Efficient Modelling Across Time of Human Actions and Interactions

Alexandros Georgios Stergiou

Efficient Modelling Across Time of Human Actions and Interactions

Efficiënte modellering door de tijd heen van menselijke acties en interacties

(met een samenvatting in het Nederlands)

Proefschrift

ter verkrijging van de graad van doctor aan de
Universiteit Utrecht
op gezag van de
rector magnificus, prof.dr. H.R.B.M. Kummeling,
ingevolge het besluit van het college voor promoties
in het openbaar te verdedigen op

maandag 27 september 2021 des ochtends te 10.15 uur

door

Alexandros Georgios Stergiou

geboren op 15 september 1995
te Thessaloniki, Griekenland

Promotor:
Prof. dr. R.C. Veltkamp

Copromotor:
Dr. R.W. Poppe

Acknowledgements

Four year ago I was accepted to work under the supervision of Ronald Poppe and Remco Veltkamp on dyadic human interactions. Over the course of these years, they have given me the opportunity to develop my own research ideas throughout my PhD. I was encouraged to expand upon research problems within the scope of my project and to further grow as a researcher with independent ideas. I would like especially thank Ronald not only for his academic support and overall enthusiasm in my work, but also for his efforts and the lengths that he has gone through for creating a welcoming environment for me and helping me to get where I am right now. Your occasional barging in the office will be missed but hopefully I will be able to replicate it myself in my next steps.

I would like to warmly thank all of my committee members for the time that they have dedicated as committee members for my thesis.

Thanks to my colleagues, friends and co-authors Georgios Kapidis, Grigorios Kalliatakis and Christos Chrysoulas for their valuable help and collaboration in our joint work on spatio-temporal feature examinations, as well as our discussions during my PhD. Thank you George for being a co-contributor to Chapter 7 and for all the great times that we had both in and outside the office. Gregory and Christos, thank you for including me in your discussions and being part of our own group back in Essex. Our discussions have given me both insights and a better understanding of research and working in academia. I am especially looking forward for our collaboration in future projects.

Thank you to my mother Glyka and my father Elias for being by my side throughout my life and academic journey. From the first days of my bachelor degree to this day, their support has been extremely valuable. Their sacrifices and hard work has been an inspiration for me to better myself.

Last, but not least, I am thankful for to my partner Deborah for her positivity, patience and unconditional love. She has been a rock to lean on and has made me deeply grateful of unplanned meetings in unanticipated places.

List of Papers

Included in the thesis

Stergiou, Alexandros and Poppe, Ronald. "Multi-Temporal Convolutions for Human Action Recognition in Videos". In: *International Joint Conference of Neural Networks (IJCNN)*. IEEE. 2021

Stergiou, Alexandros and Poppe, Ronald. "Learn to cycle: Time-consistent feature discovery for action recognition". In: *Pattern Recognition Letters* vol. 141 (2021), pp. 1–7

Stergiou, Alexandros, Poppe, Ronald, and Veltkamp, Remco C. "Learning Class-Specific Features with Class Regularization for Videos". In: *Applied Sciences* vol. 10, no. 18 (2020), p. 6241

Stergiou, Alexandros et al. "Class Feature Pyramids for Video Explanation". In: *International Conference on Computer Vision Workshop (ICCVW)*. IEEE. 2019, pp. 4255–4264

Stergiou, Alexandros et al. "Saliency tubes: Visual explanations for spatio-temporal convolutions". In: *International Conference on Image Processing (ICIP)*. IEEE. 2019, pp. 1830–1834

Stergiou, Alexandros and Poppe, Ronald. "Analyzing human-human interactions: A survey". In: *Computer Vision and Image Understanding* vol. 188 (2019), p. 102799

Other works

Stergiou, Alexandros, Poppe, Ronald, and Grigorios, Kalliatakis. "Refining activation downsampling with SoftPool". In: *International Conference on Computer Vision (ICCV)*. IEEE. 2021

Stergiou, Alexandros. "The Mind's Eye: Visualizing Class-Agnostic Features of CNNs". In: *International Conference on Image Processing (ICIP)*. IEEE. 2021

Stergiou, Alexandros and Poppe, Ronald. "Spatio-Temporal FAST 3D Convolutions for Human Action Recognition". In: *International Conference on Machine Learning Applications (ICMLA)*. IEEE. 2019, pp. 1830–1834

Contents

Samenvatting

Dit proefschrift richt zich op het analyseren van menselijke handelingen en interacties in video's. We beginnen met het identificeren van de belangrijkste uitdagingen met betrekking tot actieherkenning in video's en bekijken in hoeverre deze worden aangepakt met de huidige methoden.

Op basis van deze uitdagingen, en door ons te concentreren op het temporele aspect van acties, stellen we dat de huidige spatio-temporele kernels met een vaste grootte in 3D convolutionele neurale netwerken (CNNs) kunnen worden verbeterd om beter om te gaan met temporele variaties in de invoer. Onze bijdragen zijn gebaseerd op het vergroten van de convolutionele receptieve velden door het gebruik van ruimtelijk-temporele segmenten van video's die variëren in grootte, en daarnaast het bepalen van de lokale relevantie van de functie in relatie tot de gehele video. De zo geëxtraheerde functies bevatten informatie die het belang van lokale functies omvat over meerdere tijdsduren, waaronder de gehele video.

Vervolgens bestuderen we hoe we variaties tussen actieklassen beter kunnen aanpakken, door hun karakteristieke patronen over verschillende lagen van de architectuur te versterken. Met de hiërarchische extractie van kenmerken worden variaties van relatief vergelijkbare klassen op dezelfde manier gemodelleerd als zeer verschillende klassen. Daardoor is het minder waarschijnlijk dat onderscheid tussen vergelijkbare klassen effectief wordt gemodelleerd. De voorgestelde aanpak regulariseert feature maps door features te versterken die overeenkomen met de klasse van de video die wordt bekeken. We stappen af van klasse-agnostische netwerken en doen vroege voorspellingen op basis van dit functieversterkingsmechanisme.

De voorgestelde vernieuwingen zijn geëvalueerd op verschillende benchmark-datasets voor actieherkenning en we laten daar competitieve resultaten op zien. Op het gebied van prestaties concurreren we met de nieuwste algoritmes terwijl onze methoden efficiënter zijn wat betreft het aantal GFLOPs.

Ten slotte presenteren we een voor mensen begrijpelijke benadering die is gericht op het bieden van visuele verklaringen voor functies die zijn geleerd via spatio-temporele netwerken. We isoleren spatio-temporele regio's in 3D-CNN's die informatief zijn voor een actieklasse. We breiden deze aanpak uit om de hele netwerkarchitectuur mogelijk te maken, incrementeel kernels met verschillende complexiteiten te ontdekken en lagen met betrekking tot een specifieke klasse te modelleren.

Abstract

This thesis focuses on video understanding for human action and interaction recognition. We start by identifying the main challenges related to action recognition from videos and review how they have been addressed by current methods.

Based on these challenges, and by focusing on the temporal aspect of actions, we argue that current fixed-sized spatio-temporal kernels in 3D convolutional neural networks (CNNs) can be improved to better deal with temporal variations in the input. Our contributions are based on the enlargement of the convolutional receptive fields through the introduction of spatio-temporal size-varying segments of videos, as well as the discovery of the local feature relevance over the entire video sequence. The resulting extracted features encapsulate information that includes the importance of local features across multiple temporal durations, as well as the entire video sequence.

Subsequently, we study how we can better handle variations between classes of actions, by enhancing their feature differences over different layers of the architecture. The hierarchical extraction of features models variations of relatively similar classes the same as very dissimilar classes. Therefore, distinctions between similar classes are less likely to be modelled. The proposed approach regularises feature maps by amplifying features that correspond to the class of the video that is processed. We move away from class-agnostic networks and make early predictions based on feature amplification mechanism.

The proposed approaches are evaluated on several benchmark action recognition datasets and show competitive results. In terms of performance, we compete with the state-of-the-art while being more efficient in terms of GFLOPs.

Finally, we present a human-understandable approach aimed at providing visual explanations for features learned over spatio-temporal networks. We isolate spatio-temporal regions in 3D-CNNs that are informative for an action class. We extend this approach to allow for the traversal over the entire network architecture, incrementally discovering kernels at different complexities, and modelling layers related to a specific class.

Chapter 1
Introduction

Many videos depict people. Their actions inform us of their activities, in relation to objects and other people, as well as the cultural and social setting. Research has aimed at automating the recognition of human actions in videos. In this chapter, we provide an overview of the challenges associated with the task of human action recognition from videos. We subsequently present the structure of this thesis by discussing the contents of each chapter.

1.1 Challenges in video understanding

Based on how information is processed for images and for temporal sequences in videos, we identify two main groups of challenges.

The first group of challenges is associated with visual changes. These changes can be based on differences in the environment, in which human actions or interactions are performed, as well as the recording settings for videos. Significant variations in terms of these conditions, and low correspondence to previously processed examples, can significantly increase the difficulty of recognising the actions and interactions performed. Most notably, the viewpoint of the camera has a large effect on how actions and interactions are perceived. The lack of stereo in common video data presents difficulties in the recognition of objects in the background and foreground. Difficulties may also arise because of occlusion of the actors or their body parts, especially in scenes that include physical interaction with objects or other actors. This presents obstacles in the recognition of human actions and interactions from a single viewpoint, as characteristic movements or poses of the key body parts may not be visible. Considering that movement is present in videos, camera motion and motion blur may introduce an additional degree of difficulty for the correct recognition of actions under different settings. Other visual-based challenges include variations such as those based on luminance. Examples include lighting from multiple directions, or the reflective capabilities of material that determine how much light is absorbed and how much is reflected under which direction. In addition, lighting conditions can impact the colour consistency in video frames.

The second set of variations in human actions in videos is based on differences in the performance of the actions themselves. This is strongly associated with the human factor. As actions can be identified in many different ways, the criteria used to describe actions remain unclear. Vallacher and Wegner [268] have created a broad system to explain the formation of relationships between personality traits and how individuals perform everyday actions. The foundation of their proposed *theory of action identification* is based on three principles. The first is that human actions are maintained with respect to the prepotent identity associated with the action. Prepotent identities can, for example, correspond to one person sustaining the act of passing the ball, by projecting the exact location that the receiver will catch it. This can also be sustained by a less experienced passer, as “throwing

the ball within the general proximity of the receiver". We can find that identities serve a guideline role for the action performed. The first point that we can draw from the passer example concerns the levels of complexity that the same action can be expressed as. The second principle states that the higher the level of identity, the more likely it is to become prepotent. Low identities include more general action identifications such as "moving hands" that is a component within a very large number of actions.

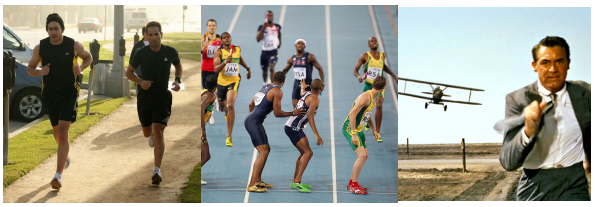
However, actions can be better understood with the use of higher-level identities such as "throwing ball to teammate", "making a corner pass" or "pass over the defending team". This however, leads to the second point that can be drawn from the example, which is that the actor's skill level can constrain his action identity creation ability. The third principle states that actions which cannot be performed by a high-level prepotent, tend to then be expressed by a lower-level identity. This is additionally connected to the individual's skills (or lack of thereof), fears or abstractions. For example, high-level identities such as "full court pass to teammate" are strongly associated with the above mentioned constraints. Instead, the person is more willing to adopt an identity such as "throw ball towards the baseline" in order to control the action. Considering how action and interaction identities are strongly connected with an individual's prepotent identity associated with that action, a great problem arises for the definition of action classes. We demonstrate in Figure 1.1 three examples that are under the same class *"basketball pass"*.

Figure 1.1. Examples of basketball passes from the same game. (A) Brief hand-off pass, (B) Wing pass, (C) Full-court outlet pass.

Understanding actions is also based on their socially conveyed meanings and the identities that have been assigned to them. For example, considering the examples in Figure 1.2, actions can vary based on the scenarios that they are performed in. An activity such as *throwing* an object relates differently in different action scenes. *Throwing* an empty bottle serves the purpose of getting rid of that object, while *throwing* a ball serves the purpose of scoring a point. In the volleyball example, other people are also important for understanding the context as throwing ball is perceived differently if the receiver is a teammate or a player of the opposite team. In the same sense, it can also be part of a playful interaction without a competitive nature (e.g. throwing a frisbee to a dog). The relations of actions to other people or objects is also important and should be considered. Given the action of *running*, the correlation to other people and objects in the immediate environment can be telling for the context that the action is performed in. There is the

possibility that the action is performed for an exercise or in a social manner, based on the accompanying people. Alternatively, the nature of the activity can be within the context of a race and thus, the participants competing against each other. The action could also be engaged in for completely different reasons as seen in the third example in Figure 1.2. Another important aspect of actions is their cultural meanings while being either general or specific. An example of this can see seen in the different forms of *dancing*. *Dancing* can be associated with a social occasion or being part of a tradition. In addition to culture, social settings can also change the interpretation of different actions. For example, a *handshake* could be perceived differently. In relation to the settings, a *handshake* could be a formal way of two people introducing each other. Within the context of a competition, a handshake could be a form of resignation as observed in chess or could be for congratulating the winner or receiver of an award.

What does the activity of ***"throwing"*** correspond to in each scenario ?



What is the relationship between ***"running"*** *and different people or objects* ?

What are the characteristics of ***"dancing"*** considering different cultures ?

How is ***"handshake"*** perceived under different social settings?

Figure 1.2. Instances of human actions. Actions can be informative in terms of the activities that are presented (e.g. *"throwing"* activity). Their relations to people and object may differ between examples (e.g. *"running"* action). There is a cultural (e.g. *"dancing"* action) or social (e.g. *"handshake"* action) aspect in terms of their characteristics.

The personal agency of each person represents significant challenges in the performance of actions. These factors and aspects describe the *how* and the *why* actions are performed in a specific manner. Through the inclusion of these factors a more distinct identity of each action can be perceived. All traits are important, as most actions are personally determined by individuals, there is a direct connection between these traits and the performance of actions.

1.2 Thesis overview

In this section, we provide an outline of the structure of the thesis and the action recognition challenges that each chapter addresses. Chapter 2 gives an overview of how current action recognition methods in computer vision address human actions, while the corresponding datasets that have been collected for action recognition and video understanding tasks are summarised in Chapter 3. Our main contributions are introduced in Chapters 4 to 7 while a discussion is provided in Chapter 8.

Chapter 2, Related Work. We detail a synopsis of current progress made in action recognition. We distinguish between approaches that have been based on hand-crafted features and methods that used learned features through optimisation. We then present the main motivation for the works that have been included in this thesis. We discuss how our approaches differe from previous efforts and the challenges that our methods address.

Chapter 3, Datasets for Video Understanding. We provide an overview of historic and current datasets used as action recognition benchmarks. We focus on datasets that exemplify milestones achieved in terms of data collection and increases in the data complexity. We then present the datasets that are used in this thesis.

Chapter 4, Improving Action Recognition through Time-Consistent Features. We present a novel method to address variations in terms of the performances of actions. Specifically, we target variations in the durations of different motions of the action. We explore how the locality of spatio-temporal convolutional patterns can be extended in order to address the relevance of the local features within the context of the entire action sequence. The newly created feature volumes encapsulate these dynamics of the local features with respect to the entire video. Through the temporal attention of the most relevant features, we are able to strengthen informative signals. This is done through a recurrent sub-network with the *Squeezed and Recursion* part of the method. The second part of the proposed method studies the cyclic consistency of the original and the global attention-aligned features. By measuring the distance in a common embedding space between the two feature volumes, a similarity measure between the two volumes can be established. Based on that similarity measure, we propose a *Temporal Gating* function that can hold a closed or open state. In a closed state the similarity between the two feature volumes is not substantial and thus information is processed independently. Instead, in an open state, the gate fuses the two volumes to create a single coherent feature volume that highlights with temporal attention the features that are informative across the entire video. We show that this inclusion of temporal attention can improve the performance of spatio-temporal action recognition models with a minimal increase in the total number of computations required.

Chapter 5, Time-Varying Convolutions for Video Understanding. We further study the extraction of features that correspond to action patterns based on different temporal durations. This chapter focuses on the representation of temporal variations in the performance of actions that are not based in the order of magnitude. We propose the extraction of features over varied size spatio-temporal windows. Through this, short-term and long-term action patterns can be better modelled as convolutional receptive fields that are less strict to fixed-sized regions. The proposed convolutional blocks named *Multi-Temporal Convolutions* utilise three branches. Each branch addresses different temporal durations over an interval. The first branch is the *local* branch of small spatio-temporal regions. The *prolonged* branch similarly models information within enlarged spatio-temporal windows while the information also relates to the features that have been extracted in the local branch. The final *global aggregated feature importance* branch uses Squeeze and Recursion from Chapter 4 to align the features from the previous two branches and to discover their relevance within the entire video sequence. Experiments demonstrate competitive results over common benchmark action recognition datasets with an additional substantial reduction in terms of the number of computational operations used. We ablate over the methods to create feature volumes for the extraction of longer spatio-temporal sizes as well as the proportions of local and prolonged features that can be extracted.

Chapter 6, Class-Specific Regularisation Across Time. We address the association of features to specific action classes. As features in current convolutional

models are learned in a hierarchical fashion, only a small set of features targets the modelling of spatio-temporal patterns that are specific to a class or a smaller sub-set of action classes. By associating class and feature information, descriptors based on class-relative information can be created. The goal of the proposed approach, named *Class Regularisation*, is to create descriptors that can amplify features specific to a class. Through this, subtle differences in the visual appearance and performance between similar classes can be intensified in order to create a distinction between them.

