frames determines how motion is interpreted, e.g., by speeding up a video of "touching", people may understand it as "hitting". Motion interpretation in a neural network works similarly – pattern matching between frames could fail if excessive temporal variability is present. Following the idea of motion magnification (Oh et al., 2018b), we believe that spatio-temporal representation learning can be influenced by adaptive manipulation of the appearance in an image sequence.

However, temporal variations in a video lead to extra and unnecessary cost for a network that is trying to predict a coarse action label rather than a fine detail in an action. Accounting for this fine detail would require the network to learn different groups of filters across layers thus increasing network's complexity. If the temporal variations in a video could be suppressed or, in other words, if the dynamics in a video could be equilibrated, then learning from spatio-temporal data could be significantly simplified. To this end, we introduce the Dynamic Equilibrium Module (DEM) that attempts to generate motion compensation based on the detected temporal variations in a video and pass this information back to the backbone network for motion manipulation.

DEM implements equation (3.2) to estimate temporal variations around a certain frame, then fuses this representation with the original spatial feature map to generate the compensation signal. Figure 3·1 shows a diagram of DEM and its interaction with a unit in the backbone network, in our case a (2+1)D layer.

Function f in DEM is realized by using dilated convolution (Yu and Koltun, 2016),

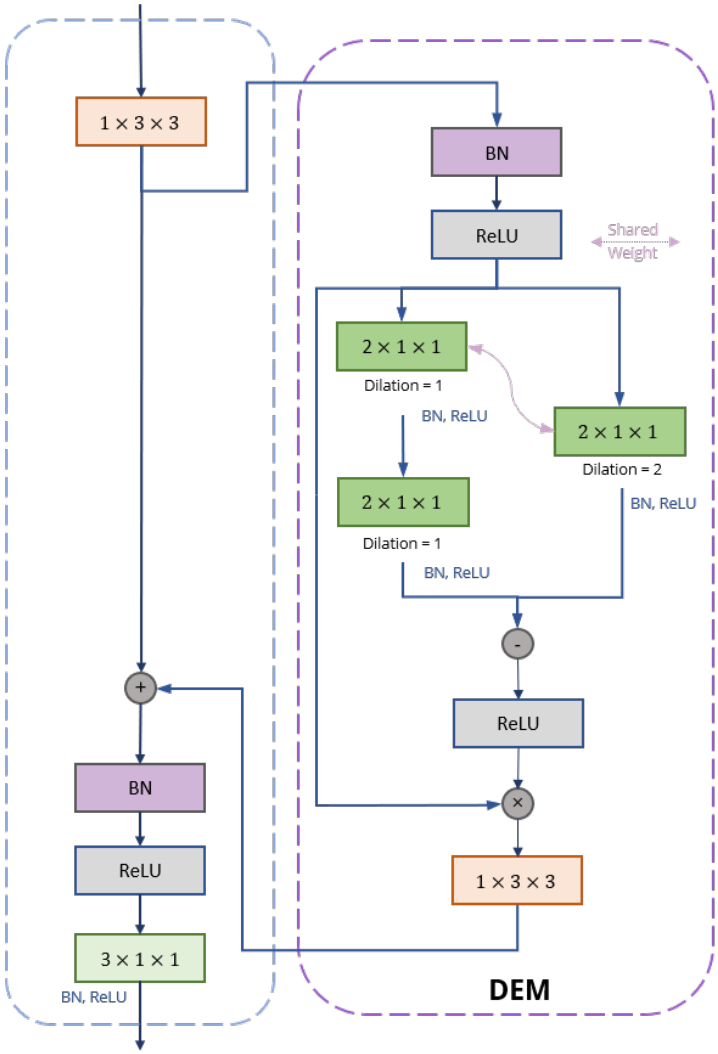


Figure 3·1: Illustration of a DEM inserted into a (2+1)D layer. Red and green blocks represent spatial and temporal convolutional layers, respectively. Arithmetic operation nodes are all pixel-wise.

where computing $f(\mathbf{x}_n, \mathbf{x}_{n+1})$ is implemented with a normal $TConv$ and computing $f(\mathbf{x}_{n-1}, \mathbf{x}_{n+1})$ is implemented by the same $TConv$ with the dilation rate of 2. See Figure 3·2 for detailed illustration on a simple example.

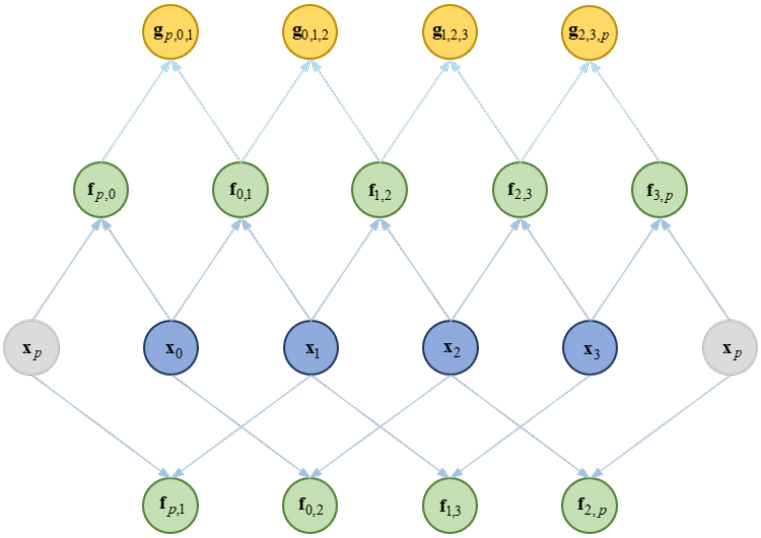


Figure 3·2: Computing a representation of temporal variations. This example shows the situation where the temporal length of input is 4. $\{\mathbf{x}_i\}_{i=0}^{3}$ denotes the input sequence and $\mathbf{x}_p$ denotes padding. $\mathbf{f}_{m,n}$ refers to the result of applying f to $\mathbf{x}_m$ and $\mathbf{x}_n$, and similarly $\mathbf{g}_{l,m,n}$. The difference between $\mathbf{g}_{l,m,n}$ and $\mathbf{f}_{l,n}$ reflects the temporal variations related to $\mathbf{x}_m$, as defined in equation (3.2).

In order to generate feedback for motion manipulation, we have considered various approaches to fusing the original spatio-temporal representation with the representation of temporal variations computed in equation (3.2), including concatenation, bi-linear pooling, and pixel-wise multiplication. We selected pixel-wise multiplication for all experiments in this thesis due to its higher efficiency and lower memory usage.