Chapter 7, Spatio-Temporal Feature Interpretation. We explore features that are learned by spatio-temporal convolutions. The main objective for this chapter is to study feature explanation as a qualitative measure for the representation of a model's performance. We first target the task of uncovering the spatio-temporal regions that are informative for an action that is performed and that 3D CNNs rely for their predictions. This is visualised as a saliency mask over time, creating a tube effect that corresponds to the space-time regional attention of the network. Our approach uses the class weights associated with the class and its features that are to be visualised alongside the respective feature activations produced over space-time locations. Based on the representation of class feature relevance over the video sequence, we also extend the approach to work hierarchically over features of multiple layers. Through a proposed cross-layer feature relevance exploration named *back step*, we can construct an association between high-level and lower-level features over different network layers. This effectively deals with the curse of dimensionality problem of traversing over different layers inside the network while also presenting their causalities. The resulting few-to-many connections create a structure that resembles a pyramid. We demonstrate the merits of our method through providing visual insights into the spatio-temporal features of each layer that correspond to the action class.

Chapter 8, Discussion and Future Research Directions. We conclude with Chapter 8 in which we summarise the methods that have been proposed in this thesis. Additionally, we discuss prominent research directions and the challenges that such future works could address.

Chapter 2

Related Work

In this chapter we discuss the recent progress that has been made in the field of action recognition and video understanding. We start by reviewing methods that utilise hand-crafted features for the extraction of spatio-temporal information in combination with a classifier model. We then overview methods that use objective tasks to discover descriptive features. We refer to the product of these descriptors as learned-features.

2.1 Recognition from hand-crafted features

Traditionally, the recognition of human actions and interactions from video starts with the extraction of image features to represent the scene. Subsequently, the extracted spatial-features are used for the classification of the video sequence with an action label. An important requirement in this process is that the extracted image features must be invariant to image conditions and the action performance. These conditions are in combination with maintaining a sufficient feature complexity to deal with subtle differences between classes.

We make a distinction between approaches that are based on local features utilising salient points in the video sequence, and approaches based on feature templates that take into account regions that roughly correspond to a person's body or body parts.

2.1.1 Local features approach

In general, local feature methods use a bottom-up approach by first detecting points of interest in a video, and then aggregate the detected regions over time and space to represent the performed actions. This selection of points is performed locally, typically at edges or motion boundaries. Popular descriptors are based on Harris corners [160, 311], SIFT [50, 156] or optical flow [307]. Typically, no direct correspondence between points and video actors or their body part exists. Consequently, factors such as camera motion, dynamic backgrounds and obstructions affect the presence of local features.

When additional depth information is available, e.g. from RGB-D recordings, local features can take into account depth gradients [141]. By utilising multiple viewpoints for the efficient mapping of 3D points, Xia and Aggarwal [296] considered the creation of a codebook based on depth sequences.

To increase the robustness of local descriptors, the distribution of points can be described based on a bag-of-words (BoW) or Fisher Vectors (FV) [72, 183]. The consideration based on the use of local descriptors is that instances of the same action class are to have similar descriptors. Improving on this and to allow a more complex feature distribution, Niebles *et al.* [180] constructed a vocabulary using latent topics models.

As an alternative to modelling the trajectories of individual points in order to capture the motion of local points, other works focused on the temporally sequential nature of human actions and interactions through modelling the changes in the distribution of interest

points over time. Zhang *et al.* [315] used spatio-temporal phases to create a histogram of bag-of-phases. An example of this is visualised in Figure 2.1. Each phase is composed of local words with specific ordering and spatial position. Instead of jointly mapping both dimensions, others have addressed separation as well [225, 264]. The computed histograms represent similar features in single or multiple frames. Histograms of visual words have also been utilised by Kong *et al.* [126] with the words derived from the quantisation of the spatial-temporal descriptors clustered. The produced clusters form high-level representations, termed interactive phases. These phases include motion relationships such as the shaking of two hands. This idea has been extended to localise interactions by spatially clustering phrases [263]. Prabhakar and Rehg [198] aimed at the inclusion of variation in the temporal domain by modelling the causality of the occurrence of visual words.

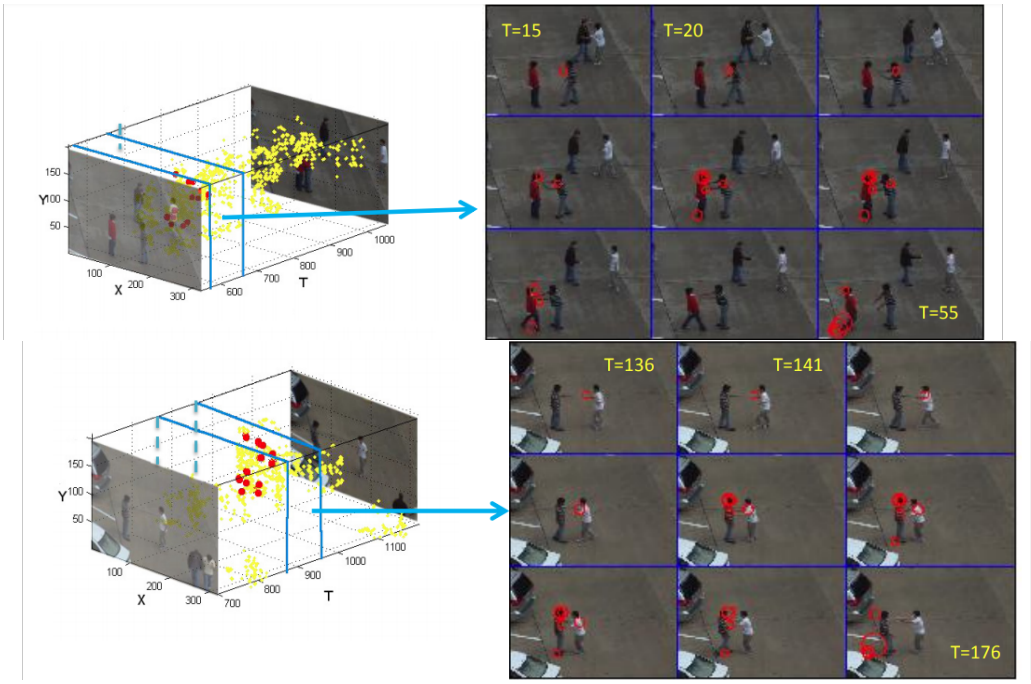


Figure 2.1. The red points are discovered co-occurring features from Zhang *et al.* The spatio-temporal phases are computed based on the causality between actor body parts. These causal relationships can represent relationships between actors over extended time durations. Figure sourced from [315].

As not all motions and attributes are informative, Kong *et al.* [127] considered only body parts that are characteristic for a specific class. Their method pools BoW responses in a coarse grid, thus allowing them to identify specific motion patterns relative to a person's location. A limit to the level of detail is the granularity of the patches and the accuracy of the person detector. Subsequent detections are linked to trajectories in order to additionally account for the temporal dimensionality of the video sequences. Mohammadi *et al.* [168] extend this grid-based approach by grouping the motion patterns as BoW vectors. Similarly, Turchini *et al.* [266] used trajectories of multiple local feature types in order to localise where actions are performed. Wang and Cordelia [280] have introduced Improved Dense Trajectories (DT), a widely adopted way of finding and describing trajectories of points. Dense Trajectories encode points as the temporal-based combination of Histograms of Oriented Gradients (HOG), Histograms of Oriented Flow (HOF) and Motion Boundary Histograms (MBH).

Other approaches have been based on the use of local features to initially isolate the actors in videos. The local features can be endings of HOG and HOF descriptors as persented by Caba *et al.* [24]. Through the localisation of a person, the context of motions and actions of other people in relation to that actor can provide useful cues for the understanding of the scene. Reddy and Shah [206] used information obtained through a scene context descriptor which combines the location and surroundings extracted with optical flow and 3D-SIFT, based on the moving and stationary pixels. Cho *et al.* [41] introduced a descriptor that takes into account the local, global and individual movement of actors in video sequences. By linking local features to persons, descriptions for the actor surroundings can also be made. Lan *et al.* [134] presented an Action Context (AC) descriptor that is based on connected action probability vectors of several people. Similarly, Choi *et al.* [42] performed joint tracking, classification of the actions of an individual and the recognition of collective activities by considering bounding boxes of extracted local features.

2.1.2 Template-based approaches

In a single frame, HOG descriptors can represent characteristic poses. For example, a high-five interaction can be described as two people facing each other with outstretched hands that meet above their heads. This notion was adopted by Bourdev *et al.* [20] to detect people engaged in specific actions, and was applied to human-human interactions by Raptis and Sigal [205]. Sefidgar *et al.* [219] formulated descriptors with discriminative key frames and their relative distance and timing within the interaction. Alternatively, Sener and Ikizler-Cinbis [221] formulated interaction detection as a multiple-instance learning problem to focus on relevant frames. This was done as not all frames in a video sequence are considered informative.

Based on HOG spatial feature descriptors, the motion around a characteristic pose can provide complementary information. Van Gemeren *et al.* [75] combined HOG and HOF descriptors to encode the characteristic frame of a two-person interaction. Yu and Yang [306] concatenated HOG and HOF descriptors and applied FV to make the detection linearly separable. This allowed the concurrent utilisation of spatial and temporal information by the model. Mousavi *et al.* [172] introduced summarised tracked local key-point features through the Histogram of Oriented Tracklets (HOT). While optical flow can be seen as the representation of the motion between two subsequent frames, tracklets describe the path of local key-points over longer time intervals.

Temporal patches can be used in combination with local descriptors to take advantage of potentially conjoint salient pose or motion information [110]. Yin *et al.* [303] employ 3D-SIFT to describe local motion events, but used a HOF to model the global motion of the video sequence. Similarly, Lathuiliere *et al.* [138] combined HOG descriptors and trajectory information from linked local features. Single-person and two-person interaction attributes, such as “two persons are standing side-by-side”, were calculated from these features.

A different approach to interest points is to detect faces or bodies using a generic face or body detector [190, 214]. Given two close detections, actions and interactions can be classified based on extracted features within the detection region [214]. Various attributes, including gross body movement and proximity, have been employed to classify the interaction. Patron *et al.* [190] also include the relative size and orientation of each person. Khodabandeh *et al.* [122] consider clusters of similar frames based on proximity and appearance of pairs of people. They included user feedback in order to improve the

purity of the clusters and thus the classification of them. The drawback of this two-stage approach is that classification is sub-optimal when the person localisation fails, for example when people partly obstruct each other. This is a common situation, especially in cases where multiple actors are in close proximity.

Such situations are mitigated with the use of Deformable Parts Models (DPMs) [66]. In DPMs, an articulated object such as a person or multiple people interacting are modelled as a set of parts with the inclusion of deformations between them. This provides a degree of flexibility in the spatial layout of the parts. As such, parts that are generally well detected, e.g. a person's head, can be coupled with parts that are traditionally more challenging to detect, such as a lower arm. Lu *et al.* [157] used DPMs to localise the rough outline of a person with optical flow being then employed to discover the motions performed in the subsequent frames. The resulting volume is then segmented into supervoxels to refine the person's outline in each frame, and classify the action. Instead of encoding the orientation of (pairs of) limbs as poselets, DPMs can also include a larger number of articulations by using a mixture of parts [300]. This approach has been used to describe the joint poses of two interacting people [299]. Hoai and Zisserman [95] extend the model to account for deformations in scale by using template examples of an action class and compare their similarity to the input video then used it to rank the detection scores. The main constraint of this method was the emphasis towards upper body movement based on the videos used.

In order to include additional degrees of variations based on the temporal performance of interactions, works have introduced a variety of methods. Ji *et al.* [109] modelled the changes in HOG descriptors over time using a Hidden Markov Model (HMM) for human interactions. With the use of the distance between actors, they consider the frames in the start, middle or end stage of the interaction. The HMM scores for all stages are fused for the final classification. The same rationale of using phases has been adopted by Cao *et al.* [26] for human activities that addressed the task of classifying a video sequence with potentially missing frames.

While DPMs only encode a particular pose or motion spatially, extensions for the inclusion of temporal information were proposed to deal with the time-varying nature of actions. Yao *et al.* [301] focused on human-object interactions capturing the movement related to a key pose using a DPM and a linked set of motion templates that also correspond to different phases of the performance. Tian *et al.* [256] have extended DPMs for action detection to model changes in pose over time. These formulations work well for the representation of coarse movements, but finer-scale movements are difficult to model because the motion is not linked to specific parts of the body. Tran and Yuan [262] also address a localisation task but consider linking regions over time based on HOG and HOF in a structured learning approach. A max-path algorithm is used to find the optimal volume that contains the action in space and time.

2.2 Action recognition from learned features

The hand-coded feature descriptors described in Section 2.1 focus on local or global spatial or spatio-temporal information. The manual selection of descriptors leaves room for improvement because the process is agnostic to the specific classification task, application domain or class of behaviours.

With the introduction of multiple convolutions [139], Convolutional Neural Networks (CNNs or ConvNets) have been used for the classification across different modalities. CNNs allow for the discovery of informative patterns through updates based on the error of

the task, e.g. predicting the action label. Consequently, they can overcome the issue of sub-optimal feature selection. While multiple convolution kernels allow for the selection of a wide range of image or video features, the stacking of consecutive convolution operations allows for a hierarchical extraction of complex features [230]. Typically, the characteristics extracted in the first layers of the network correspond to features of limited complexity such as edges and simple textures. The complexity of the features increases in relation to the network depth.

Methods based on neural networks have shown notable improvements for video understanding tasks as well. The benefit of improving accuracy based on large datasets, allows the architectures to generalise their feature assumptions over multiple instances, rather than being limited to the modelling capabilities of a predefined set of features, as in the hand-crafted methods.

The purpose of this section is to present neural network architectures for human action recognition. We then show how temporal information is modelled and incorporated in the convolutions.

2.2.1 Networks with individually-processed frames

Initial works with CNNs for action recognition have been based on the use of single frames [3, 15, 83]. Similar to the use of hand-crafted features, several methods proposed possible extensions to additionally incorporate temporal information.

Based on the classification of individual frames, Karpathy *et al.* [118] proposed three techniques to fuse the scores of multiple frames using different convolutional configurations. In the Early Fusion strategy, the input of the network is a stack of subsequent frames. Late Fusion combines the convolutional features of the first and last frames of a sequence in the final, fully connected layers. Slow Fusion is a combination of these two approaches, that empowers a progressive fusion over frames and activation maps, with the extension of convolutional layer connections through time. All three approaches are limited in their capability to deal with subtle temporal variations between classes, and large intra-class variations. It is a challenge to deal with these variations as they have to be modelled from a typically modest number of training videos.

To partly mitigate this issue, authors have investigated the use of Transfer Learning [14, 13, 32, 185, 304] from large image datasets. This is a process in which the network is first trained on a large dataset with general examples, and is subsequently re-purposed for another, more specific, classification task. In general, this means that the deeper layers are retrained for the specific domain. Consequently, fewer parameters need to be learned for the novel domain, which reduces the risk of overfitting.

2.2.2 Motion-based and stream networks

Two-stream CNNs combine regular RGB frames and optical flow images as input [229], and are an alternative approach to model temporal information. The rationale is that through images, the spatial features of an action can be encoded, while the optical flow provides information about the motion. The network consists of two streams in the network structure. The spatial-based CNN is trained on individual video frames, and the temporal stream CNN which takes stacked optical flow frames as input. The results from the two networks are concatenated with late fusion. Different information fusion methods for each stream were explored by Park *et al.* [187]. Wang *et al.* [284] used sporadically sampled fragments from the video as inputs to two-stream CNN architectures. The resulting Temporal Segment

Networks (TSN) share parameters across networks from different segments and make a prediction on each of the snippets independently. The predicted class is then the "point of agreement" between the video segments. This method capitalises on information from small temporal segments rather than using the video as a single input. Following the use of selected frames Diba *et al.* [55] also propose a representation and encoding of the sequence features in a Temporal Linear Encoding (TLE) layer, after the convolution feature extraction is performed. It is based on the aggregation of appearance features from each of the individual temporal fragments. Works have also included the use of depth data as stream inputs [73] in which features from the depth stream are distilled and simulated at test time as the test data does not include depth data.

Typically, inputs in the two-stream CNNs are processed independently and only fused as a last step. This approach prevents the exchange of information between the streams. As such, it is not possible to develop attention mechanisms that focus on specific parts of the input in either stream. One way of establishing these links is by additional shortcut connections between convolutional layers of the motion stream to the spatial stream. This provides benefits in optimising the network architecture and increasing the network depth Feichtenhofer *et al.* [65]. Residual learning [90] enables the model to avoid degradation in deep structures, which relates to the saturation of accuracy as layers of the network are not able to effectively learn the identity map and instead "threshold" to zero mappings.

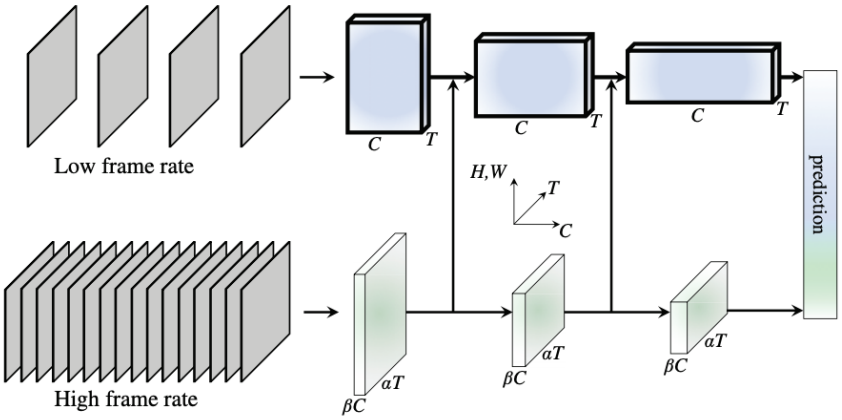


Figure 2.2. SlowFast network architecture. The slow (top) path uses lower frame rates to sample frames from the video sequence. The fast (bottom) path uses higher frame rates for sampling while include a fraction of the channels used by the slow path. Figure sourced from [64].

Works of Gkioxari *et al.* [83] and Mettes *et al.* [166] have been based on the adaptation of regional CNNs with the inclusion of multiple regions. Primary regions are considered the ones in which the main actor or actors are visible, while secondary regions contain contextual cues. The use of multiple independent or dependent regions as cues, and the separation of streams as a form to encode different features of an input, allows the focus towards discriminative regions [167, 232, 265, 295]

Instead of treating the image and motion aspects of a video in separate streams, a video sequence can be represented as a 3D volume that is composed of stacked frames. Baccouche *et al.* [5] and Ji *et al.* [108] use 3D convolutions to simultaneously encode the spatial and temporal features of such a volume. This approach is essentially an extension of the standard 2D convolutions to 3D. The resulting feature maps encode informative spatio-temporal patterns in the video volume. Tran *et al.* [258] presented the

C3D architecture and demonstrated its superiority over 2D CNNs. Improvements based on the utilisation of spatio-temporal convolutions have been further studied by Hara *et al.* [89] and Kataoka *et al.* [119]. 3D convolutions can also be used concurrently with a two-stream network. Carreira and Zisserman [31] have introduced a fusion of these two methodologies, two-stream inflated 3D-CNNs (I3D), that add a temporal dimension to the kernels of both convolutional and pooling layers. The work considers the creation of two I3D models that are applied to static image and optical flow inputs, and thus allows the 3D-CNNs to benefit from the additional information of motion patterns in optical flow streams. Spatio-temporal networks can be used as a base architecture to extend the type of information processed such as queries for people regions [79], position and motion [44] and feature neighbourhood correspondence across time [27, 285].

The larger number of parameters in 3D convolution blocks and, consequently, the demand of larger datasets for 3D-CNNs to train, have motivated the introduction of alternative convolution blocks. Notably, Qiu *et al.* [201] have proposed three supplementary blocks with different configurations of a single 2D convolutional kernel for the extraction of appearance information per frame and a temporal kernel responsible for the changes of pixel values over time loosely inspired by the separable convolutions of 2D-CNNs [43, 98]. This idea has also been used to separate spatio-temporal kernels into purely spatial and purely temporal ones by Tran *et al.* [261] with the introduction of (2+1)D convolution blocks. Others have fused both solely-spatial and spatio-temporal convolutions in an effort to emphasise the spatial signal [158, 321]. Similarly, works have also focused on the use of pairs of temporal-vertical and temporal-horizontal movements [140, 245]. 3D grouped convolutions [39, 260] have also shown benefits in performing convolutions in groups and decreasing the number of GFLOPs without a reduction in accuracy. Chen *et al.* [37] have also studied how the creation of different modalities by sub-sampling part of the activations can improve computations and increase performance.

Recent works have also aimed towards improving the efficiency of spatio-temporal convolutions architecturally. Feichtenhofer *et al.* [64], motivated by the two-stream approach of 2D [229] and 3D [28] CNNs, proposed a dual 3D-CNN architecture (pathway) with each of the sub-networks using RGB frames of different frame rates. Apart from the difference in frame rate sampling, with slow and fast rates, each pathway also used a different number of channels. Notably, the number of channels in the fast pathway is 1/8 of that of the slow pathway. This decision was made because the fast pathway is supplementary to the slow pathway as the representation spatio-temporal feature capabilities are weakened based on the number of frames that are omitted. An example of the network architecture is shown in Figure 2.2. Based on the recent success in terms of efficiency for image-based models [98, 194], works on video-based models also focused on the creation of more efficient architectures while also improving performance. Models such as X3D [63] define a set of expansion parameters for the frame rate, temporal dimension input size, spatial dimensions input sizes, number of layers per block, number of channels and in-block channel numbers. Based on a backbone architecture, different models can be produced based on these parameters in order to address different computational complexities and memory usage. Other works [116] have further expanded the use of X3D with different temporal resolutions similar to SlowFast. Others [298] have focused on the use of features across multiple feature complexity through a spatio-temporal fusion of information from different network layers. The created architecture was named Temporal Pyramid Network (TPN), as the information for the final predictions are used in a pyramid-like fashion.

Other approaches towards spatio-temporal network efficiency have been towards the creation of modules that individually process the spatial and temporal modalities. Lin *et*

al. [148] proposed a Temporal Shift Module (TSM) which uses 2D convolutions frame-wise but temporally shifts channels. This corresponds to frame-wise channel activations from preceding and succeeding frames, thus forming a temporal feature dependency across the frames. Works have further aimed at the encoding of the channel dependencies in order to improve this temporal shifting operation [151]. Similar to this, Sudhakaran *et al.* [248] used a learnable gate that splits information to be temporally processed by shifting channels as in TSM and spatially convolved activations as in a Grouped Spatio-temporal aggregation Module (GSM) [158]. Jiang *et al.* [111] proposed a Spatio Temporal Motion (STM) network composed of two modules. The first is the Channel-wise Spatiotemporal Module (CSTM) which extracts spatio-temporal features similar to (2+1)D convolutions with kernels performed in groups. The second Channel-wise Motion Module (CMM) learns feature differences across frames in a channel-wise fashion. Further works [144] have focused on the temporal excitation while spatially aggregating frame activations. AdaFused [164] has been build on top of TSM and uses a learnable policy for the frame channels to keep, reuse or skip during the adaptive temporal fusion of two neighbouring frame features. Other approaches [152] use a local feature branch and a global feature branch to address temporal cross-frame feature variations. Hussein *et al.* [101] proposed a 2D module, named TimeCeption, utilising multiple frames as inputs and modelling the temporal feature dependencies through temporal depth-wise convolutions. Based on the Inception block, different kernel sizes can be used which enables the exploration of features across multiple time scales.

Recent advances in reinforcement learning and evolutionary algorithms have contributed to a reduction in human supervision for creating robust network architectures with neural architecture search (NAS) [323]. This trend has enabled the construction of architectures for specific tasks rather than general architectures [324]. In the video domain, NAS has been originally performed with the use of an acyclic graph with TSN being the backbone architecture [192]. EvaNet [195] was one of the first architectures to employ 3D convolutions within the search space. For the optimisation task, the connectivity between operations remains fixed, while the type of operations is optimised based on a search space of available operations. This effectively reduced the computational complexity of the architectural optimisation. Later works of Ryoo *et al.* [213] have used both RGB and optical flow inputs with each network architecture and their cross-layer connections discovered as part of the optimisation process. Recent works of Kondratyuk *et al.* [125] build on image-based MobileNet [98] search spaces while also including the expansion parameters of X3D. However, as has been pointed out in the majority of the aforementioned works, the design of a 3D-CNN through a gradient-based method is especially challenging considering the extensive computational and time requirements.

2.2.3 Recurrent networks

While CNNs can recognise image components and learn to combine them to classify different classes, they lack the ability to recognise patterns across time explicitly. Stream-based networks and 3D convolutions can take into account motion, but do not explicitly deal with variations in the temporal performance of an action or interaction. An alternative approach is to use recurrent neural networks (RNNs) that model temporal patterns using a state representation. The key idea is to use some form of temporal recursion in the network that allows the persistence of information through sequences of inputs. Thus the temporal variations in videos can be efficiently modelled alongside the spatial variations.

Recurrent neural networks have been effectively used as a supplementary architecture to CNNs for extracting temporal features. In such architectures, spatial information is extracted though CNNs and is then passed to recurrent networks to learn the temporal characteristics of each interaction class [7, 52]. Zhao *et al.* [318] proposed an approach based on the normalisation of each layer of the network with batch normalisation [105]. The architecture is combined with a 3D-CNN using a two-stream fusion of the RNN and CNN. The use of multiple recurrent networks has also been extended to include tree structures (RNN-T) [143], to perform a hierarchical recognition process in which each RNN is responsible for learning an action instance based on an Action Category Hierarchy (ACH). This allows for the distinction between very dissimilar classes high in the hierarchy, while subtle differences between related classes such as a handshake and a fist bump are dealt with in the lower nodes.

Recurrent Neural Networks suffer from vanishing gradients. This issue causes the updates in the network weights of the top layers to gradually diminish as the number of data-processing iterations increases. This hinders learning the temporal parameters effectively. To overcome this issue, Long Short-Term Memory (LSTM) RNNs [96] have been introduced that include additional "memory cell" modules that decide through gating whether to keep the processed information. As such, they are capable of maintaining information over longer periods, which allows them to learn long-term dependencies [46]. This is essential for the modelling of interaction classes as the distinctive information is often present in different phases of the interaction.

Other works [57, 147, 271] have shown that the combination of convolutions and long-term recursions performs well for recognition tasks in videos. Donahue *et al.* [57] were effective in both image and video description by directly connecting powerful feature extractors such as CNNs with recurrent models. Similarly, Baccouche *et al.* [5] extracted features from the 3D-CNN architecture and extended the work to a two-step recognition process with a LSTM. The first step was the use of 3D convolutions for the extraction of spatio-temporal features. The second step is based on these learned features that are passed to the LSTM so the model can make predictions on the entire video sequence. As such, the network can benefit from both short-term and long-term temporal information.

2.3 Addressing motion permutations and class-specific features

Our work differentiates from these methods as our scope focuses on three different aspects of spatio-temporal information.

We initially study patterns over different spatio-temporal sizes (in Chapters 4 and 5). Through the extraction of features across different durations and spatial windows, a more robust and diverse representation of the video data can be created. We demonstrate that these patterns are not separate from each other. Instead, a strong dependence exists between features extracted from different spatio-temporal region sizes. Enforcing an alignment between these features to a common representation can create local features that also correspond to extended regions while also creating patterns of extended size that are respectively consistent along shorter sequences.

We secondly study the correspondence between features and classes (in Chapter 6). As certain features or combinations of features relate more to a specific class, we investigate how the regularisation of layer features based on their importance to a specific class can

improve the accuracy of classifications. The direct link between classes and layer features can provide insights in terms of both features as well as the attention regions that are associated with a specific class across multiple layers within the network.

Our final aim in this thesis is to provide visual explanations for the spatio-temporal features that are learned by 3D-CNNs (in Chapter 7). We explore feature visualisations as a visually inoperable qualitative measure for spatio-temporal models. Methods that can demonstrate the inner workings of CNNs not only improve their overall transparency but can also demonstrate future directions towards improving spatio-temporal networks.

The aforementioned aims of our work are evaluated based on commonly used action recognition datasets. We provide a detailed overview of the publicly available datasets in the following chapter alongside our chosen datasets for each task.

Chapter 3

Datasets for Video Understanding

In this chapter we overview available datasets for human action recognition. We explore the datasets both historically as well as based on their public availability. We also detail our choices for the datasets used in this thesis.

3.1 Overview of video action datasets

The past two decades have presented a large growth over both the number of action recognition datasets as well as their sizes. This relates to the inclusion of more complex spatio-temporal human actions. The introduction of CNNs to action recognition tasks has pushed the requirements for increased datasets in order to train accurate and robust models. Labelled datasets are suitable benchmarks because they allow for a direct comparison between methods. This generally leads to better understanding of the algorithmic advantages and limitations, and therefore leads to performance progression.

This chapter is structured as follows. We first list the available datasets for human action recognition. We also present each of the datasets that we believe have addressed a specific aspect of video understanding and explain their importance. The following sections Sections 3.2.1 to 3.2.5 present the datasets that we used during our experiments and we detail our choices for selecting them.

In this section we provide a catalogue of the most widely used and popular datasets for action recognition chronologically. The majority of these datasets have also been a point of study in recent literature reviews [33, 92, 102, 128, 197, 233, 242, 274].

Table 3.1 demonstrates a comprehensive overview over action recognition datasets. It demonstrates the number of action classes that have been allocated alongside the total clips/videos that each dataset includes. Column H./N.H. indicates if the video actions are performed solely by human actors or could also include other actors (e.g. animals or animations). The duration label shows the average clip duration in either seconds (s.) or minutes (m.).

The KTH [218] was one of the first datasets that captured basic spatio-temporal actions. In total 2391 action sequences were used with greyscaled frames. The labels only included six actions: *walk*, *jog*, *run*, *box*, *hand-wave* and *hand-clap*. Variations between video sequences include indoor or outdoor settings and different clothes. The dataset has been widely used in earlier methods utilising spatio-temporal features such as spatio-temporal ROI extractors [56, 181, 290], bags-of-features [165, 179, 215, 281] and (Improved) Dense Trajectories [279]. KTH has been largely surpassed by modern datasets.

The Hollywood [136] and Hollywood 2 [161] datasets were the first datasets to move from videos obtained in lab-environments to extraction of actions from media such as

Table 3.1. Video datasets for action recognition. The presented datasets are arranged chronologically. Bold text denotes datasets used in our experiments.

Dataset	Avail.	Action		Actors		Duration	Year	Purpose
		Classes	Instances	H./N.H.	# Act.			
KTH [218]	✓	6	2K	✓ / -	25	∼ 12s.	2004	greyscaled videos
CAVIAR [17]	✓	9	28	✓ / -	<30	∼ 8s.	2004	Wide-angle view of actions
Weizmann [18]	✓	10	90	✓ / -	8	∼ 12s.	2004	fixed camera axtions
ViSOR [272, 273]		N/A	N/A	✓ / -	∼ 250	∼ 12s.	2005	Videos surveillance
IXMAS [287]	✓	11	390	✓ / -	10	∼ 5s.	2006	RGB and motion caption data
Coffee & Cigarettes [137]		2	245	✓ / -	∼ 5	∼ 5s.	2007	Smoking/drinking in movies and TV
CASIA Action [1]	✓	15	1,446	✓ / -	24	N/A	2007	Actions from video cameras
UCF Sports [209]	✓	9	150	✓ / -	<100	∼ 5s.	2008	Actions from sports videos
Hollywood [136]	✓	8	475	✓ / -	<100	∼ 16s.	2008	Human actions in films
UIUC [259]	✓	14	532	✓ / -	<100	∼ 6s.	2008	Few examples dataset
UT-interaction [212]	✓	6	90	✓ / -	60	∼ 17s.	2009	Outside recordings
BEHAVE [19]	✓	10	163	✓ / -	<50	40s.	2009	Human interactions
UCF-11 [150]	✓	11	1K	✓ / -	100+	∼ 5s.	2009	YouTube videos of human actions
i3DPost MuHAVi [82]		12	>1K	✓ / -	100+	N/A	2009	Multi-view human actions
Hollywood2 [161]	✓	12	3K	✓ / -	100+	∼ 12s.	2009	Human actions in films
TV-Human Interactions [191]	✓	4	300	✓ / -	100+	∼ 3s.	2010	Sourced from TV shows
UCF-50 [206]	✓	50	5K	✓ / -	100+	∼ 15s.	2010	Web videos of human actions
Olympic Sports [178]	✓	16	800	✓ / -	100+	∼ 3s.	2010	Human actions in sports
HMDB-51 [133]	✓	51	7K	✓ / -	100+	∼ 3s.	2011	Human motions from movies
CCV [114]	✓	20	9K	✓ / -	100+	∼ 80s.	2011	Videos sourced from the web
ASLAN [124]	✓	432	4K	✓ / -	100+	∼ 5s.	2011	Videos for action similarity
UCF-101 [235]	✓	101	13K	✓ / -	100+	∼ 15s.	2012	Web videos of human actions
CAD-60 [251]	✓	12	60	✓ / -	<30	∼ 45s.	2012	Videos of human motions
THUMOS'13 [104, 112]	✓	101	13K	✓ / -	100+	∼ 15s.	2013	Web-videos extending UCF-101
CAD-120 [129]	✓	12	120	✓ / -	<60	∼ 45s.	2013	Videos of human motions
Sports-1M [118]	✓	487	1M	✓ / -	1,000+	∼ 9s	2014	Multi-labelled spirts actions
THUMOS'14 [104, 113]	✓	101	16K	✓ / -	100+	∼ 15s.	2014	extension of THUMOS'13
ActivityNet-100 [25]	✓	100	5K	✓ / -	100+	∼ 2m.	2015	Untrimmed web videos
Watch-n-Patch [293]	✓	21	2K	✓ / -	7	∼ 30s.	2015	RGB-D data for daily activities
ActivityNet-200 [25]	✓	200	15K	✓ / -	100+	∼ 2m.	2016	Untrimmed web videos
YouTube-8M [2]		N/A	N/A	✓ / ✓	N/A	N/A	2016	Multi-labelled YouTube videos
Charades [228]	✓	157	67K	✓ / -	267	∼ 30s.	2016	Daily activities videos
ShakeFive2 [74]	✓	5	153	✓ / -	33	∼ 7s.	2016	Human interactions w/ pose data
OA [142]	✓	48	480	✓ / ✓	<100	5s.	2016	Ongoing actions in web videos
CONVERSE [60]		10	N/A	✓ / -	N/A	N/A	2016	Human interactions
DALY [288]	✓	10	4K	✓ / -	100+	∼ 8s.	2016	Daily human activities
Okutama Action [8]	✓	12	4700	✓ / -	∼ 400	∼ 60s.	2017	Actions with aerial view
Kinetics-400 (K-400) [120]	✓	400	306K	✓ / -	1,000+	∼ 10s.	2017	Large-scale human actions dataset
Someting-Someting v1 [87]	✓	174	109K	✓ / -	100+	∼ 4.1s.	2017	Human actions with objects
AVA [88]	✓	80	392K	✓ / -	100+	∼ 2.7s.	2017	Atomic human-object interactions
Moments in Time (MiT) [169]	✓	339	1M	✓ / ✓	1,000+	3s.	2017	Event-based high variance data
MultiTHUMOS [302]	✓	65	16K	✓ / -	100+	∼ 4.8s.	2017	Densely-labelled action recognition
Diving-48 [145]	✓	48	18K	✓ / -	N/A	∼ 3s.	2018	Competitive diving videos dataset
EPIC-KITCHENS-55 [48]	✓	2,747	40K	✓ / -	35	3.1s.	2018	Ego-centric actions
Kinetics-600 (K-600) [29]	✓	600	495K	✓ / -	100+	∼ 10s.	2018	Large-scale human actions dataset
VLOG [70]	✓	30	122K	✓ / -	10.7K	∼ 10s.	2018	Lifestyle VLOGs dataset
Something-Something v2 [87]	✓	174	221K	✓ / -	100+	∼ 3.8s.	2018	Human actions with objects
Kinetics-700 [30]	✓	700	650K	✓ / -	1,000+	∼ 10s.	2019	Large-scale human actions dataset
Jester [162]	✓	27	148K	✓ / -	1,376	3s.	2019	Webcam hand gestures
HACS-Clips [316]	✓	200	482K	✓ / -	1,000+	2.0s.	2019	Fixed-duration clips of human actions
HACS-Segments [316]	✓	200	139K	✓ / -	1,000+	2.0s.	2019	Segments from YouTube videos
IG65M [77]		N/A	65M	N/A	N/A	N/A	2019	Actions in Instagram videos
AViD [196]	✓	887	450K	✓ / -	1,000+	∼ 9s.	2020	Diverse with blurred faces
HVU [54]	✓	3K	572K	✓ / -	1,000+	∼ 10s.	2020	Multi-label/task video understanding
HAA500 [45]	✓	500	10K	✓ / -	1,000+	∼ 2.1s.	2020	Diverse atomic actions
Kinetics-700 (2020) [234]	✓	700	647K	✓ / -	1,000+	∼ 10s.	2020	Large-scale human actions dataset
FineGym [223]	✓	530	33K	✓ / -	100+	∼ 2s.	2020	Actions from gymnastics
EPIC-KITCHENS-100 [49]	✓	4,053	90K	✓ / -	37	∼ 3.1s.	2020	Ego-centric actions