Although DEM could be inserted between a pair of spatial and temporal convolutions without any modification, the number of parameters in the whole network would increase after insertion. Therefore, similarly to the R(2+1)D Network (Tran et al., 2018), in all experiments we adjusted the number of midplane channels, i.e., the number of spatial filters, in all the (2+1)D convolutional layers with a DEM to ensure

that the total number of parameters in the network is equivalent to that of R3D networks. More specifically, the number of parameters N_{param} in one 3D convolutional layer can be calculated by

$$N_{3D} = t \times d \times d \times N_{in} \times N_{out} \tag{3.3}$$

where t refers to the temporal length, d is the spatial width and height, and N_{in}, N_{out} are the numbers of channels in the input and output tensors, respectively. A (2+1)D convolutional layer would then have $N_{(2+1)D}$ parameters:

$$N_{(2+1)D} = (1 \times d \times d \times N_{in}) \times N_{mid} + (t \times 1 \times 1 \times N_{mid}) \times N_{out} \tag{3.4}$$

where N_{mid} denotes the number of midplane channels. After DEM insertion, the number of parameters in a (2+1)D + DEM unit would be

$$\begin{aligned} N_{(2+1)D+DEM} &= (1 \times d \times d \times N_{in}) \times N_{mid} + (t \times 1 \times 1 \times N_{mid}) \times N_{out} \\ &\quad +2 \times (2 \times 1 \times 1 \times N_{mid}) \times N_{mid} \\ &\quad +(1 \times d \times d \times N_{mid}) \times N_{mid} \end{aligned} \tag{3.5}$$

We solve for N_{mid} in order to have the same number of network parameters in the three cases. Table 3.1 provides detailed results of such an adjustment for 18-layer models. The diagram of an R(2+1)D-18 network is provided in Figure 3·3.

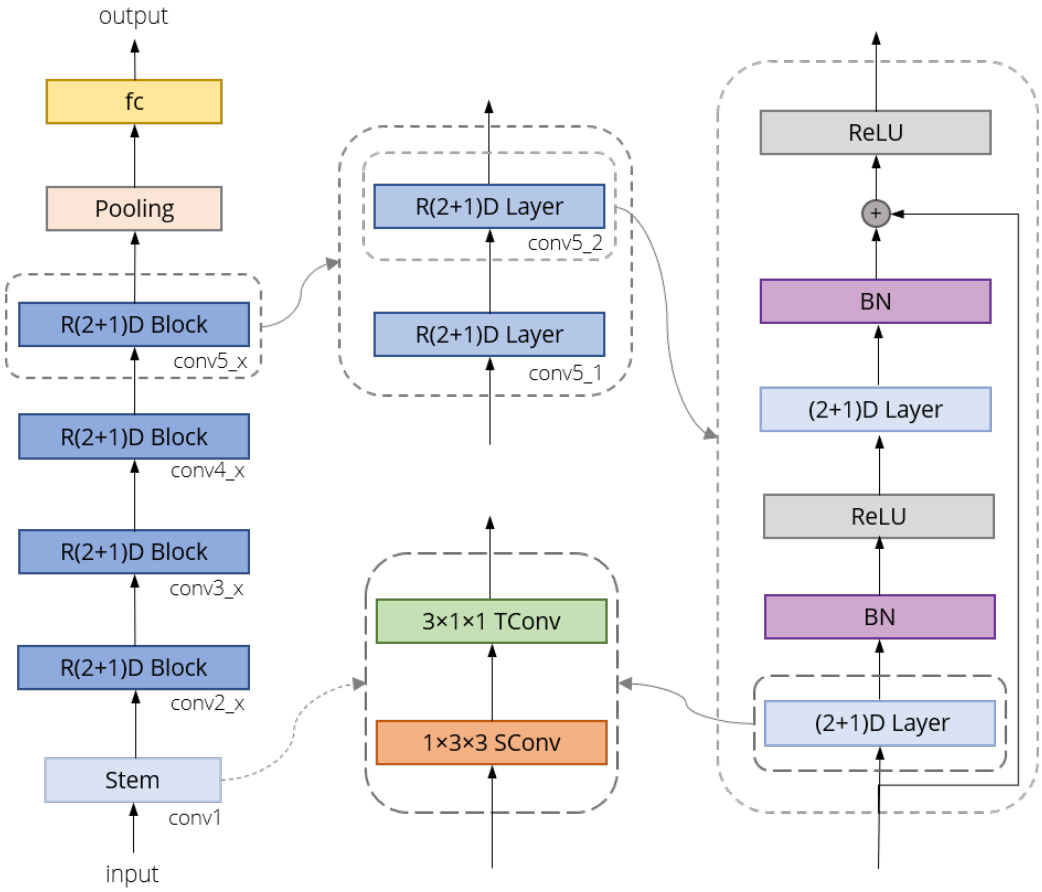


Figure 3·3: Illustration of an R(2+1)D-18 network. Stem (conv1) layer contains only one (2+1)D layer. Replacing all the (2+1)D layers by 3D convolutions results in R3D network. DEM can be inserted into (2+1)D layers between the spatial convolution and temporal convolution, as illustrated in Figure 3·1.

layer	output size	R3D-18	R(2+1)D-18	R(2+1)D-18 w/ DEM
conv1	$L \times 56 \times 56$	3×7^2, 64, stride 1×2^2	$1 \times 7^2, 83, 3 \times 1^2$, 64, stride 1×2^2	$1 \times 7^2, 20, 3 \times 1^2$, 64, stride 1×2^2
conv2_x	$L \times 56 \times 56$	$\begin{bmatrix} 3 \times 3^2, 64 \\ 3 \times 3^2, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 3^2, 144, 3 \times 1^2, 64 \\ 1 \times 3^2, 144, 3 \times 1^2, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 3^2, 67, 3 \times 1^2, 64 \\ 1 \times 3^2, 67, 3 \times 1^2, 64 \end{bmatrix} \times 2$
conv3_x	$\frac{L}{2} \times 28 \times 28$	$\begin{bmatrix} 3 \times 3^2, 128 \\ 3 \times 3^2, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 3^2, 230/288, 3 \times 1^2, 128 \\ 1 \times 3^2, 288, 3 \times 1^2, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 3^2, 98/134, 3 \times 1^2, 128 \\ 1 \times 3^2, 134, 3 \times 1^2, 128 \end{bmatrix} \times 2$
conv4_x	$\frac{L}{4} \times 14 \times 14$	$\begin{bmatrix} 3 \times 3^2, 256 \\ 3 \times 3^2, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 3^2, 460/576, 3 \times 1^2, 256 \\ 1 \times 3^2, 576, 3 \times 1^2, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 3^2, 197/269, 3 \times 1^2, 256 \\ 1 \times 3^2, 269, 3 \times 1^2, 256 \end{bmatrix} \times 2$
conv5_x	$\frac{L}{8} \times 7 \times 7$	$\begin{bmatrix} 3 \times 3^2, 512 \\ 3 \times 3^2, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 3^2, 921/1152, 3 \times 1^2, 512 \\ 1 \times 3^2, 1152, 3 \times 1^2, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 3^2, 394/538, 3 \times 1^2, 512 \\ 1 \times 3^2, 538, 3 \times 1^2, 512 \end{bmatrix} \times 2$
	$1 \times 1 \times 1$	global average pooling, fully-connected layer, and softmax		

Table 3.1: Comparison of network structures for R3D and R(2+1)D without and with DEM. Convolutional residual blocks are shown in brackets, next to the number of times each block is repeated in the stack. The dimensions given for filters and outputs are time, height, and width. For R3D, a single layer is denoted as [dimension, number of output channels], while for R(2+1)D it is represented as [dimension of spatial convolution, number of mid-plane channels, dimension of temporal convolution, number of output channels].