web videos, movies and TV series. Both datasets include action segments from movies with the initial version of the dataset including 475 videos and 8 action labels: *answer phone*, *get out car*, *handshake*, *hug person*, *kiss*, *sit down*, *sit up* and *stand up*. The second version increased the dataset size to 3669 videos with the addition of four additional action labels: *drive car*, *eat*, *fight person* and *run*. Although the datasets included more variations and added action label complexity, they were still with controlled video conditions. This included limited camera motion, motion blur and background clutter. Later datasets aimed towards the inclusion of such settings.

At the time that UCF-50 [206] was first introduced, it included the largest and most diverse collection of human actions in videos with 50 different action labels and over 5K videos. The videos had been sourced from YouTube and include both amateur and professional videos effectively addressing the requirements for diverse video conditions from previous datasets [18, 212, 136, 161]. The use of web-videos for the creation of datasets has been a basic property in later datasets for action recognition.

The first large-scale human actions video dataset was Sports-1M [118]. It included at total of 487 labels for sport-related actions. The dataset was made specifically for CNN models in order to utilise the 1M clips sourced from YouTube as a common benchmark. The dataset was also intended for pre-training networks. Similarly to the UCF variants [150, 206, 235] the videos were obtained from YouTube. Because of the large size of the dataset, a taxonomy similar to ImageNet [131] was used to create six internal nodes for multiple actions: *Aquatic Sports*, *Team Sports*, *Winter Sports*, *Ball Sports*, *Combat Sports* and *Sports with Animals*. Each class has on average 2000 videos however the dataset is weakly annotated with approximately 5% of the data including more than one label. However, as the labelling is noisy, motion cues are not clearly distinguishable and therefore certain actions are classified by specific objects or backgrounds that are associated with an action.

Following Sports-1M as a dataset for deep learning models, ActivityNet [25] included general daily human actions that have been manually labelled. The initial version included 100 action categories and approximately 5K videos. The second version doubled the number of classes to 200 while also increasing the number of videos to 15K which was also used for the ActivityNet 2016 challenge [23]. Although the dataset was used as a common benchmark for models, it has been largely surpassed by larger and more diverse datasets such as Kinetics [31].

Something-something [87] dataset has been introduced for human-object interactions. The dataset consists of first person videos where the actions performed are based on every-day objects. The goal of the dataset is fine-grained video understanding with the first dataset being composed of 108,499 videos while the second version including 220,847 videos. The dataset is especially useful for training and benchmarking video models as the videos are of relatively high resolution. However, the dataset only addresses human-object interactions in first person and not general human actions and interactions as is done in other datasets (e.g. Kinetics, HACS and MiT).

The Holistic Video Understanding (HVU) dataset [54] contains approximately 572K clips organised hierarchically with different levels of taxonomies. The dataset is built over the AVA [88], Kinetics-400 [120], HMDB-51 [133], UCF-101 [235] and HACS [316] datasets creating a super-set of all of the aforementioned datasets. It includes a total of over 3K labels based on the semantic aspects of video scenes, objects, actions and events. Apart from video classification, the dataset can also be used for the supplementary tasks of video captioning and video clustering. One limiting factor of the dataset as an action recognition benchmark is its recency. Although established datasets such as Kinetics and MiT still

suffice in terms of the complexity and challenges, there is not a significant number of works based on the dataset. This means it is more difficult to understand the merits of a newly proposed method because it can only be compared to a limited number of other works.

The Anonymised Videos from Diverse countries (AViD) [196] is a publicly available dataset from various countries. The aim of the dataset is to address data bias of datasets in which the majority of the videos are collected from specific countries. As culture is a factor that can effect the way an action is performed [268], the diversification of data through the inclusion of examples from different countries is important. In addition, the dataset is static instead of requiring a download script such as Kinetics, HACS or HVU and therefore the number of videos does not change based on their availability on online hosting websites (such as YouTube). Videos include blurred faces to anonymise the actor identities. The dataset includes a total of 887 classes with clips being sourced from Kinetics, Charades and Moments in Time. Actions that were dependant on facial expressions such as "smiling" or "applying eyeliner" were removed as the dataset is anonymised, resulting in 736 classes. The dataset introduces 157 additional action categories with a total number of 450K videos. Similar to HVU, due to its recency, the dataset has not yet been widely used for either benchmarking or pre-training.

Recent efforts have also been made towards the automation of the data collection process in order to enable the creation of datasets for action-related tasks with minimum human effort requirements. This includes the use of skeleton-based data to simulate the performance of an action based on frames [121] or through the use of virtual parametarised environments [103].

3.2 Datasets used in this thesis

Based on the characteristics of the overviewed datasets, we select seven datasets for our experiments. Our choice has been motivated by either the requirements of a large pre-training dataset, datasets for benchmarking or transfer-learning. We present these datasets and explain our choices in the following subsections.

3.2.1 Human Motion Database (HMDB-51)

One of the first datasets to address the limited number of action categories and classes of previous video datasets such as Hollywood [135] (with 8 action classes), UCF Sports [209] (with 9 actions), Hollywood2 [161] (with 12) and Olympic Sports [178] (with 16), was HMDB-51 [133]. Although the number of classes is close to UCF-50 [206], HMDB-51 is less associated with sports and instead includes various action classes. Some intra-class variations include visibility of body parts, camera motions and viewpoints and clip quality. The videos used have been sourced from movies segmented to smaller video segments corresponding to action categories such as *hand-waving*, *sword fighting* and *running*. In total, it includes 6,766 clips with each action class having a minimum of 101 clips. The videos in the dataset have an average duration of approximately 3 seconds with a frame rate of 30. However, there is a significant amount of variations in terms of the labelled completion times between actions. For example, the shortest example includes only 40 frames (for class *kick*), while the longest clip from class *brushing hair* consists of 649 frames. Although the dataset has been widely used for comparisons, the number of examples is now considered small and the number of classes limited. It is considered less effective for use by current models for benchmarking because of this.

Because of the small number of both classes and examples compared to current data requirements of models, we use HMDB-51 as a fine-tuning dataset to test the feature transferability of our models.

3.2.2 UCF-101

Following the introduction of HMDB-51, UCF-101 [235] is the enlargement of the prior UCF-50 dataset to 101 action classes. Most of the previous datasets have been based on the use of either actors with fixed backgrounds [18, 218, 287] or scripted actions and interactions [133, 136, 161]. In contrast, UCF-101 was based on videos sourced from YouTube, which includes actions and interactions that are not pre-planned. The total number of clips is 13,320 with a fixed frame rate of 25. The average clip duration is approximately 15s. , with the amount of variation in terms of duration between classes being higher of that of HMDB-51. The class with the lower average number of frames is *Jump Rope* with an average of 346 frames, while class *Rock Climbing Indoor* reaches a maximum number of in-clip frames of 832. A subset with 24 classes has also been used in the THUMOS'13 challenge. Additionally, there are three different splits that can be used to create the training and validation sets.

In our experiments, we use UCF-101 similarly to HMDB-51 for testing the fine-tuning capabilities of our proposed models. All of our experiments use the first split (split1) for the creation of the training and validations sets.

3.2.3 Kinetics Video Datasets

The previously described datasets only included a limited number of actions and a relatively small number of videos in total. Data in the range of tens of thousands does not suffice for accurate class predictions for human actions with large degrees of variations. The inclusion of larger and more diverse datasets not only benefits the feature variations that can be modelled, but also can effectively address data-related problems such as overfitting. The Kinetics dataset was originally introduced by Kay *et al.* [120] with the original variant (K-400) including 400 action classes and a total of approximately 306K videos. The goal of the dataset is to provide a common benchmark similar to the image-based datasets of ImageNet [51] for image recognition or COCO [149] for object segmentation. The dataset is been widely popularised and used as a common benchmark for action recognition models. Later variants (K-600) [29] increased the dataset to 600 action classes with a new total of around 495K clips. The 600 expansion shares 368 classes with the original 400 variant with the 32 remaining classes being renamed or removed altogether. The later 700 (K-700) [30] variant included 100 additional classes in order to both enlarge the dataset to a greater number of classes as well as to deal with the approximate 5% size reduction per year of the dataset based on YouTube video availability. It is also worth noting that this reduction does not account for region-restricted videos and thus reductions do vary across countries. The latest version [234] does not include additional classes from the later 700. It instead aims to maintain the number of total clips to around 650K as approximately 15K videos became unavailable. The average video duration is $\sim$ 10 seconds for all variants.

We use three of the Kinetics variants including K-400 and K-700 with both the 2019 and 2020 versions. Our main comparisons with models from the literature are done over the 400 variant as it contains the largest number of results to compare to. The additional experiments on the 700 (2019) variant in Chapter 4 and (2020) in Chapter 5 are performed in order to include experiments on datasets with increased number of classes and variations

in videos. We do not distinguish results between the 2019 and 2020 versions as the difference only amounts to approximately 15K different videos between the two datasets, which is about 2% of the dataset. In comparison to the 400 variant, K-700 is 2 times larger in terms of number of clips.

3.2.4 Moments in Time (MiT)

The Moments in Time (MiT) [169] is built on the same principles as Kinetics with the introduction of a collection of one million temporal action segments. The clips have durations of 3 seconds and can include both human and non-human actions. In comparison to Kinetics, the dataset is significantly more diverse in terms of both the 339 labels that are included, as well as the intra-class video variations that are observed. MiT also includes a significantly higher number of videos per action class in comparison to Kinetics with approximately 2.4K clips for each action class. Because of the diverse nature of MiT, in terms of both labelling and videos, class prediction is considered to be more difficult than other large-scale datasets that include more fine-grained action labels [25, 87, 118, 120, 228].

We use the MiT dataset in our experiments in order to include additional results on a large and highly variant dataset, apart from Kinetics. An example of how class labels of different complexity levels result in large intra-class video variations can be seen for class *"opening"* which may include clips of opening doors, curtains, eyes or mouths. These differences in notions for class verbs can provide challenges in the visual recognition of a class.

3.2.5 Human Action Clips and Segments (HACS)

The Human Action Clips and Segments (HACS) [316] dataset includes two variants. The first is the *Clips* variant that uses 200 action labels over 1.5M 2-second clips. These labels include clips from videos that contain both the underlying action classes (labelled as positive) and those that do not include the labelled action classes (labelled negative). All clips have been sourced from approximately 500K YouTube videos. There are about 482K positive clips while the rest are clips in which no actions are performed. Both the positive and negative segments are similar in terms of the actors, backgrounds and objects that may be used, but differ in terms of performing or not performing the corresponding action. The second annotation type, named *Segments*, consists of temporal segments used for temporal action localisation. The HACS dataset is currently one of the largest datasets for human action recognition with approximately 2.4 times the number of clips per class compared to K-700. This significantly helps accurate model benchmarking and makes HACS a suitable pre-training dataset.

In our experiments in Chapters 4 to 6, we use HACS as a pre-training dataset and as a dataset for benchmarking. Because of the large number of clips per class, the dataset is ideal for both tasks which allows us not only to pre-train the convolutional weights of our models but also produce benchmarks in comparison to models from the literature.

Chapter 4

Improving Action Recognition through Time-Consistent Features

In this chapter we explore the challenges of modelling temporal performance variations across human actions and interactions in video sequences. Our focus for addressing the motion variations of actions is aimed at the use of a novel method named *Squeeze and Recursion Temporal Gates*, that aligns the temporally short and spatially local feature patterns, and the motions that they represent within the context of the entire video sequence [1] .

4.1 Introduction

Despite the significant progress in the state-of-the-art models for human action and interaction recognition, notable challenges in capturing temporal variations are still present. Problems such as inconsistencies in temporal durations of actions, differences in the performed sets of movements, as well as changes in appearance based on viewpoint, remain either partially, or not at all addressed. These factors can significantly impact performance of models with the accuracy fluctuating from class to class given intra-class dissimilarities.

Human actions and interactions can vary based on the actors or settings that they are preformed in. Even though the underlying action class in different action instances remains the same, it present notable degrees of variation, because the action identity [269] that is associated with the main actor of each instance is different. These variations are based on the interpretation that each actor has for an action or the different skill levels of actors in different instances. The varying degree of complexity of the labels assigned to an underlined action constitutes the creation of a common set of difficult to define movements. The ambiguity of a common measure for identifying each action is reflected by the movements performed and their overall temporal length. Thus, although the action can be considered as variations of the same action class, their temporal durations that are relevant to the action are notably different. Specifically, addressing these temporal variations not only provides a broader set of features, that better correspond to a large number of intra-class instances, but also improves the overall prepotent identity [269] that is expected to be learned for that class. An approach based on which temporal information of each instance can be studied through attention. With the use of temporal attention, it is possible to study the relevance of temporal patterns over varying durations, with respect to the action classes that are performed.

[1] The code corresponding to this chapter's method is available at: https://git.io/JfuPi

Considering how differences in the temporal length and performance of an action are connected with the action's prepotent identity, we focus on creating a feature alignment between temporally short motion patterns and their extended counterparts across the entire video sequence. The proposed method named Squeeze and Recursion Temporal Gates (SRTG) uses the extracted convolutional features over constrained spatio-temporal locations and incorporates their general relevance across the entire temporal extent of the input. The created volume, that additionally encapsulates the dynamics of the local features with respect to the entire video sequence, is compared to the strictly local features. Through strengthening highly similar features across the two feature volumes, a generalised variant of the local patterns can be created. The resulting activation volume from feature fusion can increase the temporal locality and extend representations to incorporate feature changes across temporal length variations.

In Section 4.2 we review approaches that are based on the fusion of feature attention and local convolutional features. In Section 4.3 we provide a complete description of our proposed Squeeze and Recursion Temporal Gates. Our environment setting alongside our experiment results are presented in Section 4.4. Fine-tuning experiments of our methods are evaluated in Section 4.5. We discuss our results and conclude in Section 4.6.

4.2 Attention fusion for convolutional features

We identify two main groups of approaches that incorporate attention based on the extracted convolutional features. The first group of methods is based on the use of attention as a mask that can be applied over the input. The second category considers feature-based approaches that fuse attention between locally extracted patterns with globally informative features.

4.2.1 Feature mask through attention

Initial attempts for masking convolutional features, in image tasks, were based on the creation of Soft-attention [35, 106] that weighs pixels across regions. The notion of attention was then later explored by Wang *et al.* [276] where they proposed a separate *Attention Module*. The Attention Module is based on an encoder-decoder structure that creates the corresponding feature mask which can relate to either low-level or high-level features, based on the complexity of the layer that the module is applied to. Similarly, Chen *et al.* [36] used feature attention both spatially and channel-wise. This resulted in the creation of masks that not only corresponded to spatial locations of objects within the input images, but also solely addressing the features that relate to these objects. Later works have been explored that addressed both the use of recurrent connections across the encode-decoder attention structure [314], spatial and global feature attention across residual encoder-decoder connections [231] and incorporation of localisation [249].

In the video domain, works of Shikhar *et al.* [226] create a (soft) attention mechanism by extracting spatial (2D) convolution features and use each frame instance as input to a recurrent network. Girdhar and Ramanan [80] proposed a dual attention module to address class-agnostic and class-specific informative regions. Chen *et al.* [38] also use a dual attention block inside a CNN where the first attention branch selects the features that best correspond to the video objects, while the second branch highlights their spatio-temporal locations across the video sequence. Other works that consider spatio-temporal attention have been based on the feature activation in a specific spatio-temporal location, in relation

to the adjacent positions. This approach has been named *Non-local neural networks* [285] which captures dependencies across couples of adjacent space-time positions.

Despite the great promise that these methods have shown for the extraction of robust spatio-temporal features, there is still a lack of explicitly addressing the locality of the extracted convolutional spatio-temporal features within the context of the entire video. Most of the works either focus on the creation of masks as soft-spatio-temporal-attention [38, 80, 226] or the study of feature dependencies across pairs of space-time locations [285]. There is room for improvement by exploring the relevance of local features within the context of the entire video sequence.

4.2.2 Action recognition through attention-based calibration

Hu *et al.* [100] have investigated how cross-channel dependencies in convolutional layers can be used to emphasise specific image-based features. Their proposed *Squeeze and Excitation* block uses a vectorised average version of convolutional activations, created through the squeeze function, which act as a global channel descriptor. The created down-sampled version of the activation map is then used by the excite method and utilised as a non-linear gating technique to produce a weighted activation map. The per-channel weighted map is then applied to the original features to create feature-calibrated activations. Based on the use of global information, additional works include *Gather and Excite* [99], which uses a regional-based version of Squeeze and Excitation, and *Point-wise Self Attention* [317] which can connect feature map regions to create self-adaptive attention. Other works have been based on the use of residual connections with *Bottleneck Attention Modules* (BAM) [188] creating attention maps across the spatial and channel dimensions of the module inputs. BAM follows the notion of utilising global information [100, 99, 317] for discovering channel attention, while spatial attention is discovered by convolving the original input. Variations such as CBAM [291] consider the creation of channel-based and spatial-based attention individually.

In the field of video recognition, one of the first attempts for using attention as a method for calibration, was by Long *et al.* [154]. In their work, attention was expressed in the form of a per-channel condensed vector similar to that of *Gather and Excite* and was used to cluster channels within the activation map. The clustering procedure was done for RGB frame features, convolutional features from optical flow and audio inputs. Clusters were then concatenated to create general descriptors for all the mentioned combinations of input types. Other approaches have drawn inspiration from non-local mean denoising operations [22] using a non-local averaging over images and extend it to videos. Qiu *et al.* [202] have proposed the overall creation of an additional stream as a global attention path that can be updated by backpropagation.

Our proposed attention-calibration method (SRTG) differs from current video-based approaches as our aim is to directly address the divergence between short-term motions, through the exploration of coherence of the extracted convolutional patterns across the entire duration of the video segment. As CNN layer activations are constrained by the locality of their receptive fields, our method attempts to calibrate the locally-learned spatio-temporal patterns with the general motion patterns. This calibration is done by reconsidering the temporal attention of features across the entire video sequence and the correspondence to the relevance of the locally extracted patterns.

4.3 Squeeze and Recursion Temporal Gates (SRTG)

In this section, we provide a formal description of the proposed Squeeze and Recursion Temporal Gates (SRTG) blocks in Section 4.3.1. We additionally describe the criteria for calculating the feature alignment between the locally extracted convolutional features and the global information activations in Section 4.3.2. In Section 4.3.3 we overview the possible configurations within Residual blocks.

Formally, layer activations are denoted as $\mathbf{a}_{(C\times T\times H\times W)}$ with C channels, T frames, H height and W width, respectively. The backbone blocks, that SRTG is applied to, include residual connections, so we therefore formulate the final accumulated activations $a^{[l]}$ as the sum of the previous block activations $a^{[l-1]}$ and the current computed features $z^{[l]}$ ($a^{[l]} = z^{[l]} + a^{[l-1]}$). We use block indices based on l.

4.3.1 Squeeze and Recursion

Squeeze and Recursion (SR) can be used in conjunction with any produced spatio-temporal activation $\mathbf{a}^{[l]} = g(\mathbf{z}^{[l]})$ given a non-linear activation function $g()$ applied to a volume of convolutional features $\mathbf{z}^{[l]}$. The process for global information creation is similar to that in Squeeze and Excitation [100] for images. For each SR block input, activation maps are averaged at their spatial dimension to create a vectorised temporal feature descriptor of the original volume. Therefore, the produced vector encapsulates average feature activation values of each frame squeezed, by their spatial size. This is used to find an average temporal attention for each feature.

Recurrent cells. We use a recurrent sub-network in order to determine the temporal feature importance of each channel of the vectorised input activation map $pool(\mathbf{a}^{[l]})_{(t)}$. The sequential structure of recurrent cells allows for the discovery of features that are generally informative over temporal video sequences. We primarily consider LSTM cells [96] and provide a brief description of their inner workings as SR sub-network.

LSTM sub-net configuration. A visual example of the sub-network is shown in Figure 4.1. The effects of salient features are emphasised within the first operation of the recurrent cell with the *forget gate layer* ($\mathbf{f}_{(t)}$), where low intensity activations are discarded. Given the vectorised input $pool(\mathbf{a}^{[l]})_{(t)}$, a decision is made with the inclusion of informative features from the previous frame $\mathbf{h}_{(t-1)}$ with the use of channel weight $\mathbf{w}_f$ and bias $\mathbf{b}_f$. Features to be stored are discovered in two parts. First, the product of the sigmodial (σ) *input gate layer* $\mathbf{i}_{(t)}$, which determines the values that are to be updated. At the same time, the vector of candidate values $\widetilde{\mathbf{C}}_{(t)}$ is created as:

$$\mathbf{f}_{(t)} = \{\sigma(\mathbf{w}_f * [\mathbf{h}_{(t-1)}, pool(\mathbf{a}^{[l]})_{(t)}] + \mathbf{b}_f)\} \tag{4.1}$$

$$\mathbf{i}_{(t)} = \{\sigma(\mathbf{w}_i * [\mathbf{h}_{(t-1)}, pool(\mathbf{a}^{[l]})_{(t)}] + \mathbf{b}_i)\} \tag{4.2}$$

$$\widetilde{\mathbf{C}}_{(t)} = \{tanh(\mathbf{w}_C * [\mathbf{h}_{(t-1)}, pool(\mathbf{a}^{[l]})_{(t)}] + \mathbf{b}_C)\} \tag{4.3}$$

Updates to the previous cell state $\mathbf{C}_{(t-1)}$ are done by initially deciding the features that are found inconsistent across time and that will be omitted ($\mathbf{f}_{(t)} * \mathbf{C}_{(t-1)}$). The second part of the update procedure consists of the forget and input gates which determine the new candidate values ($\mathbf{i}_{(t)} * \widetilde{\mathbf{C}}_{(t)}$), with the current cell state $\mathbf{C}_{(t)}$ computed as:

$$\mathbf{C}_{(t)} = \mathbf{f}_{(t)} * \mathbf{C}_{(t-1)} + \mathbf{i}_{(t)} * \widetilde{\mathbf{C}}_{(t)} \tag{4.4}$$

The final output of the recurrent cell $\mathbf{h}_{(t)}$, for temporal state (t), is the combination of current cell state $\mathbf{C}_{(t)}$, the previous hidden state $\mathbf{h}_{(t-1)}$, and current input $pool(\mathbf{a}^{[l]})_{(t)}$. The output is filtered by a sigmoid layer $(\mathbf{o}_{(t)})$ which determines the part of the input to be updated:

$$\begin{aligned} \mathbf{h}_{(t)} &= \mathbf{o}_{(t)} * tanh(\mathbf{C}_{(t)}), where \\ \mathbf{o}_{(t)} &= \{\sigma(\mathbf{w}_o * [\mathbf{h}_{(t-1)}, pool(\mathbf{a}^{[l-1]})_{(t)}] + \mathbf{b}_o)\} \end{aligned} \tag{4.5}$$

The produced hidden states $(\mathbf{h}_{(t,\forall t \in T)})$ are concatenated in a coherent sequence of filtered spatio-temporal feature activations and used conjointly with the original input $(\mathbf{a}^{[l]})$ through an element-wise multiplication operation. The produced activation map $(\mathbf{a}^{\star[l]})$ effectively incorporates the global feature dynamics for the discovered features of different spatio-temporal region sizes.

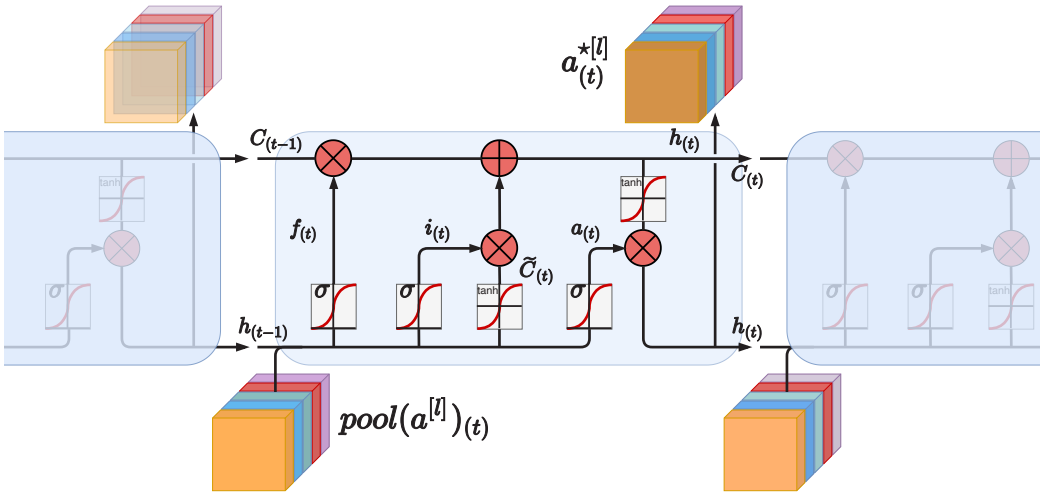


Figure 4.1. LSTMs cells. Overview of a sequential chain of recurrent cells for the discovery of globally informative local features. Each input corresponds to a temporal activation map and produces a feature vector of the same size as the input.

4.3.2 Cyclic Consistency

Supplementary to SR, cyclic mapping of each temporal instance is a widely used method [58, 286] to evaluate the similarity between pairs of temporal sequences. The basic premise considers the per-temporal location one-to-one mapping within two time sequences. We present a symbolic representation of the main idea in Figure 4.2. We define each feature space of the two temporal sequences that are considered as an *embedding space*. The defined embedding spaces are cycle-consistent if and only if each point at temporal instance t in the embedding space $\boldsymbol{A}$, has a minimum distance point in embedding space $\boldsymbol{B}$ at time t. Respectively, point t in embedding space $\boldsymbol{B}$ is required to also have a minimum distance point in embedding space $\boldsymbol{A}$ at time t. We demonstrate how two points do not exhibit cyclic consistency in Figure 4.2. In this case a temporal cyclic error occurs.

Through the use of cyclic-back consistency between points of two volumes, we can create a baseline in terms of their overall similarly. Considering the expected differences between individual features within the two feature spaces, they incorporate an overall

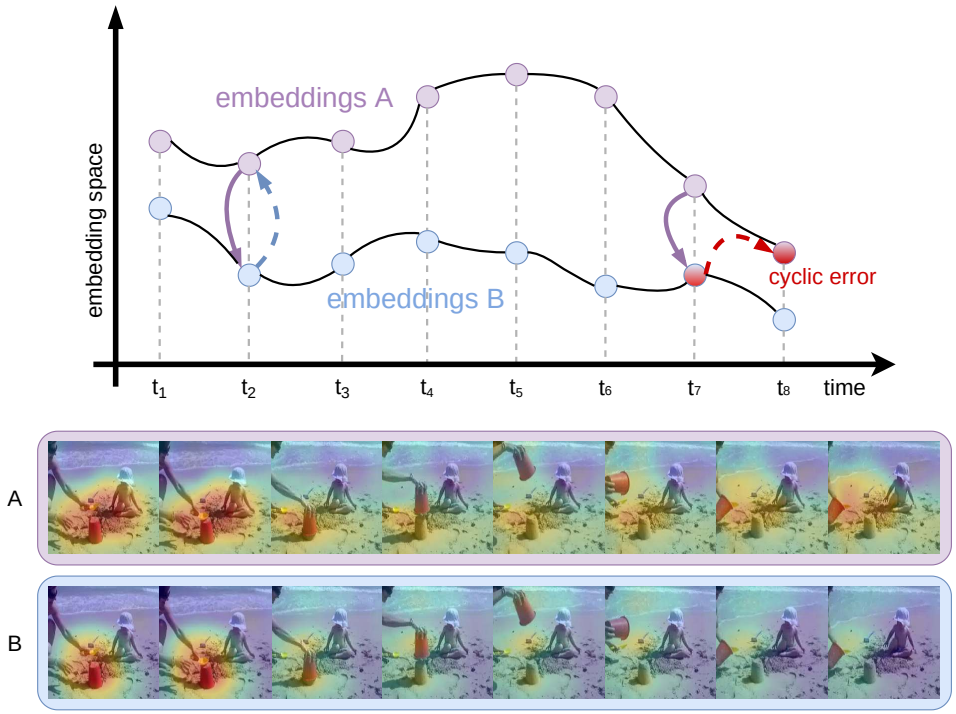


Figure 4.2. Temporal cyclic error. Cycle-consistent points should cycle back to themselves (as with points at t_2). Points that do not present this trait, exhibit a temporal cyclic error (e.g. at t_7). Salient areas for each embedding are visualised with *CFP* [240].

similarity given a cyclic consistency alignment. Therefore, cyclic consistency is a suitable method to measure the (temporal) variations between embeddings.

Soft nearest neighbour distance. Considering the vastness of embedding spaces in convolutional activations, the creation of a meaningful similarity measure is challenging. This challenge includes the selection of the nearest points in adjacent embedding spaces from points of a different space. The discovery of minimum distance spaces across the two representations is informative in terms of their feature correspondence. For this reason, *soft matches* of points in projected embeddings [84] are preferred. The mindset behind soft matching is the selection of a point in an embedding space through the weighted sum of all possible matches with higher weights for points closer by. The closest actual observation is then selected based on its distance to the soft match point.

The soft nearest neighbour of an activation $\mathbf{a}^{\boldsymbol{A}}_{(t)}$ in embedding space $\boldsymbol{B}$ is discovered through the euclidean (L_2) distances between observation $\mathbf{a}^{\boldsymbol{B}}_{(t)}$ and all points in $\boldsymbol{B}$. This process considers each frame as a separate instance for which we want to find the minimum distance point in the adjacent embedding space. We weigh the similarity to each point in adjacent embedding space $\boldsymbol{B}$ to activation $\mathbf{a}^{\boldsymbol{A}}_{(t)}$ using the softmax exponential difference between all activation pairs:

$$\tilde{\mathbf{a}}^{(\boldsymbol{B}\rightarrow\boldsymbol{A})}_{(t)} = \sum_{i}^{T} \mathbf{z}_{(i)} * \mathbf{a}^{\boldsymbol{B}}_{(i)}, \; where: \quad \mathbf{z}_{(i)} = \frac{e^{-||\mathbf{a}^{\boldsymbol{A}}_{(t)} - \mathbf{a}^{\boldsymbol{B}}_{(i)}||^2}}{\sum_{i}^{T} e^{-||\mathbf{a}^{\boldsymbol{A}}_{(t)} - \mathbf{a}^{\boldsymbol{B}}_{(i)}||^2}} \tag{4.6}$$

The softmax function creates a normal distribution of similarity weights $\mathcal{N}(\mu, \sigma^2)$, centred on the adjacent space activation at the time instance with the minimum distance from activation $\mathbf{a}^{\boldsymbol{A}}_{(t)}$. With the calculation of the nearest neighbour $\tilde{\mathbf{a}}^{(\boldsymbol{B}\rightarrow\boldsymbol{A})}_{(t)}$, the distance to

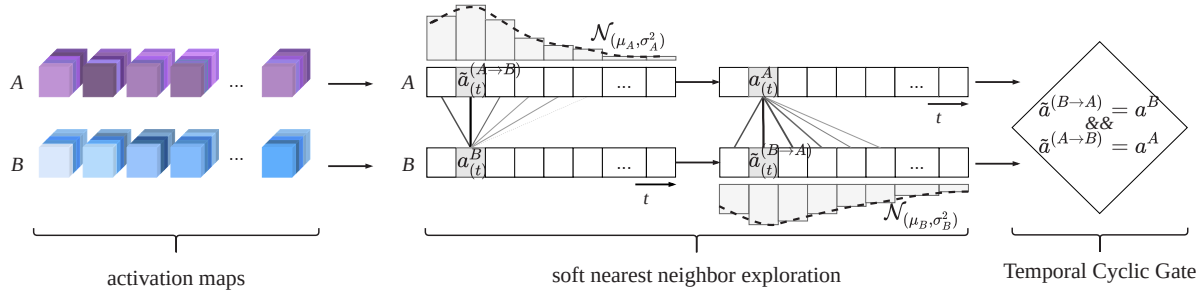


Figure 4.3. Fusion with Temporal Gating. Activations ($\mathbf{a}^{\boldsymbol{B}}_{(t_i)}$) from embedding space $\boldsymbol{B}$ are compared to the activations ($\mathbf{a}^{\boldsymbol{A}}_{(t_j)}$) in embedding space $\boldsymbol{A}$. Each frame-wise activation map ($\mathbf{a}^{\boldsymbol{B}}_{(t)}$) is calculated based on its soft nearest neighbour ($\tilde{\mathbf{a}}^{\boldsymbol{A}\rightarrow\boldsymbol{B}}_{(t)}$) in space $\boldsymbol{A}$. Afterwards, $\tilde{\mathbf{a}}^{\boldsymbol{B}\rightarrow\boldsymbol{A}}_{(t)}$ in embedding space $\boldsymbol{B}$ is also calculated. The gate is open when $\tilde{\mathbf{a}}^{\boldsymbol{A}\rightarrow\boldsymbol{B}}$ and $\tilde{\mathbf{a}}^{\boldsymbol{B}\rightarrow\boldsymbol{A}}$ are exactly and sequentially equal to $\mathbf{a}^{\boldsymbol{A}}$ and $\mathbf{a}^{\boldsymbol{B}}$.

the nearest frames in $\boldsymbol{B}$ can be computed. Based on the initially considered frame $\mathbf{a}^{\boldsymbol{A}}_{(t)}$, we obtain the closest time instance activations, through the selection of the minimum L2 distance from the found soft match:

$$\mathbf{a}^{(\boldsymbol{B}\rightarrow\boldsymbol{A})}_{(t)} = \underset{i}{\operatorname{argmin}}(||\tilde{\mathbf{a}}^{(\boldsymbol{B}\rightarrow\boldsymbol{A})}_{(t)} - \mathbf{a}^{\boldsymbol{B}}_{(i)}||^2) \tag{4.7}$$

Temporal embedding points are *consistent* if and only if the initial temporal location t matches precisely the temporal location of the discovered point in adjacent embedding space $\boldsymbol{B}$, $\mathbf{a}^{(\boldsymbol{B}\rightarrow\boldsymbol{A})}_{(t)} = \mathbf{a}^{\boldsymbol{B}}_{(t)}\ \forall t \in \{1, ..., T\}$, as visualised in Figure 4.3. To establish a consistency check for temporal points of space $\boldsymbol{A}$, the same procedure is repeated with the consideration of every frame in embedding space $\boldsymbol{B}$, by calculating the soft nearest neighbour in $\boldsymbol{A}$. The embeddings are considered *cycle consistent* if and only if all points on both embedding spaces map back to themselves: $\mathbf{a}^{(\boldsymbol{B}\rightarrow\boldsymbol{A})}_{(t)} = \mathbf{a}^{\boldsymbol{B}}_{(t)}$ *and* $\mathbf{a}^{(\boldsymbol{A}\rightarrow\boldsymbol{B})}_{(t)} = \mathbf{a}^{\boldsymbol{A}}_{(t)}\ \forall t \in \{1, ..., T\}$.