Chapter 4

Experimental Results

4.1 Experimental Setup

4.1.1 Datasets

Kinetics-400 (Kay et al., 2017) is a large-scale human-centric video dataset collected from Youtube, containing 400 human action classes, 240k training videos and 20k validation videos. Since the testing subset is reserved for competition and its labels are not provided, we use the validation subset for testing our models. Unfortunately, training on Kinetics-400 would be extremely time-consuming on our hardware, thus in most experiments we trained and tested our models on a subset of Kinetics-400, called **miniKinetics** (Kinetics-200) (Xie et al., 2018). It includes 200 action classes with 80k and 5k videos for training and validation. As the availability of dataset videos varies over time due to deletion or withdrawal, there might be fewer videos that could be effectively downloaded. In our experiments, we were able to collect only 77,152 and 4,988 videos as training and validation subsets.

UCF-101 (Soomro et al., 2012) is another trimmed video dataset for human action recognition, consisting of 13,320 videos with 101 annotated classes. The dataset is officially divided into training (70%) and testing (30%) subsets in three different splits.

HMDB-51 (Kuehne et al., 2011) comprises 6,766 videos collected from real movies and YouTube and annotated into 51 classes. Similarly, HMDB-51 also has three splits for training (80%) and testing (20%).

4.1.2 Training Configuration

Our model was implemented in PyTorch (Paszke et al., 2019). We used Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.99$. The initial learning rate is 0.004 and is divided by 10 when validation loss becomes saturated. Our models are trained from scratch and all the convolutional layers are initialized with Kaiming Initialization (He et al., 2015). Following (He et al., 2018), we initalize γ vectors to 1 and β vectors to 0 in all the synchronized batch normalization layers.

We extract 16 frames as one sample. The extracted frames are first resized to 171×128 multiplied by a random factor within $[1.0, 1.2]$ (spatial jittering) and then randomly cropped to 112×112, i.e., every input sample has $16 \times 112 \times 112$ dimension. We also randomly flip the frames horizontally with the probability of 0.5 and randomly rotate them by an angle between $-15°$ and $15°$. In each epoch, we sample 4 clips per video randomly (temporal jittering) so that the size of an epoch is increased to $\sim$ 30k. Training takes about 40 epochs on miniKinetics. We also terminate the training process when learning rate drops below 4e-7.

Due to limited computing resources, we deployed NVIDIA APEX[1] on 4 Tesla P100s/V100s for mixed-precision training and testing. Compared to full precision (32-bit), mixed precision leads to memory savings and computing acceleration (on V100 machines), at the cost of possible performance loss. Furthermore, since we found out that the original DEM would lead to numerical overflow under mixed-precision mode, we alternatively use ReLU-6 in DEM instead of ReLU in all experiments. We have to point out that the original design with ReLU works well in full-precision. In the experiments on miniKinetics, we use the total batchsize of 192, simulated by gradient accumulation on a physical batchsize of 96.

[1] `https://github.com/NVIDIA/apex`

4.1.3 Evaluation Metrics

We report both clip-based and video-based performance using top-1/top-5 accuracy. The clip-based metric assumes each clip from a video shares the label with the video. In this case, the accuracy for a dataset is computed using the predictions for all clips extracted from the dataset. In the video-based metric, predictions (probability distribution) for multiple clips from the same video are averaged to form a global description of the video and then used in the accuracy computation. Video-based accuracy describes the ability of a model to predict labels in a normal off-line setting, while clip-based accuracy is important for tasks when only a limited part of a video is available, such as in action anticipation. Following common settings, we extract 10 clips uniformly from each video in the validation subset in case of miniKinetics and 3 clips in case of the other two datasets. Unlike in the training stage, the extracted frames are center-cropped to avoid randomness.

4.2 Results and Discussion for miniKinetics

4.2.1 Main Results

In order to demonstrate the effectiveness of the DEM module, we compare the performance of the original R(2+1)D-18 model with R(2+1)D-18 equipped with DEM on the miniKinetics dataset. Following the most commonly-used evaluation scheme (Tran et al., 2018), we train and test the models using clips composed of 16 consecutive frames, i.e., subsampling rate of 1 when extracting a clip from video. Tables 4.1 and 4.2 show video-based and clip-based top-1 accuracy performance of both models, respectively.The first row in both tables reports the case when both the training and testing are performed with the sub-sampling rate of 1. Clearly, R(2+1)D-18 with DEM (w/) outperforms the original model without DEM (w/o).

Figure 4·1 shows the distribution of accuracy difference between the model with

Table 4.1: Video-based performance evaluation on miniKinetics for R(2+1)D-18 with (w/) and without (w/o) DEM. SR and Acc. @1 are abbreviations of the subsampling rate and Top-1 Accuracy, respectively. Using the accuracy of the model trained and tested with the same subsampling rate as the reference, Δ Acc. is calculated by testing the model at a different subsampling rate and subtracting the corresponding accuracy from the reference value.

Training SR	Testing SR	Testing Acc. @1 /%		Δ Acc. @1 /%	
		w/o DEM	w/ DEM	w/o DEM	w/ DEM
1	1	52.52	**53.53**	0.00	0.00
	2	51.63	**53.13**	-0.89	-0.40
	4	47.27	**49.36**	-5.25	-4.17
2	1	50.50	**54.30**	-1.74	-1.55
	2	52.24	**55.85**	0.00	0.00
	4	50.66	**54.75**	-1.58	-1.10
4	1	48.22	**49.00**	-7.04	-7.27
	2	53.67	**54.26**	-1.59	-2.01
	4	55.26	**56.27**	0.00	0.00

Table 4.2: Clip-based performance evaluation on miniKinetics for R(2+1)D-18 w/ and w/o DEM.

Training SR	Testing SR	Testing Acc. @1 /%		Δ Acc. @1 /%	
		w/o DEM	w/ DEM	w/o DEM	w/ DEM
1	1	40.45	**42.30**	0.00	0.00
	2	40.25	**42.45**	-0.20	+0.15
	4	38.04	**40.61**	-2.41	-1.69
2	1	38.69	**41.90**	-2.73	-2.79
	2	41.42	**44.69**	0.00	0.00
	4	41.77	**45.19**	+0.35	+0.50
4	1	36.35	**36.81**	-9.09	-9.29
	2	41.63	**42.56**	-3.81	-3.54
	4	45.44	**46.10**	0.00	0.00

DEM against the model without DEM across all miniKinetics classes. Note, that 129 out of 200 classes are predicted more accurately when using DEM. The top 15 and bottom 15 classes in terms of accuracy improvement due to the insertion of DEM are shown in Figure 4·2.