Temporal gates. The averaged temporal feature attention is part of the produced activation vector. However, it does not necessarily correspond to a one-to-one similarity with the local spatio-temporal activations. The direct fusion of two activations without considering dissimilarities in terms of their representations can lead to unrepresentative volumes. We compute cyclic consistency between the pooled activations $pool(\mathbf{a}^{[l]})$ and the outputted recurrent cells $\mathbf{a}^{\star[l]}$ to evaluate the feature similarities between the two volumes. We utilise cyclic consistency as a gating mechanism to fuse the recurrent cell hidden states with unpooled versions of the activations when they are temporally cycle consistent. This ensures that only time-consistent information of the local patterns is added back to the network. An overview of the gate states is shown in Figure 4.4.

4.3.3 SRTG configurations

Cycle-consistency calculations can be performed across multiple parts of a convolution block. Shown in Figure 4.5, we investigate six different approaches when testing SRTG as part of a convolution block. Each variant fuses global and local information in sections of the blocks. Configurations only differ in the relative locations of the SRTG and the LSTM input. Based on the residual version chosen, we consider Simple blocks with two conv operations and Bottleneck blocks with three conv operations. Not all SRTG configurations apply to the Simple blocks.

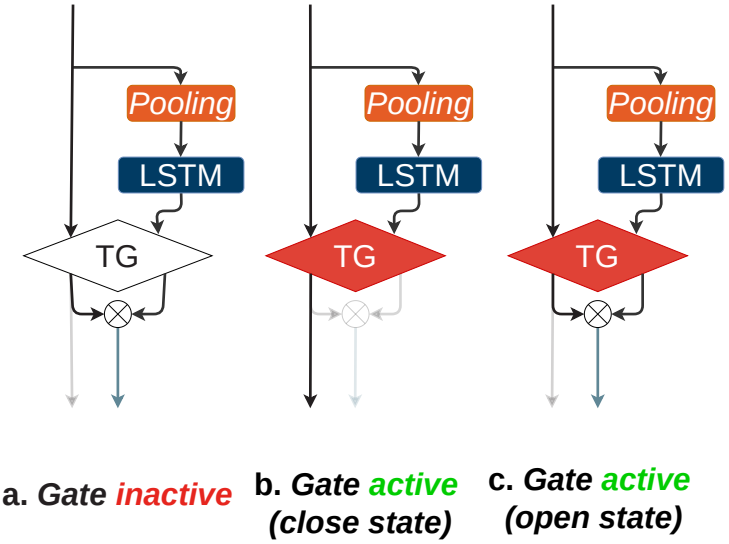


Figure 4.4. Temporal gate states. (a) Inactive state of gate, where no cyclic consistency is calculated. (b) Gate in an active and closed state in which cyclic consistency is not established. (c) Active gate where the original activations and the calibrated ones are cyclic consistent and thus are fused together by element-wise multiplication (⊗)

Start. SRTG is performed on the block's input thus input information is aligned on both global and local information. This is used in both Simple and Bottleneck residual blocks.

Top. Activations of the first convolution are used by the LSTM, with fused features being used by the final convolution. This is specific to Bottleneck blocks.

Mid. SRTG is added at the middle of Simple blocks and after the second convolution at Bottleneck blocks.

End. Local and global features are fused at the end of the final convolution, before the concatenation of the residual connection. This is only used in Bottleneck blocks.

Res. The SRTG module is applied to the residual connection. This transforms the residual connection to further include global spatio-temporal features combining them with convolutional activations from either Simple or Bottleneck blocks.

Final. SRTG is added at the end of the residual block, which allows for the activations to be calculated jointly with their representations across time on the entire video. This can be used in both Simple and Bottleneck blocks.

4.4 Main results

For this section, we overview our experiment setting for training in Section 4.4.1. The main results on the HACS dataset are presented in Section 4.4.2. Model comparisons on Kinetics-700 are shown in Section 4.4.3. The final large-scale dataset on which we test the proposed method is Moments in Time and the results can be found in Section 4.4.4. Pairwise comparisons are summarised in Section 4.4.5.

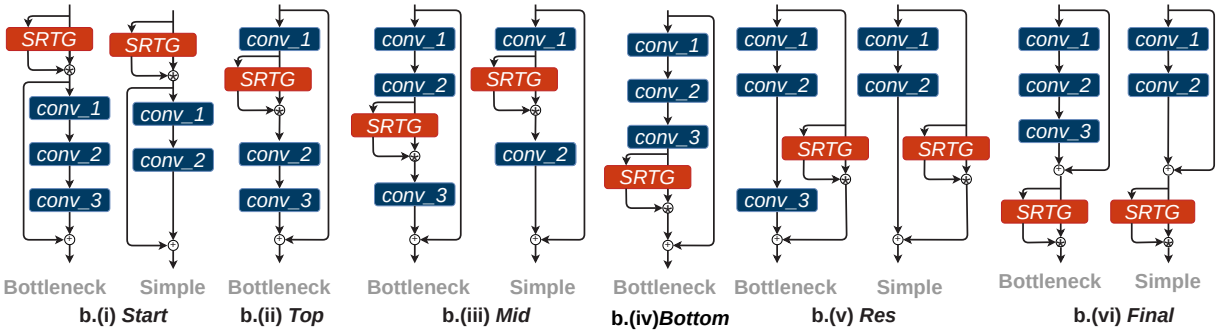


Figure 4.5. SRTG configurations based on residual blocks. Residual Networks [90] can include multiple SRTG configurations based on the use of either Simple blocks with two convolutional operations or Bottleneck blocks with three convolutional operations.

4.4.1 Experiment environment settings

The proposed SRTG modules are evaluated on ResNet backbone architectures. The selection of the ResNet architecture was based on its wide application over multiple works in action recognition [37, 39, 63, 64, 260, 261]. To extend our comparisons, we consider both 3D and (2+1)D [261], spatio-temporal convolutions over 3 architectures of different depths. For simplicity, we denominate ResNet architectures with 3D convolutions as r3d and with (2+1)D as r(2+1)d.

Interval selection. The frame selection is performed by a uniform random sampling. Based on the average clip length, for each dataset, we use equivalently sized temporal strides. This is done in order to ensure that the frames across the majority of the video sequence will be used to create the input volume. The average clip length in HACS is 60 frames, and we therefore use a stride of 2. For Kinetics, the average clip length is 250 frames so we set the temporal stride to 5. Similarly, for MiT we use a temporal stride of 3 with average duration of 90 frames.

Computational Inference. We employ two different measures to report inference costs. We report computational costs (FLOPs) similar to works of [62, 63, 64, 260] through sampling 10 clips from a single video and perform 3 crops along the spatial dimensions of size 256×256. The inference time is reported as the number of FLOPs used per spatio-temporal view (clips times crops). This provides a standardised measure of computing inference when comparing across models as shown for Tables 4.2 to 4.4.

Multigrid batch schedule. All of our experiments utilise a multigrid scheme [292] for improvements in the training speeds. The method is based on variable-sized mini-batches created from a sampling grid of possible sizes. Scaling the mini-batch size is done with respect to the original batch and spatio-temporal dimensionality by satisfying $b \times t \times h \times w = B \times T \times H \times W$, in which (b, t, h, w) represent the scaled batch, time, height and width dimensions of the input data, while (B, T, H, W) are the original dimensions. This further ensures that computational costs (GFLOPs) remain similar between the scaled and the original batches. It is worth noting that multigrid considers a *proportional* spatial scaling, thus increases and decreases in the height or width values are done in the same manner. We demonstrate the hierarchical schedule of multigrid in Figure 4.6.

Overall, the schedule alternates between two frequencies with *long cycles* in which the batch size changes after a specified number of iterations and *short cycles* that move to different mini-batch sizes at each iteration. In our experiments, long cycles iterate over

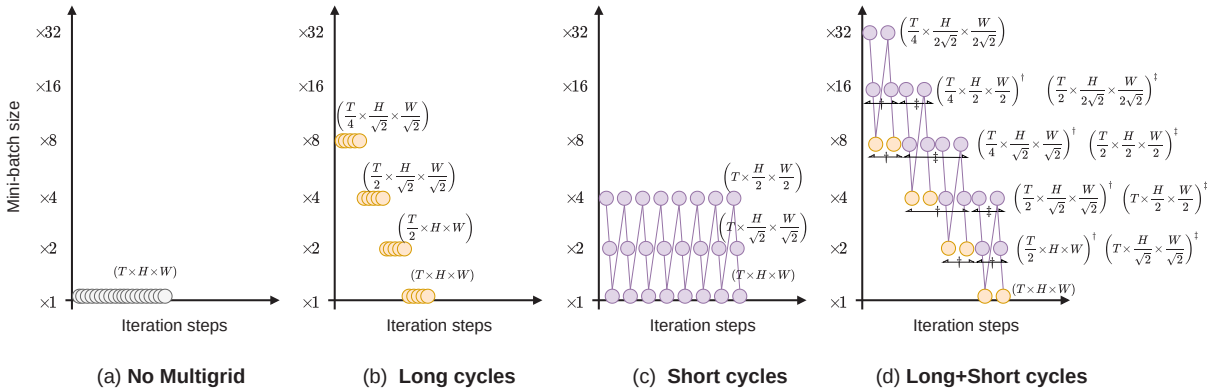


Figure 4.6. Multigrid training schedule. (a) **Baseline** uses a fixed batch size. (b) **Multigrid long cycles** loop over 4 different batch sizes within an epoch. These changes are done in a sequential order. (c) **Multigrid short cycles** iterate over adjacent sizes with each iteration step re-adjusting the batch size. (d) **Multigrid long + short cycles** fuse both (b) and (c) together to a single schedule.

4 different batch sizes for a single epoch, with total input sizes of $\left(8B\times\frac{T}{4}\times\frac{H}{\sqrt{2}}\times\frac{W}{\sqrt{2}}\right)$, $\left(4B\times\frac{T}{2}\times\frac{H}{\sqrt{2}}\times\frac{W}{\sqrt{2}}\right)$, $\left(2B\times\frac{T}{2}\times H\times W\right)$ and $(B\times T\times H\times W)$. Temporal reductions in frames are handled by uniform frame selection from the default sampled frames, while spatial reductions are done through bilinear interpolation. In contrast to long cycles, iterations in the short cycles take volume sizes of $\left(4B\times T\times\frac{H}{2}\times\frac{W}{2}\right)$, $\left(2B\times T\times\frac{H}{\sqrt{2}}\times\frac{W}{\sqrt{2}}\right)$ and $(B\times T\times H\times W)$ which effectively only progressively reduce the spatial dimensions (H, W). Short cycles are combined with long cycles to form the batch size schedule that we used. This is based on a combination of sampling strategies as shown in Figure 4.6. The learning rate follows the linear scaling rule [86] given the changes that are based on the long cycle. Reductions to batch sizes to half the size of the previous iteration will also result in a reduction by half for the learning rate. This is also done for increases in the batch size for which the learning rate is also increased by an equal amount.

Training. Results on HACS are obtained with weights for all of our networks being randomly initialised based on the recipe described by He *et al.* [91]. Frames are resized to size $320\times X$, where X is the long side in the original frame, in order to maintain the aspect ratio. The frames are then cropped to 224×224 sizes. The base number of frames that we use is 16. Both temporal and spatial dimensions are reduced/adjusted based on the defined multigrid mini-batch size schedule. The initial learning rate is set to $lr_0 = 0.1$ and reduced by 10 for steps [40, 70, 120] for a total of 150 epochs. We choose 150 total epochs as no further improvements were observed on higher number of iterations. We also use weight-decay which is set to 10^{-5}. We use a standard SGD optimiser with momentum [200] which we set to 0.9. The base mini-batch size is set to 32 and increased according to each cycle schedule. Structurally, we use a global spatio-temporal average pooling layer for feature vectorisation and a fully-connected layer for the class probabilities.

Spatial data augmentation. To improve the generalisation capabilities of our models, we further include spatial data augmentations, that are performed sequentially over frames in our data augmentation pipeline. We set a sequential probability of 80% which translates to roughly 80% of the data being further augmented after their spatial crops. A per-augmentation method probability of 40%, which corresponds to the probability that each of the augmentation method has in order to be applied. Our spatial augmentations

include Gaussian and mean blurring, RGB value changes ($\pm\{1, ..., 15\}$) and geometrical augmentations. Choices for augmentation methods was made based on [282].

Table 4.1. Comparison of r3d-34 with SRTG configurations on HACS.

Config	Gates	top-1 (%)		top-5 (%)	
		3D	(2+1)D	3D	(2+1)D
No SRTG	✗	74.82	75.70	92.84	93.57
Start	✓	75.70 (+0.88)	76.44 (+0.74)	93.23 (+0.39)	93.78 (+0.21)
Mid	✓	75.49 (+0.67)	76.68 (+0.98)	93.22 (+0.38)	93.75 (+0.18)
Res	✓	76.70 (+1.88)	77.09 (+1.39)	93.31 (+0.47)	93.86 (+0.29)
Final	✓	**78.60** (+3.78)	**80.39** (+4.49)	**93.57** (+0.73)	**94.27** (+0.70)

4.4.2 Results on HACS

SRTG module configurations. Our initial comparisons are done over different SRTG module configurations with a 34-layer ResNet backbone. We use two networks with 3D convolutions (r3d-34) and (2+1)D convolutions (r(2+1)d-34). Architecturally, ResNet-34 contains Simple blocks with two conv layers instead of the Bottleneck blocks with three conv layers. We therefore only evaluate the Start, Mid, Res and Final configurations. Results for HACS are summarised in Table 4.1 and are obtained by training from scratch. We show that all of the tested SRTG module configurations perform better than the original network without SRTG. This demonstrates the merits of our more flexible treatment of the temporal dimension. This effect appears to be stronger when the filtering is applied later. Experiments with 3D convolutions show improvements in the range of 0.88–3.78% for top-1 accuracy and 0.39–0.73% for top-5 accuracy. The top performing configurations are the Final with +3.78% top-1 and +0.73% top-5 accuracies and Res with +1.88% and 0.47% top-1 and top-5 accuracies. Similar results are also obtained with (2+1)D convolutions, with accuracy increases in relation to the original network used as baseline, and range between 0.74–4.69% for top-1 and 0.18–0.70% for top-5 accuracies. Based on these results, following experiments use the final block configuration.

Main results. In Table 4.2 we present our results in comparison to both state-of-the-art models as well as against baseline ResNet models without SRTG modules. We also overview the number of parameters and the computational costs (GFLOPs) for each tested architecture.

3D convolutions. For results obtained with 3D convolutions, we notice an average improvement of 2.3% on the top-1 accuracy across all three configurations. For **SRTG r3d-34** this increase in accuracy is of 3.8% for the top-1 and 0.8% for the top-5. The achieved accuracies with SRTG are shown to be closer to larger corresponding networks without SRTG with **SRTG r3d-34** performing similar to r3d-50 and **SRTG r3d-50** accuracies being comparable to those of r3d-101. When considering state-of-the-art architectures, r3d-50 is close in performance to I3D [31] while requiring less than half the number of floating-point operations (GFLOPs). **SRTG r3d-101** achieves comparable accuracies to TSM [148] with margins of +0.2% top-1 and 0.8% for the top-1 and top-5.

(2+1)D convolutions. For SRTG networks that instead use (2+1)D convolutions, we notice a reasonable increase in accuracy within the ranges of 1.8–3.5% for the top-1 and 0.5–1.1% for the top-5 accuracies. However, this also translates to an increase in the number of GFLOPs, as (2+1)D convolutions use two operations with spatial-only kernels followed

Table 4.2. Action recognition model comparisons on HACS. Weight initialisation sources are denoted by their respective indicators.