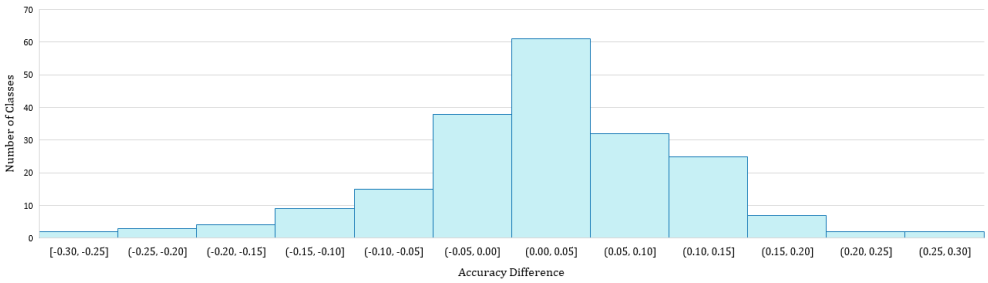


Figure 4·1: Histogram of top-1 accuracy difference due to the use of DEM and computed across all miniKinetics action classes. The difference is calculated as the per-class accuracy of R(2+1)D-18 with DEM minus that of R(2+1)D-18 without DEM. All 200 classes are included in this evaluation.

4.2.2 Handling Temporal Variations

In order to verify the ability of DEM to handle temporal variations in videos, Tables 4.1 and 4.2 show additional results from experiments using different combinations of training and testing sub-sampling rates. Experiments were performed in two scenarios: the same sub-sampling rates in training and testing, and different sub-sampling rates in training and testing. We discuss each scenario below.

Experiments with matched sub-sampling rates. We increased the training and testing sub-sampling rates simultaneously from 1 to 2 to 4 (Tables 4.1 and 4.2). In the case of sub-sampling by 2, we dropped every other frame but still extracted 16 frames from a video. Clearly, the 16 extracted frames capture temporal information over a 32-frame span in the original video. In other words, a higher temporal sub-sampling rate leads to a larger temporal receptive field thus capturing information over a longer time span, i.e., larger-scale temporal information represented in

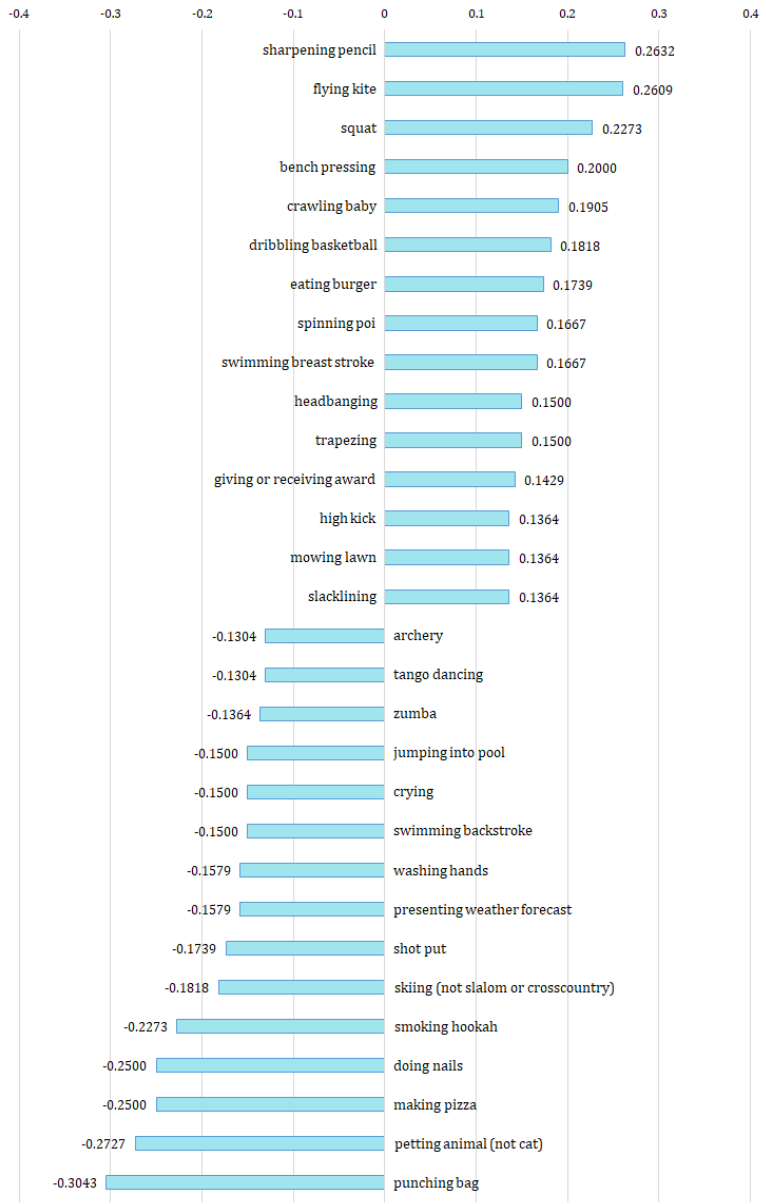


Figure 4·2: Top-1 accuracy difference for individual miniKinetics action classes. The difference is calculated as the per-class accuracy of R(2+1)D-18 with DEM minus that of R(2+1)D-18 without DEM. Only the top and bottom 15 classes are shown. There are 200 classes in total.

a video. The sub-sampling also creates a larger pixel displacement, thus simulating faster-moving objects (video with higher dynamics). When training and testing with the sub-sampling rate of 2, the model with DEM outperforms the model without DEM by a larger margin than in the case of sub-sampling by 1 (3.61% top-1 accuracy improvement compared to 1.01% for video-based evaluation and 3.27% improvement compared to 1.85% for clip-based evaluation). This seems to indicate that DEM contributes to handling higher dynamics resulting from sparser temporal sampling and thus enables the network to benefit from longer time-scale information and suppresses the side effect of larger displacements at the same time. It is worth noting that when training and testing with the sub-sampling rate of 4, R(2+1)D-18 with DEM seems to perform only slightly better than the original model without DEM (only 1.01% top-1 accuracy improvement for video-based evaluation and 0.66% improvement for clip-based evaluation). We believe that in the case of video-based evaluation this is due to the fact that the average length of videos in miniKinetics is only 300 frames and given that we sample ten 16-frame clips with uniformly-distributed start points, the total temporal receptive field of the network already spans the whole video even for sub-sampling by 2 (ten clips each covering 32 frames in the original video). Therefore, an increase of the sub-sampling rate above 2 will not lead to the extraction of longer time-scale information and thus the improvement on video-based benchmark will be smaller. However, although we also noticed that there is only marginal improvement on clip-based benchmark with the sub-sampling rate of 4, we have no clear explanations for these results so far.

Experiments with mismatched sub-sampling rates. The models trained and tested with the same sub-sampling rate are expected to achieve the best performance since training and testing are conducted on similar dynamics. In order to evaluate the DEM's capability to generate a *robust* representation of temporal variations, we use

different sub-sampling rates in training and testing. It is clear from Tables 4.1 and 4.2 that when training with sub-sampling rate of 1, R(2+1)D with DEM has smaller fluctuations in top-1 accuracy changes due to sub-sampling mismatch than R(2+1)D-18 without DEM, which indicates a contribution of DEM to more robust spatio-temporal learning. For training with sub-sampling rate of 2, both models produce similar fluctuations in clip-based evaluation, but in video-based evaluation the model with DEM produces smaller fluctuations than the one without DEM. The two models perform similarly in both evaluations when the sub-sampling rate is increased to 4. We believe the underlying reason for this is that training with a larger sub-sampling rate enhances model's robustness to temporal variations.

In all the above experiments, we implemented R(2+1)D-18 according to the original Caffe implementation[2]. We have to point out that, for unclear reasons, this implementation does not strictly follow the parameter-equivalent principles discussed in Section 3.2. The first convolutional layer, i.e., the stem layer, was developed with the midplane size of 45 in the original implementation, instead of 83, a value resulting from formula (3.4). For a fair comparison, we use the original version of R(2+1)D as a reference and adjust our R(2+1)D with DEM accordingly. Therefore, all the models with DEM in the experiments thus far had the midplane size of 14 in their stem layer instead of 20. However, we found this implementation does not lead to any special benefits and may in turn, be harmful to our models with DEM since the midplane is too small compared to common configurations for this structure. Therefore, in all the subsequent experiments our implementation of R(2+1)D-18 and R(2+1)D-18 with DEM uses parameters from Table 3.1 that are derived from equations (3.4) and (3.5).

[2] `https://github.com/facebookresearch/VMZ`

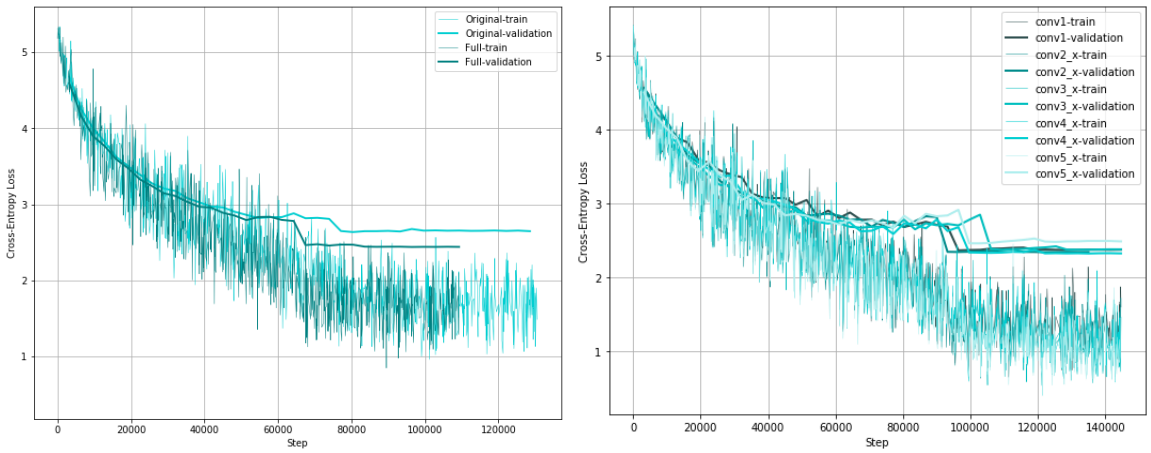
4.3 Ablation Study

As mentioned in Chapter 3.1, we believe DEM, constructed for Eulerian motion manipulation, should be applicable to handling both low-level and high-level temporal variations. In order to further study its ability to deal with temporal variations at different levels, we add the module separately to each layer in R(2+1)D-18. The impact of this addition is shown in Table 4.3.

conv1	conv2_x	conv3_x	conv4_x	conv5_x	Accuracy - Video		Accuracy - Clip	
					top-1	top-5	top-1	top-5
					55.01	80.97	43.01	70.19
√					59.84	84.34	48.21	74.52
	√				61.85	84.90	49.41	75.16
		√			61.07	84.84	49.47	75.20
			√		61.63	84.94	50.03	75.40
				√	59.54	83.66	47.73	74.10
√	√	√	√	√	58.30	83.02	45.91	72.80

Table 4.3: Impact of the insertion depth of DEM. A checked box indicates the layer into which DEM was inserted.

Since conv1 is a simple (2+1)D layer, it is reasonable that the improvement by only adding a DEM only here is less impactful than by adding the same DEM to other layers (residual blocks with four (2+1)D layers in each of them). Still, even with a DEM inserted in the first layer, it outperforms the original network by a large margin, thus demonstrating DEM's effectiveness in explicit learning of temporal variations. The insertion of DEM into any of the middle 3 layers (conv2_x, conv3_x, and conv4_x) leads to similar performance in each case, which means DEM is able to contribute to spatio-temporal learning at various stages of the network. Adding DEM to conv5_x results in less improvement, which we believe is because of the slowness of spatio-temporal features (Carreira et al., 2018; Huang et al., 2018) at



(a) (b)

Figure 4·3: Training and testing loss. (a) Comparison between R(2+1)D-18 without DEM (Original) and with DEM in all layers (Full); (b) Comparison of R(2+1)D-18 with DEM inserted in one layer only.

this stage. Since conv5_x layers are very close to the final fully-connected layer, the most detailed temporal information in the representation is filtered out and thus the temporal variations are much weaker here than in the previous layers, which makes DEM less beneficial at this stage. These experimental results also serve as evidence of the underlying mechanisms in DEM, designed to capture and handle temporal variations, as we expected.

It is interesting to note that inserting one DEM into each of the layers (the model we used in previous sections) leads to a reduced performance compared to inserting it into one layer only. This phenomenon may imply that an excessive suppression of temporal variations hinders the generation of accurate predictions. Another possible reason is the increased complexity of the model with multiple DEMs compared to those with a single DEM, which can be deduced from Figure 4·3. This figure shows that, although the training loss for the model with multiple DEMs is similar to that of the original R(2+1)D (Figure 4·3(a)), it is higher than the training loss for all models with a single DEM (Figure 4·3(b)). This is an indication that optimization

becomes harder when adding a DEM to each layer.

Comparing Table 4.3 with Tables 4.1 and 4.2, we also find that R(2+1)D-18 with and without DEM both benefit from increasing the number of midplane channels in the stem layer' R(2+1)D-18 with DEM improves more than the one without DEM.