Model	Pre	top-1	top-5	GFLOPs × views	Params
MF-Net [39]†	K-400	78.3	94.6	11.1×50	8.0M
I3D [31]†		79.9	94.5	108.0×50	12.0M
TSM [148]†		81.4	95.5	65.0×10	24.3M
TAM [62]†		82.2	95.2	86×12	25.6M
SF-101 [64]†		83.7	**96.8**	213.0×30	53.7M
r3d-34 [119]*	K-700	74.8	92.8	26.6×30	63.7M
r3d-50 [119]*		78.4	93.8	52.6×30	36.7M
r3d-101 [119]*		80.5	95.2	78.0×30	69.1M
r(2+1)d-34 [119]*		75.7	93.8	37.8×30	61.8M
r(2+1)d-50 [119]*		81.3	94.5	83.3×30	34.8M
r(2+1)d-101 [119]*		82.9	95.7	163.0×30	67.2M
ir-CSN-101 [260]†	IG65	83.8	93.8	63.6×10	22.1M
ip-CSN-101 [260]†		84.1	93.9	63.6×10	24.5M
SRTG r3d-34 [243] **(ours)**	-	78.6	93.6	26.6×30	83.8M
SRTG r3d-50 [243] **(ours)**	-	80.3	95.5	52.7×30	56.9M
SRTG r3d-101 [243] **(ours)**	-	81.6	96.3	78.1×30	107.1M
SRTG r(2+1)d-34 [243] **(ours)**	-	80.4	94.3	37.8×30	82.1M
SRTG r(2+1)d-50 [243] **(ours)**	-	83.8	96.6	83.4×30	55.0M
SRTG r(2+1)d-101 [243] **(ours)**	-	**84.3**	**96.8**	163.1×30	105.3M

† models and weights from official repositories.
* re-implemented models and weights.

by temporal-only kernels. In terms of their accuracies compared to the baseline r(2+1)d networks without SRTG, increases are similar to those obtained with 3D convolutions. Improvement margins, across models, are between 1.3–4.7% for top-1 and 0.5–1.1% for top-5. The largest increase in performance is observed for **SRTG r(2+1)d-34** with +4.7% top-1. The best performing model is **SRTG r(2+1)d-101** which however has significant computational and memory requirements. **SRTG r(2+1)d-50** shows similar performance to SlowFast-101 [64] with only minimal increases in the number of GFLOPs and parameters. The smaller **SRTG r(2+1)d-34** can achieve accuracies comparable to the significantly larger r3d-101 and I3D while only using less than half of the number of GFLOPs of the other two models. This is also without being pre-trained to a different dataset.

Computational complexity. In Table 4.2 the accuracies are presented with respect to the computational complexity of each model (GFLOPs) and the number of parameters. Compared to the original baseline models, SRTG does not significantly effect the number of floating-point operations of the entire network. The added number of GFLOPs amounts to $\ll 1\%$ of that of the original model making the our proposed block very lightweight to compute. The importance of this can be understood by the large number of GFLOPs that is already required for performing spatio-temporal 3D convolutions. We note that there is a significantly larger difference in the number of GFLOPs based on the space-time convolution used, with either 3D or (2+1)D than the inclusion of SRTG in the network.

Table 4.3. Comparison with K-700 state-of-the-art. Flop calculation is similar to that in Table 4.2. * denotes our reproduced models.

Model	Pre-train	GFLOPs × views	top-1	top-5
I3D [31] *	K-600	108×50	53.0	69.2
TSM [148] *	HACS	65.0×10	54.0	72.2
MF-Net [39] *		11.1×50	54.3	73.4
ir-CSN-101 [260] *		63.6×10	54.7	73.8
SF-50 [64] *		65.7×30	56.2	75.7
SF-101 [64] *		213.0×30	**57.3**	77.2
SRTG r3d-34 [243] **(ours)**		26.6×30	49.1	72.7
SRTG r3d-50 [243] **(ours)**		52.7×30	53.5	74.2
SRTG r3d-101 [243] **(ours)**		78.1×30	56.5	76.8
SRTG r(2+1)d-34 [243] **(ours)**		37.8×30	49.4	73.2
SRTG r(2+1)d-50 [243] **(ours)**		83.4×30	54.2	74.6
SRTG r(2+1)d-101 [243] **(ours)**		163.1×30	56.8	**77.4**

4.4.3 Results on Kinetics-700

The selection of the Kinetics-700 dataset was done with the criteria of testing our proposed approach over an extended number of action classes to additionally demonstrate the generalisation capabilities of our approach on a larger scale. We present our findings in Table 4.3. Due to the recency of the dataset, and its significant size increase over the previous 400/600 variants, there is sparsity in the reported results of Table 4.2.

ResNet 34. The two tested 34-layer **SRTG r3d-34** and **SRTG r(2+1)d-34** networks achieve similar results with their top-1 and top-5 accuracy margins being $\leq 0.5\%$. Due to their limited computational complexity, their results are lower in comparison to models of higher complexity such as TSM [148] and ir-CSN-101 [260].

ResNet 50. Our **SRTG r3d-50** performs similar to I3D on the top-1 accuracy while only requiring half the number of GFLOPs. The top-5 accuracy is higher than that of the I3D and more competitive to the top-5 accuracy of MF-Net [39]. **SRTG r(2+1)d-50** with (2+1)D convolutions shows a small performance increase in comparison to the 3D variant. Accuracy rates achieved are similar to those of MF-Net and ir-CSN-101.

ResNet 101. The two largest tested architectures that include SRTG are comparable to results obtained from state-of-the-art methods coming second to only SlowFast-101 [64] in top-1 accuracy. They are also the top-performing models in terms of the top-5 accuracy. **SRTG r3d-101** performance can be compared to that of SlowFast-50 without however addressing temporal variations with two networks. This demonstrates that SRTG can be a viable approach for bridging the gap between image-based adapted models used for video data, and architectures created explicitly for spatio-temporal information. The (2+1)D **SRTG r(2+1)d-101** variant shows accuracy rates close to SlowFast-101 in both the top-1 and top-5 accuracies.

Table 4.4. Comparison with MiT state-of-the-art. Models denoted with (†) include additional information input sources. * denotes our own implementation.

Model	Arch. size	top-1	top-5
EvaNet [195]	NAS[323]	31.8	N/A
AssembleNet [213]		**34.3**	**62.7**
MF-Net [39] *	Fixed	27.3	48.2
I3D [31]		29.5	56.1
CoST [140]		32.4	60.0
SoundNet [169] †		7.6	18.0
TSN+Flow [169] †		15.7	34.7
SRTG r3d-34 [243] **(ours)**		28.5	52.3
SRTG r3d-50 [243] **(ours)**		30.7	55.6
SRTG r3d-101 [243] **(ours)**		33.6	58.5
SRTG r(2+1)d-34 [243] **(ours)**		29.0	54.2
SRTG r(2+1)d-50 [243] **(ours)**		31.6	56.8
SRTG r(2+1)d-101 [243] **(ours)**		33.7	59.1

4.4.4 Results on Moments in Time

Table 4.4 shows performance of the the top-1 and top-5 accuracies (%) of current state-of-the-art models in Moments in Time (MiT). Comparisons made are based on both models on pre-defined architectures (denoted with "Fixed") as well as models that employ different types of Neural Architecture Search (NAS) [323] that effectively optimise the model structure to better address the data. Our SRTG models show promising results with **SRTG r3d-101** and **SRTG r(2+1)d-101** having similar performance to AssembleNet and CoST, while outperforming the EvaNet [195] learned architecture by +1.9% in top-1 accuracy. The accuracies are also close to the top-performing AssembleNet with only a -0.6% margin for the top-1 accuracy. In comparison to pre-defined architectures such as those of MF-Net, I3D and CoST, **SRTG r(2+1)d-101** outperforms them with a margin of 1.3–6.4%. Additional results on 50-layered ResNets also show the merits of the proposed SRTG module with accuracies comparable to EvaNet. **SRTG r3d-50** largely outperforms similarly complex fixed architecture models such as I3D and MF-Net by margins in top-1 accuracy of +1.2% and +3.4% respectively. The **SRTG r(2+1)d-50** model shows almost identical performance to EvaNet without Neural Architecture Search. A marginal deficit in top-1 performance is observed in comparison to CoST (-0.8%). **SRTG r3d-34** and **SRTG r(2+1)d-34** also show promising results while having accuracy rates higher than models that use multiple data type inputs. In comparison to learned models [195, 213], SRTG models come with added computational reductions, as there is no additional objective towards permuting the base model.

Table 4.5. Pairwise comparisons for r3d networks with and without SRTG on HACS, K-700 and MiT.

(a) Comparisons for r3d original architectures with/out SRTG enabled

Dataset	accuracy (%)	r3d-34		r3d-50		r3d-101	
		None	**SRTG**	None	**SRTG**	None	**SRTG**
HACS	top1	74.8	**78.6** (+3.8)	78.4	**80.4** (+2.0)	80.5	**81.7** (+1.2)
	top5	92.8	**93.6** (+0.8)	93.8	**95.4** (+1.7)	95.2	**96.3** (+1.1)
K-700	top1	46.1	**49.1** (+3.0)	49.1	**53.5** (+4.4)	52.6	**56.5** (+3.9)
	top5	67.1	**72.7** (+5.6)	72.5	**74.2** (+1.7)	74.6	**76.8** (+2.2)
MiT	top1	24.9	**28.5** (+3.6)	28.2	**30.7** (+2.5)	31.5	**33.6** (+2.1)
	top5	50.1	**52.3** (+1.2)	53.5	**55.6** (+2.1)	57.4	**58.5** (+1.1)

(b) Comparisons for r(2+1)d original architectures with/out SRTG enabled

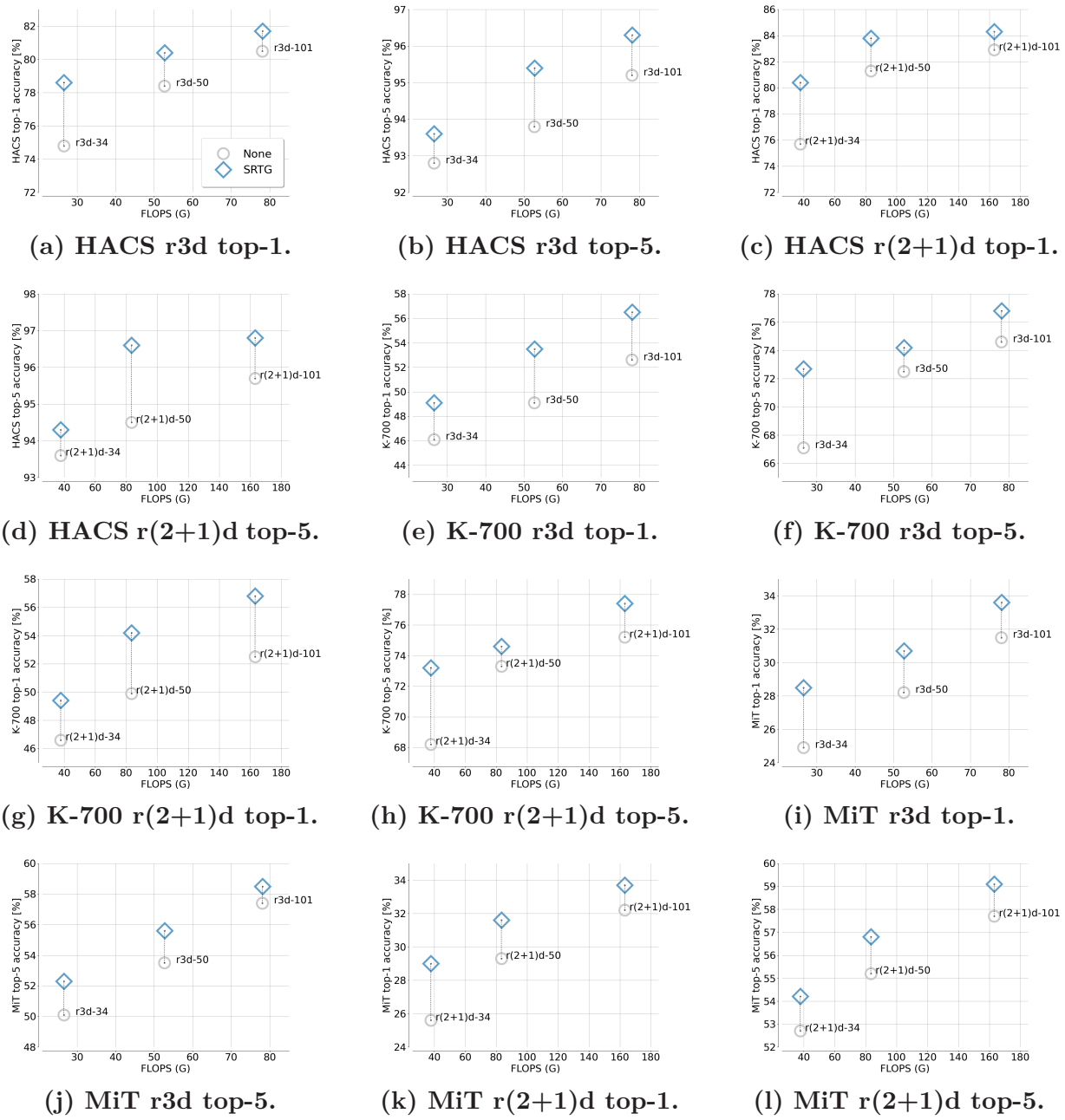
Dataset	accuracy (%)	r(2+1)d-34		r(2+1)d-50		r(2+1)d-101	
		None	**SRTG**	None	**SRTG**	None	**SRTG**
HACS	top1	75.7	**80.4** (+4.7)	81.3	**83.8** (+2.5)	82.9	**84.3** (+1.4)
	top5	93.6	**94.3** (+0.7)	94.5	**96.6** (+2.1)	95.7	**96.8** (+1.1)
K-700	top1	46.6	**49.4** (+2.8)	49.9	**54.2** (+4.3)	52.5	**56.8** (+4.3)
	top5	68.2	**73.2** (+5.0)	73.3	**74.6** (+1.3)	75.2	**77.4** (+2.2)
MiT	top1	25.6	**29.0** (+3.4)	29.3	**31.6** (+2.3)	32.2	**33.7** (+1.5)
	top5	52.7	**54.2** (+1.5)	55.2	**56.8** (+1.6)	57.7	**59.1** (+1.4)

4.4.5 Ablation studies

The scope of this section is the evaluation of models with and without SRTG. We demonstrate results in a pairwise fashion between the original baseline architectures and the proposed ones with SRTG. Experiments present a condensed view of these comparisons across the three large-scale datasets.

SRTG pairwise comparisons. In Tables 4.5a and 4.5b we detail full pairwise comparisons in terms of performance on HACS, K-700 and MiT for networks with and without SRTG. As presented in Table 4.5a, r3d configurations with SRTG can achieve an average +2.3%, +3.7% and +2.7% top-1 accuracy improvements on HACS, K-700 and MiT datasets, respectively. Similarly, for top-5 this corresponds to +1.3%, +2.8% and +1.5% increases in the accuracy rates for HACS, K-700 and MiT. For r(2+1)d architectures in Table 4.5b top-1 and top-5 accuracies on HACS are improved on average by +2.9% and +1.3%, respectively. Similar increases are also observable for K-700 and MiT datasets with +3.8% and +2.4% improvement in their top-1 and +2.8%, +1.5% for their top-5 accuracies. The extended experimentation on both network architectures and datasets show that the inclusion of SRTG modules can significantly benefit the overall performance of spatio-temporal models without direct implications on the overall architecture.

Comparison visualisations. The accuracy improvements are visualised in Figure 4.7 where the full pairwise comparisons are shown with respect to the number of FLOPs. Networks that include SRTG are denoted with blue while networks that do not are denoted



(a) HACS r3d top-1. (b) HACS r3d top-5. (c) HACS r(2+1)d top-1.

(d) HACS r(2+1)d top-5. (e) K-700 r3d top-1. (f) K-700 r3d top-5.

(g) K-700 r(2+1)d top-1. (h) K-700 r(2+1)d top-5. (i) MiT r3d top-1.

(j) MiT r3d top-5. (k) MiT r(2+1)d top-1. (l) MiT r(2+1)d top-5.

Figure 4.7. Accuracy to computational complexity trade-offs on HACS, K-700 and MiT for r3d and r(2+1)d architectures with and without SRTG.

with grey. Interestingly, the computational overhead of SRTG is $\ll 1\%$ of that of the entire architecture making the module computationally lightweight.

4.5 Feature transferability evaluation

One common aspect of CNNs is the utilisation of their weights trained on large-scale datasets and their transfer to smaller, typically more fine-grained, datasets. Given that the learned features can be generalised, this weight transfer process can significantly benefit performance. To further evaluate the generalisation capabilities of SRTG-enabled networks, we present results with the corresponding models being pre-trained on large

Table 4.6. Transfer Learning on UCF-101: Top-1 and top-5 accuracies after pre-training.