4.4 Transfer to UCF-101 and HMDB-51

In order to demonstrate generality of the proposed module, we fine-tune and test the developed models on UCF-101 and HMDB-51 datasets. The models are pre-trained on miniKinetics with the aforementioned configurations. During fine-tuning, we use Stochastic Gradient Descent (SGD) with momentum of 0.9 as the optimizer. The initial learning rate is set to 0.0004 and divided by 10 when validation loss saturates. As recommended by (Hara et al., 2018), we only fine-tune the conv5_x and fully-connected layers in search of best performance. We also fully fine-tune the models on Split 1 of both datasets and confirm that there is only a marginal gap ($< 1\%$) between full fine-tuning and partial fine-tuning using this strategy. The results are shown in Table 4.4.

	UCF-101		HMDB-51	
	w/o DEM	w/ DEM	w/o DEM	w/ DEM
Split 1	70.58	**73.46**	42.55	**43.14**
Split 2	70.13	**73.38**	37.64	**41.11**
Split 3	70.13	**73.38**	42.16	**42.22**
Average	70.28	**73.41**	40.78	**42.16**

Table 4.4: Video-based performance evaluation on UCF-101 and HMDB-51 for R(2+1)D-18 with and without DEM. All the models are pre-trained on MiniKinetics. Top-1 Accuracy is reported.

It can be concluded from these results that R(2+1)D-18 with DEM generalizes well to other domains and still outperforms the original version without DEM. Based on the observation from Chapter 4.3, that representation in conv5_x contains limited temporal variations, we are able to claim that the ability of DEM to handle temporal variations is transferable to other datasets.

Chapter 5

Conclusions

5.1 Summary of the thesis

In this thesis, we focused on coarse-grained action recognition. We presented an effective insertable Dynamic Equilibrium Module to explicitly handle temporal variations in a video. Such variations, we believe, are difficult to handle by many spatio-temporal networks and require increased network complexity to accurately model video. Our module generates feedback to the backbone network in order to achieve motion equilibrium. As we showed, our module achieves performance gains on several mainstream action recognition benchmarks, thus indicating more robust spatio-temporal learning.

5.2 Future Work

Due to limited time and computing resources, some experiments designed for further validation and explanation of the proposed module have not been carried out. The intended main evaluation on Kinetics-400 and Something-Something datasets is currently in progress but has not been tuned to a desirable performance by the time the submission of this thesis. We hope to form a more systematic analysis of this module in the future.

As an extension of the current project, we are also looking forward to working on fine-grained video modeling tasks, such as frame in-painting and prediction, and exploring the relationship between learning for motion and learning for higher-level

aspects of video understanding, such as spatio-temporal localization, complex activity understanding, and human-object interaction.

References

Bai, S., Kolter, J. Z., and Koltun, V. (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv:1803.01271*.

Buschman, T. J. and Miller, E. K. (2007). Top-Down Versus Bottom-Up Control of Attention in the Prefrontal and Posterior Parietal Cortices. *Science*, 315(5820):1860–1862.

Carreira, J., Pătrăucean, V., Mazare, L., Zisserman, A., and Osindero, S. (2018). Massively Parallel Video Networks. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science, pages 680–697, Cham. Springer International Publishing.

Carreira, J. and Zisserman, A. (2017). Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, Honolulu, HI. IEEE.

Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y. (2014). On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *arXiv:1409.1259*.

Crasto, N., Weinzaepfel, P., Alahari, K., and Schmid, C. (2019). MARS: Motion-Augmented RGB Stream for Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7882–7891, Long Beach, CA. IEEE.

Diba, A., Sharma, V., and Gool, L. V. (2017). Deep Temporal Linear Encoding Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1541–1550, Honolulu, HI. IEEE.

Donahue, J., Hendricks, L. A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T. (2015). Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 10, Boston, MA, USA. IEEE.

Dosovitskiy, A., Philipp Fischer, Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., and Brox, T. (2015). FlowNet: Learning Optical Flow with Convolutional Networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, Santiago. IEEE.

Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., and Zisserman, A. (2019). Temporal Cycle-Consistency Learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1801–1810, Long Beach, CA. IEEE.

Eck, D. and Schmidhuber, J. (2002). A First Look at Music Composition using LSTM Recurrent Neural Networks. Technical Report IDSIA-07-02, Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale.

Fan, L., Huang, W., Gan, C., Ermon, S., Gong, B., and Huang, J. (2018). End-to-End Learning of Motion Representation for Video Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6016–6025, Salt Lake City, UT, USA. IEEE.

Feichtenhofer, C. (2020). X3D: Expanding Architectures for Efficient Video Recognition. In *arXiv:2004.04730*.

Feichtenhofer, C., Fan, H., Malik, J., and He, K. (2019). SlowFast Networks for Video Recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 6202–6211, Seoul, Korea. IEEE.

Feichtenhofer, C., Pinz, A., and Zisserman, A. (2016). Convolutional Two-Stream Network Fusion for Video Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1933–1941, Las Vegas, NV, USA. IEEE.

Girdhar, R., Ramanan, D., Gupta, A., Sivic, J., and Russell, B. (2017). ActionVLAD: Learning Spatio-Temporal Aggregation for Action Classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3165–3174, Honolulu, HI. IEEE.

Güney, F., Sevilla-Lara, L., Sun, D., and Wulff, J. (2019). "What Is Optical Flow For?": Workshop Results and Summary. In Leal-Taixé, L. and Roth, S., editors, *European Conference on Computer Vision Workshops*, volume 11134, pages 731–739, Cham. Springer International Publishing.

Guo, K., Ishwar, P., and Konrad, J. (2010). Action Recognition Using Sparse Representation on Covariance Manifolds of Optical Flow. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 188–195, Boston, MA, USA. IEEE.

Guo, K., Ishwar, P., and Konrad, J. (2013). Action Recognition From Video Using Feature Covariance Matrices. *IEEE Transactions on Image Processing*, 22(6):2479–2494.

Hara, K., Kataoka, H., and Satoh, Y. (2018). Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6546–6555, Salt Lake City, UT, USA. IEEE.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, Santiago, Chile. IEEE.

He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., and Li, M. (2018). Bag of Tricks for Image Classification with Convolutional Neural Networks. *arXiv:1812.01187*.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Hommos, O., Pintea, S. L., Mettes, P. S. M., and van Gemert, J. C. (2018). Using Phase Instead of Optical Flow for Action Recognition. In Leal-Taixé, L. and Roth, S., editors, *European Conference on Computer Vision Workshop (ECCVW)*, volume 11134, pages 678–691, Cham. Springer International Publishing.

Horn, B. K. P. and Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203.

Huang, D.-A., Ramanathan, V., Mahajan, D., Torresani, L., Paluri, M., Fei-Fei, L., and Niebles, J. C. (2018). What Makes a Video a Video: Analyzing Temporal Information in Video Understanding Models and Datasets. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7366–7375, Salt Lake City, UT. IEEE.

Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., and Brox, T. (2017). FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1647–1655, Honolulu, HI. IEEE.

Jayaraman, D. and Grauman, K. (2016). Slow and Steady Feature Analysis: Higher Order Temporal Coherence in Video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3852–3861.

Ji, S., Xu, W., Yang, M., and Yu, K. (2013). 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231.

Joe Yue-Hei Ng, Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., and Toderici, G. (2015). Beyond short snippets: Deep networks for video classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4694–4702, Boston, MA, USA. IEEE.

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., and Zisserman, A. (2017). The Kinetics Human Action Video Dataset. *arXiv:1705.06950*.

Kingma, D. P. and Ba, J. (2015). Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, San Diego, CA.

Klaeser, A., Marszalek, M., and Schmid, C. (2008). A Spatio-Temporal Descriptor Based on 3D-Gradients. In *British Machine Vision Conference (BMVC)*, pages 99.1–99.10, Leeds. British Machine Vision Association.

Koutnik, J., Greff, K., Gomez, F., and Schmidhuber, J. (2014). A Clockwork RNN. In *International Conference on Machine Learning (ICML)*, pages 1863–1871.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, Lake Tahoe, NV, USA.

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., and Serre, T. (2011). HMDB: A large video database for human motion recognition. In *International Conference on Computer Vision (ICCV)*, pages 2556–2563, Barcelona, Spain. IEEE.

Laptev and Lindeberg (2003). Space-time interest points. In *IEEE International Conference on Computer Vision (ICCV)*, pages 432–439 vol.1, Nice, France. IEEE.

Laptev, I., Marszalek, M., Schmid, C., and Rozenfeld, B. (2008). Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, Anchorage, AK, USA. IEEE.

Li, C., Zhong, Q., Xie, D., and Pu, S. (2019). Collaborative Spatiotemporal Feature Learning for Video Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7872–7881, Long Beach, CA. IEEE.

Li, F., Gan, C., Liu, X., Bian, Y., Long, X., Li, Y., Li, Z., Zhou, J., and Wen, S. (2017). Temporal Modeling Approaches for Large-scale Youtube-8M Video Understanding. *arXiv:1707.04555*.

Li, Z., Gavrilyuk, K., Gavves, E., Jain, M., and Snoek, C. G. (2018). VideoLSTM convolves, attends and flows for action recognition. *Computer Vision and Image Understanding*, 166:41–50.

Lin, J., Gan, C., and Han, S. (2019). TSM: Temporal Shift Module for Efficient Video Understanding. In *IEEE International Conference on Computer Vision (ICCV)*, pages 7083–7093, Seoul, Korea. IEEE.

Lin, R., Xiao, J., and Fan, J. (2018). NeXtVLAD: An Efficient Neural Network to Aggregate Frame-Level Features for Large-Scale Video Classification. In Leal-Taixé, L. and Roth, S., editors, *European Conference on Computer Vision Workshop (ECCVW)*, volume 11132, pages 206–218, Cham. Springer International Publishing.

Liu, X., Lee, J.-Y., and Jin, H. (2019). Learning Video Representations From Correspondence Proposals. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4273–4281, Long Beach, CA. IEEE.

Lu, Z.-L. and Sperling, G. (1995). The functional architecture of human visual motion perception. *Vision Research*, 35(19):2697–2722.

Meyer, S., Djelouah, A., McWilliams, B., Sorkine-Hornung, A., Gross, M., and Schroers, C. (2018). PhaseNet for Video Frame Interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 498–507.

Ng, J. Y.-H., Choi, J., Neumann, J., and Davis, L. S. (2018). ActionFlowNet: Learning Motion Representation for Action Recognition. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1616–1624, Lake Tahoe, NV. IEEE.

Ng, J. Y.-H. and Davis, L. S. (2018). Temporal Difference Networks for Video Action Recognition. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1587–1596, Lake Tahoe, NV. IEEE.

Oh, T.-H., Jaroensri, R., Kim, C., Elgharib, M., Durand, F., Freeman, W. T., and Matusik, W. (2018a). Learning-Based Video Motion Magnification. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *European Conference on Computer Vision (ECCV)*, volume 11208, pages 663–679, Cham. Springer International Publishing.

Oh, T.-H., Jaroensri, R., Kim, C., Elgharib, M., Durand, F., Freeman, W. T., and Matusik, W. (2018b). Learning-Based Video Motion Magnification. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science, pages 663–679, Cham. Springer International Publishing.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H., Larochelle, H., Beygelzimer, A., d\textquotesingle Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8026–8037, Vancouver, Canada. Curran Associates, Inc.

Qiu, Z., Yao, T., and Mei, T. (2017). Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 5534–5542, Venice. IEEE.

Ranjan, A. and Black, M. J. (2017). Optical Flow Estimation Using a Spatial Pyramid Network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2720–2729, Honolulu, HI. IEEE.

Sevilla-Lara, L., Liao, Y., Guney, F., Jampani, V., Geiger, A., and Black, M. J. (2017). On the Integration of Optical Flow and Action Recognition. *arXiv:1712.08416*.

Simonyan, K. and Zisserman, A. (2014). Two-Stream Convolutional Networks for Action Recognition in Videos. In *Advances in Neural Information Processing Systems (NIPS)*, page 9, Montreal, Quebec, Canada. Curran Associates, Inc.

Soomro, K., Zamir, A. R., and Shah, M. (2012). UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *arXiv:1212.0402*.

Srivastava, N., Mansimov, E., and Salakhutdinov, R. (2015). Unsupervised Learning of Video Representations using LSTMs. In *International Conference on Machine Learning (ICML)*, pages 843–852.

Stroud, J. C., Ross, D. A., Sun, C., Deng, J., and Sukthankar, R. (2018). D3D: Distilled 3D Networks for Video Action Recognition. *arXiv:1812.08249*, page 13.

Sun, D., Yang, X., Liu, M.-Y., and Kautz, J. (2018). PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, Salt Lake City, UT, USA. IEEE.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 3104–3112. Curran Associates, Inc.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). Learning Spatiotemporal Features with 3D Convolutional Networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, Santiago, Chile. IEEE.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. (2018). A Closer Look at Spatiotemporal Convolutions for Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6450–6459, Salt Lake City, UT. IEEE.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). WaveNet: A Generative Model for Raw Audio. *arXiv:1609.03499*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is All you Need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 5998–6008. Curran Associates, Inc.

Wadhwa, N., Rubinstein, M., Durand, F., and Freeman, W. T. (2013). Phase-based video motion processing. *ACM Transactions on Graphics*, 32.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K. (1989). Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(3):328–339.

Wang, H., Klaser, A., Schmid, C., and Liu, C.-L. (2011). Action recognition by dense trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3169–3176, Colorado Springs, CO, USA. IEEE.

Wang, H. and Schmid, C. (2013). Action Recognition with Improved Trajectories. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3551–3558, Sydney, Australia. IEEE.

Wang, H., Tran, D., Torresani, L., and Feiszli, M. (2019a). Video Modeling with Correlation Networks. *arXiv:1906.03349*.

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., and Van Gool, L. (2016). Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *European Conference on Computer Vision (ECCV)*, pages 20–36, Cham. Springer International Publishing.

Wang, X., Girshick, R., Gupta, A., and He, K. (2018). Non-local Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7794–7803, Salt Lake City, UT, USA. IEEE.

Wang, X., Jabri, A., and Efros, A. A. (2019b). Learning Correspondence From the Cycle-Consistency of Time. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2566–2576, Long Beach, CA. IEEE.

Wang, X., Wang, L., and Qiao, Y. (2013). A Comparative Study of Encoding, Pooling and Normalization Methods for Action Recognition. In Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M. Y., Weikum, G., Lee, K. M., Matsushita, Y., Rehg, J. M., and Hu, Z., editors, *Asian Conference on Computer Vision (ACCV)*, volume 7726, pages 572–585, Berlin, Heidelberg. Springer Berlin Heidelberg.

Wang, Y., Long, M., Wang, J., and Yu, P. S. (2017). Spatiotemporal Pyramid Network for Video Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2097–2106, Honolulu, HI. IEEE.

Wei, D., Lim, J. J., Zisserman, A., and Freeman, W. T. (2018). Learning and Using the Arrow of Time. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8052–8060, Salt Lake City, UT. IEEE.

Willems, G., Tuytelaars, T., and Van Gool, L. (2008). An Efficient Dense and Scale-Invariant Spatio-Temporal Interest Point Detector. In Forsyth, D., Torr, P., and Zisserman, A., editors, *European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science, pages 650–663, Berlin, Heidelberg. Springer.

Wu, C.-Y., Zaheer, M., Hu, H., Manmatha, R., Smola, A. J., and Krahenbuhl, P. (2018). Compressed Video Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6026–6035, Salt Lake City, UT. IEEE.

Wu, H.-Y., Rubinstein, M., Shih, E., Guttag, J., Durand, F., and Freeman, W. (2012). Eulerian video magnification for revealing subtle changes in the world. *ACM Transactions on Graphics*, 31.

Xie, S., Sun, C., Huang, J., Tu, Z., and Murphy, K. (2018). Rethinking Spatiotemporal Feature Learning: Speed-Accuracy Trade-offs in Video Classification. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *European Conference on Computer Vision (ECCV)*, volume 11219, pages 318–335, Cham. Springer International Publishing.

Xu, J., Ranftl, R., and Koltun, V. (2017). Accurate Optical Flow via Direct Cost Volume Processing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5807–5815, Honolulu, HI. IEEE.

Yang, C., Xu, Y., Shi, J., Dai, B., and Zhou, B. (2020). Temporal Pyramid Network for Action Recognition. In *arXiv:2004.03548*.

Yu, F. and Koltun, V. (2016). Multi-Scale Context Aggregation by Dilated Convolutions. In *International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico.

Zach, C., Pock, T., and Bischof, H. (2007). A Duality Based Approach for Realtime TV-L 1 Optical Flow. In Hamprecht, F. A., Schnörr, C., and Jähne, B., editors, *Pattern Recognition*, volume 4713, pages 214–223. Springer Berlin Heidelberg, Berlin, Heidelberg.

Zhang, D., Dai, X., and Wang, Y.-F. (2018). Dynamic Temporal Pyramid Network: A Closer Look at Multi-scale Modeling for Activity Detection. In Jawahar, C., Li, H., Mori, G., and Schindler, K., editors, *Asian Conference on Computer Vision (ACCV)*, Lecture Notes in Computer Science, pages 712–728, Cham. Springer International Publishing.

Zhao, J. and Snoek, C. G. M. (2019). Dance With Flow: Two-In-One Stream Action Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9935–9944, Long Beach, CA. IEEE.

Zhao, Y., Xiong, Y., and Lin, D. (2018a). Recognize Actions by Disentangling Components of Dynamics. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6566–6575, Salt Lake City, UT. IEEE.

Zhao, Y., Xiong, Y., and Lin, D. (2018b). Trajectory Convolution for Action Recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, page 12, Montreal, Quebec, Canada.

Zhou, B., Andonian, A., Oliva, A., and Torralba, A. (2018a). Temporal Relational Reasoning in Videos. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *European Conference on Computer Vision (ECCV)*, volume 11205, pages 831–846, Cham. Springer International Publishing.

Zhou, Y., Sun, X., Zha, Z.-J., and Zeng, W. (2018b). MiCT: Mixed 3D/2D Convolutional Tube for Human Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 449–458, Salt Lake City, UT. IEEE.

Zhu, C., Tan, X., Zhou, F., Liu, X., Yue, K., Ding, E., and Ma, Y. (2018a). Fine-Grained Video Categorization with Redundancy Reduction Attention. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *European Conference on Computer Vision (ECCV)*, volume 11209, pages 139–155, Cham. Springer International Publishing.

Zhu, X., Xiong, Y., Dai, J., Yuan, L., and Wei, Y. (2017). Deep Feature Flow for Video Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4141–4150, Honolulu, HI. IEEE.

Zhu, Y., Lan, Z., Newsam, S., and Hauptmann, A. (2018b). Hidden Two-Stream Convolutional Networks for Action Recognition. In Jawahar, C. V., Li, H., Mori, G., and Schindler, K., editors, *Asian Conference on Computer Vision (ACCV)*, volume 11363, pages 363–378, Cham. Springer International Publishing.

Zhu, Y. and Newsam, S. (2018). Random Temporal Skipping for Multirate Video Analysis. In Jawahar, C. V., Li, H., Mori, G., and Schindler, K., editors, *Asian Conference on Computer Vision (ACCV)*, volume 11363, pages 542–557, Cham. Springer International Publishing.

Zou, W., Zhu, S., Yu, K., and Ng, A. Y. (2012). Deep Learning of Invariant Features via Simulated Fixations in Video. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 3203–3211. Curran Associates, Inc.

CURRICULUM VITAE

Qili Zeng

Education

M.S. in Computer Science *2018 - 2020*
Boston University, Boston, United States
Thesis: *Learning Temporal Variations for Action Recognition*
Thesis Advisor: Professor Janusz Konrad

B.Eng. in Information Engineering *2014 - 2018*
Southeast University, Nanjing, China
Thesis: *Real-Time Head Pose Estimation System*
Thesis Advisor: Professor Xin Geng

Experience

- **Research Assistant** *2019 - 2020*
  Visual Information Processing Lab, BU
  Action Recognition
  Advisor: Prof. Janusz Konrad

- **Research Intern** *2018*
  Youtu Lab, Tencent Corporation
  Temporal Action Localization
  Mentor: Dr. Ji Wang

- **Research Intern** *2017*
  Institute of Computing Technology, Chinese Academy of Sciences
  Person Re-identification
  Advisor: Prof. Hong Chang

- **Research Assistant** *2016*
  National Mobile Communications Research Laboratory, SEU
  Object Tracking
  Advisor: Prof. Ming Chen

- **Research Intern** *2015 - 2016*
  Institute of RF- & OE-ICs, SEU
  Parameter Extraction and Signal Modeling of GaN HEMT
  Advisor: Prof. Fengyi Huang