Model	Pre-training	top-1	top-5
I3D	K-400	92.4	97.6
TSM	K-400	92.3	97.9
ir-CSN-152	IG65M	95.4	99.2
MF-Net	K-400	93.8	98.4
SF r3d-50	ImageNet	94.6	98.7
SF r3d-101	ImageNet	95.8	99.1
r3d-101	HACS+K-700	95.8	98.4
r(2+1)d-101		95.5	98.7
SRTG r3d-34 **(ours)**		94.8	98.1
SRTG r3d-50 **(ours)**		95.7	98.5
SRTG r3d-101 **(ours)**		**97.3**	**99.5**
SRTG r(2+1)d-34 **(ours)**		94.1	97.8
SRTG r(2+1)d-50 **(ours)**		95.7	98.8
SRTG r(2+1)d-101 **(ours)**		**97.3**	99.2

datasets and fine-tuned on the smaller UCF-101 [235] and HMDB-51 [133]. Through this experimentation, we can eliminate biases that are based on the pre-training datasets used, while computing accuracy rates with respect to SRTG modules. For the feature transferability with SRTG ResNets, we only train the final class prediction layer while re-using the weights from the previously trained models in the convolutional layers. We additionally set the base learning rate to 0.01 and reduce it by a factor of ten at epochs 40 and 60 for a total of 80 epochs. Our choice of 80 epochs was based on not observing further improvements. We include experiments for SRTG ResNets on both UCF-101 and HMDB-51 with their respected networks being pre-trained on multiple datasets.

We present our transfer-learning results in Table 4.6. The smaller **SRTG r3d-34** and **SRTG r(2+1)d-34** networks outperform more complex architectures such as I3D and TSM with margins of 1.7–2.4% and 1.8–2.5% respectively. This shows that SRTG can increase accuracy rates and also that the combination of the HACS and K-700 datasets has notable benefits when fine-tuning networks. Both 3D and (2+1)D variants perform similar to MF-Net in top-1 with only +1.0% and +0.3% top-1 accuracy increments. Both **SRTG r3d-50** and **SRTG r(2+1)d-50** demonstrate accuracy rates similar to the top-performing models with only SlowFast-101 achieving better results on top-1 with marginal difference of -0.1%. The best performing model across our experiments was **SRTG r3d-101** with top-1 accuracy of 97.3% and 99.5% top-5 accuracy. The second highest performing model was the (2+1)D variant with a marginal decrease $\ll 0.1\%$ in top-1 accuracy compared to **SRTG r3d-101**. Compared to the non-SRTG counterpart, **SRTG r3d-101** demonstrates a +1.5% top-1 accuracy increment and a +1.1% for the top-5. Similarly, **SRTG r(2+1)d-101** improves by +1.8% for the top-1 and +0.5% for the top-5 accuracies compared to the r(2+1)d-101 baseline.

In our last set of comparisons in Table 4.7, we present results from weight initialisation of models on different datasets. The accuracy rates remain consistent for the pre-training datasets with the margins between datasets being $< 1.5\%$. The consistency in accuracy rates is because of the large sizes of these datasets, thus including a large number of

Table 4.7. Pre-train dataset comparisons on UCF-101 and HMDB-51.

Model	Pre-training	GFLOPs × views	UCF-101 top-1 (%)	HMDB-51 top-1 (%)
SRTG r3d-34	HACS	26.6×30	94.8	74.3
	HACS+K-700		95.8	74.2
	HACS+MiT		95.2	74.2
SRTG r(2+1)d-34	HACS	37.8×30	94.1	72.9
	HACS+K-700		94.6	73.2
	HACS+MiT		95.6	74.5
SRTG r3d-50	HACS	52.7×30	95.7	75.6
	HACS+K-700		96.8	76.0
	HACS+MiT		96.5	76.0
SRTG r(2+1)d-50	HACS	83.4×30	95.7	75.3
	HACS+K-700		96.0	75.7
	HACS+MiT		96.3	76.0
SRTG r3d-101	HACS	163.1×30	97.3	77.5
	HACS+K-700		97.4	78.0
	HACS+MiT		**97.6**	**78.4**

examples per class, and the overall robustness of our method. On average, the offset between pre-training models across datasets is 0.7% for UCF-101 and 0.5% for HMDB-51.

4.6 Discussion and conclusions

We have focusded on the challenges of human action and interaction recognition from spatio-temporal data. Feature changes in performance are strongly associated to a person's prepotent identity for an action. We have addressed temporal feature imbalances through a temporal feature calibration module named Squeeze and Recursion Temporal Gates (SRTG).

The proposed SRTG module calibrates spatio-temporal convolutional features based on the importance of the extracted features across the video sequence. By processing the local spatio-temporal activations with the use of recurrent cells (LSTMs) we capture multi-frame feature dynamics. These feature dynamics capture the correspondence of the local features across the entirety of the clip. The created activations, with respect to information over the video sequence, are evaluated in terms of its cyclic consistency with the convolutional features. We introduce a gate function for fusing together cyclic consistent volumes that show relevance between their temporally local and extended features. Equivalently, volumes that present large dissimilarities and are not cyclic consistent are not fused together. Through this, we achieve a degree of relevance between local and extended features.

We have evaluated our work on three large scale datasets: HACS, Kinetics-700 and Moments in Time, and over three ResNet architectures with either 3D or (2+1)D spatio-temporal convolutions. We have demonstrated competitive results to state-of-the-art architectures and in most cases outperform them. This also comes with negligible additional computations as our method is both memory and compute-efficient. Our ablation studies based on the original models without SRTG additionally validate our claims that the addition of SRTG to a 3D CNN architecture can yield further accuracy improvements.

We believe that the study of feature relevance across time can further benefit current

action recognition models by discovering temporal motions that are not constrained by the locality of kernels. Despite the alignment of the extracted spatio-temporal activations to information from the entire video sequence, limitations still exist. A dependency to the local spatio-temporal features is still present within the align features. A research direction in order to address this is the creation of kernels that can extract spatio-temporal features over different receptive field sizes. Such search can bring about a way to flexibly discover the relevance of features across multiple space-time sizes. This is explored in the following chapter.

Chapter 5

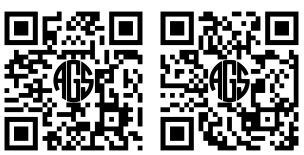
Time-Varying Convolutions for Video Understanding

This chapter addresses the locality of spatio-temporal convolutions by increasing their receptive fields. We focus on the creation of spatio-temporal activations that can capture features across different space-time modalities. The creation of these size-varying patterns is done through our novel *Multi-Temporal Convolutions*, that model both local and prolonged features within a single activation volume. The proposed modules are integrated in 3D CNNs by replacing their original 3D convolutions. We explore the benefits of activation volumes that represent time-varying features through testing across multiple datasets [1] .

5.1 Introduction

In the previous chapter we have focused on the relevance of local features in the context of the entire video. In our proposed method, we aligned features that can emphasise or suppress local features based on their relevance across the entire video segment used as input. By calibrating local features, their activations can reflect their overall significance as action descriptors. Although the inclusion of the feature dynamics across the video sequence within the local extracted patterns can be beneficial in terms of their representative capabilities, there is still a dependency on the fixed-size local patterns. Therefore, variations in the action and its feature durations that are not in the order of magnitude remain unaddressed. Based on this observation, we propose a method that can extract spatio-temporal patterns at different timescales. The recognition of temporal variations in this flexible manner can also maintain the spatial modelling power of the method.

The description of human actions based on their features strongly affects the temporal window in which they are performed [268]. The action identity that is used to describe the sequence of movements a person performs is affected by their complexity. The general observation is that complexity also impacts duration with longer and more complex actions generally requiring larger durations. Vallacher & Wagner [269] have described how action identities can include uncertainties or inadequacies based on different levels. For example, given the "basketball shot" example in Figure 5.1, if the sequence is broken into smaller parts, different action identities are discovered. However, not all local sets of movements are sufficient to fully describe the action performed. In different local segments of the video, the sets of movements can lead to different individual identities. Based on the example, the second group of frames can include more fine-grained identities such as "looking for



† ‡

[1] The code for the method is available at: https://git.io/JfuPi (†)
with the down-sampling operation available at: https://git.io/JL5zL (‡)

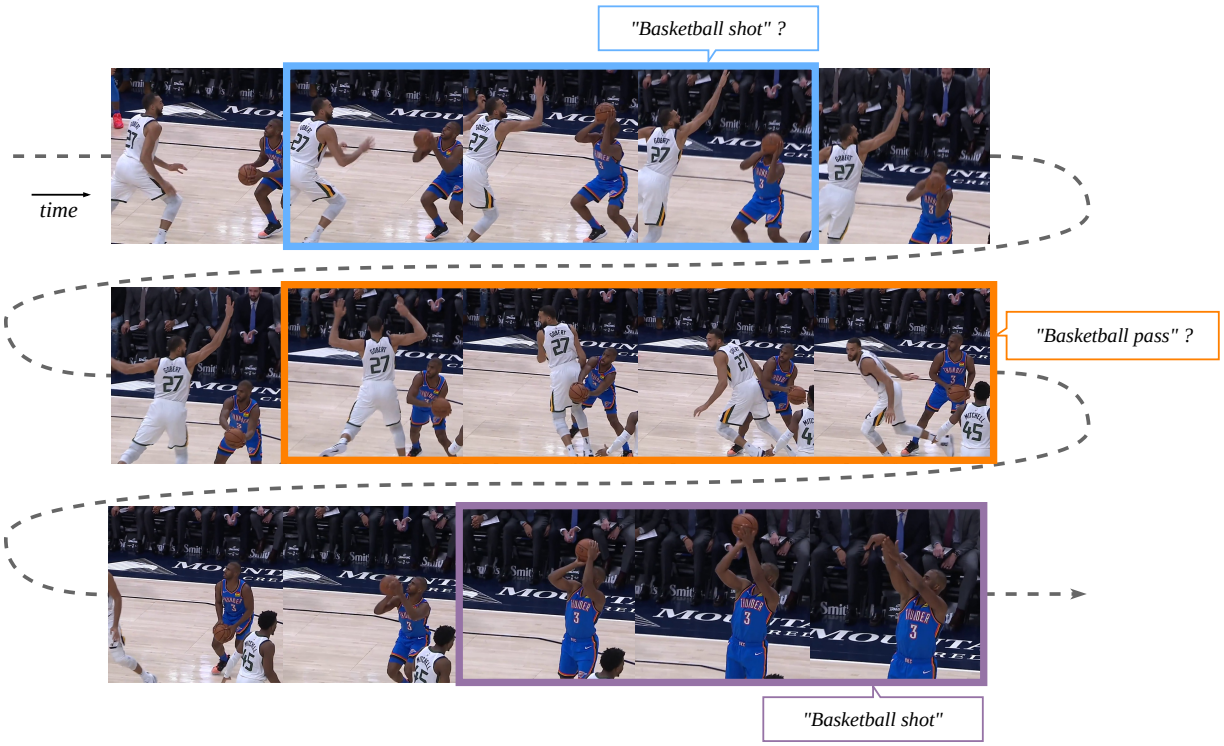


Figure 5.1. Temporal segments from class "basketball shot". Motions performed in different spatio-temporal local segments may relate to different action identities. The sole consideration of such features can lead to action misclassifications.

teammate", "moving the ball to lower torso" and "extending the ball for a bounce pass", which do not necessarily directly relate to "basketball shot" but could potentially relate to "basketball pass". If however, a larger part of the clip is shown, we can understand that the underlying action identity that could better describe the example is indeed "basketball shot". Based on this, we argue that actions have a degree of complexity which does not allow their description by solely considering small local patterns. Overall, action identities should be formed through the extraction of features across multiple temporal modalities. The inclusion of temporal segments of different durations can create features that are more descriptive of the action identity.

Considering that the extraction of strictly local motion patterns may not directly correspond to the target action identity, we study how the feature extraction process can be improved with the inclusion of patterns under different lengths to better capture high-level short-term and long-term identities associated with actions. We investigate the effects and relations between short local patterns and features of prolonged durations through our proposed Multi-Temporal Convolutions (*MTConv*) [244] utilising three branches for the extraction of spatio-temporal features across varying durations. The branch structure includes a *local branch* for spatio-temporal features within small spatio-temporal (local) segments, a *prolonged branch* for patterns of extended durations that include an inherited correspondence to the local features, and finally a *global aggregated feature importance branch* which further utilises SRTG to explore the feature dynamics of the aforementioned branches. Our proposed approach can address temporal patterns that are performed over different time-scales as well as align local and prolonged features in a shared feature space. Through the use of three branches for different temporal modalities, we focus on

the discovery of multi-temporal patterns by extending the convolutional receptive field to spatio-temporal locations of varying sizes.

In Section 5.2 we discuss related stream-based approaches for human action recognition in videos. In Section 5.3 we provide a complete description of the proposed convolution blocks and their layer architecture. Evaluations and tests based on our method with different settings are presented in Section 5.4. A discussion alongside conclusions are found in Section 5.5.

5.2 Temporal streams in video-based action recognition

We discuss two different approaches that are based on the creation of streams for addressing temporal information. The first set of methods consists of approaches that are based on the use of additional hand-crafted temporal features, such as optical flow, to represent temporal variations. The second group utilises information from frame sequences in streams based on either frame rates, groups of features or by separating low and high frequencies of activations.

5.2.1 Streams with hand-crafted temporal features

Optical flow is a straightforward approach for the representation of motion features in videos. Two-stream networks [229] combine spatial features extracted from video frames in the first stream and the supplementary temporal patterns of the second stream that use optical flow sequences. Information from the two streams is fused at the end to produce class predictions. Additional works have studied approaches for information fusion across the two streams [187]. Lateral connections between the two streams have also been proposed to include temporal information within the spatial pattern extraction process [65]. Other works have investigated the division of inputs into temporal segments [284] and spatial-based and temporal-based encoding of segments [55]. Tu *et al.* [265] proposed the utilisation of streams addressing different spatio-temporal regions concerning the entire regions for the actor in the video and the locations that movements are most prevalent. Each stream then uses both RGB frames and optical flow for the creation of a four-stream model.

Although these works provide a straightforward approach to explicitly model temporal information and their subsequent variations, the strong dependence on hand-coded optical flow data is limiting. As features relating to the spatial or temporal extent are learned separately, it prevents learning complex spatio-temporal features jointly in an end-to-end approach.

5.2.2 Temporal streams with 3D Convolutions

Considering the temporal modelling limitations of 2D two-stream CNNs, some works have adopted the stream-based approach with 3D convolutions [31]. Through this adaptation, inputs to each stream include stacks of either RGB frames extracted from a video sequence or a stack of optical flow representations for the cross-frame movements. Although the use of 3D convolutions in the RGB stream can effectively also extract temporal features by convolving stacks of frames, there is still a dependency on the representation quality of optical flow features in the second stream. To mitigate this issue, later works of

Feichtenhofer *et al.* [64] included a dual 3D model in which inputs were stacked RGB frames sampled over different temporal iteration steps within the same video sequence. The sampling-based slow and fast frame rates showed to be beneficial in the extraction of motion patterns of shorter and longer durations. This approach has further been employed for the creation of global feature paths using entire videos as inputs, and local feature paths with local spatio-temporal segments, using two separate network pathways [202]. Block-based approaches with octave convolutions [37] have also been used to model temporal variation in the frequency domain.

Despite the great promise that these methods have shown for the extraction of robust spatio-temporal features, the extraction of local spatio-temporal patterns is not done in relation to how informative they are at a global scale. The aim of our tri-branch method is to address within convolutional blocks, temporal feature disparities through the extraction of periodically varying space-time features and utilise the correlations between these features through a global attention mechanism.

5.3 Multi-Temporal convolutions

In this section, we describe multi-temporal convolutions (MTConvs) and their inner workings in terms of how information is processed. We then present the structure of the created blocks (MTBlocks, shown in Figure 5.2) that include the proposed modules.

Formally and in line with the previous chapter, layer activations are denoted as $\mathbf{a}_{(C\times T\times H\times W)}$ with C channels, T frames, H height and W width, respectively. Activations for each for the respective branches are denoted with $\mathbf{a}^{\mathcal{L}}$ for the local branch ($\mathcal{L}$) and $\mathbf{a}^{\mathcal{P}}$ for the prolonged branch ($\mathcal{P}$). Layers are indexed with l and indicated as $\mathbf{a}^{[l]}$ with $\mathbf{a}^{[l],\mathcal{L}}$, $\mathbf{a}^{[l],\mathcal{P}}$ in branch notation.

5.3.1 Local and Prolonged branches

A portion of layer l channels ($\widetilde{C}$) is used by each of the local and prolonged branches. To determine the number of channels for each branch, a channel ratio parameter δ is used. The channel size of the input activations to the layer ($\mathbf{a}^{[l-1]}$) is defined as C. Channels for the local branch ($C_{\mathcal{L}}$), based on ratio δ, use the lowest integer value approximation (through the homonym function denoted with $\lfloor floor \rfloor$). Respectively, the prolonged branch channels ($C_{\mathcal{P}}$) use the maximum integer value approximation ($\lceil ceil \rceil$). We overview how the branch channel distribution is performed in Equation 5.1:

$$\begin{gathered} C_{\mathcal{L}} + C_{\mathcal{P}} = \widetilde{C} \; where \\ C_{\mathcal{L}} = \lfloor \delta * \widetilde{C} \rfloor \; and \; C_{\mathcal{P}} = \lceil (1-\delta) * \widetilde{C} \rceil \end{gathered} \tag{5.1}$$

Inputs are first processed by the $\mathcal{L}$ and $\mathcal{P}$ branches. The local branch ($\mathcal{L}$) uses a single input with activation volumes $\mathbf{a}^{[l-1]}$ of size ($C\times T\times H\times W$). The prolonged branch ($\mathcal{P}$) uses a pair of inputs ($\mathbf{a}^{[l],\mathcal{L}}, \mathbf{a}^{[l-1]}$) with the first volume being the original layer input similar to the local branch. The second input is the output feature activations from the local branch $\mathcal{L}$ and of size ($C_{\mathcal{L}}\times T\times H\times W$). Dual inputs are used in the prolonged branch ($\mathcal{P}$) as spatio-temporal patterns of elongated duration and spatial sizes are strongly correlated with local features, which are extracted by $\mathcal{L}$. With prolonged features incorporating the complexity of local short-term ones, the $\mathcal{P}$ branch effectively operates over $C_{\mathcal{L}} + C_{in}$

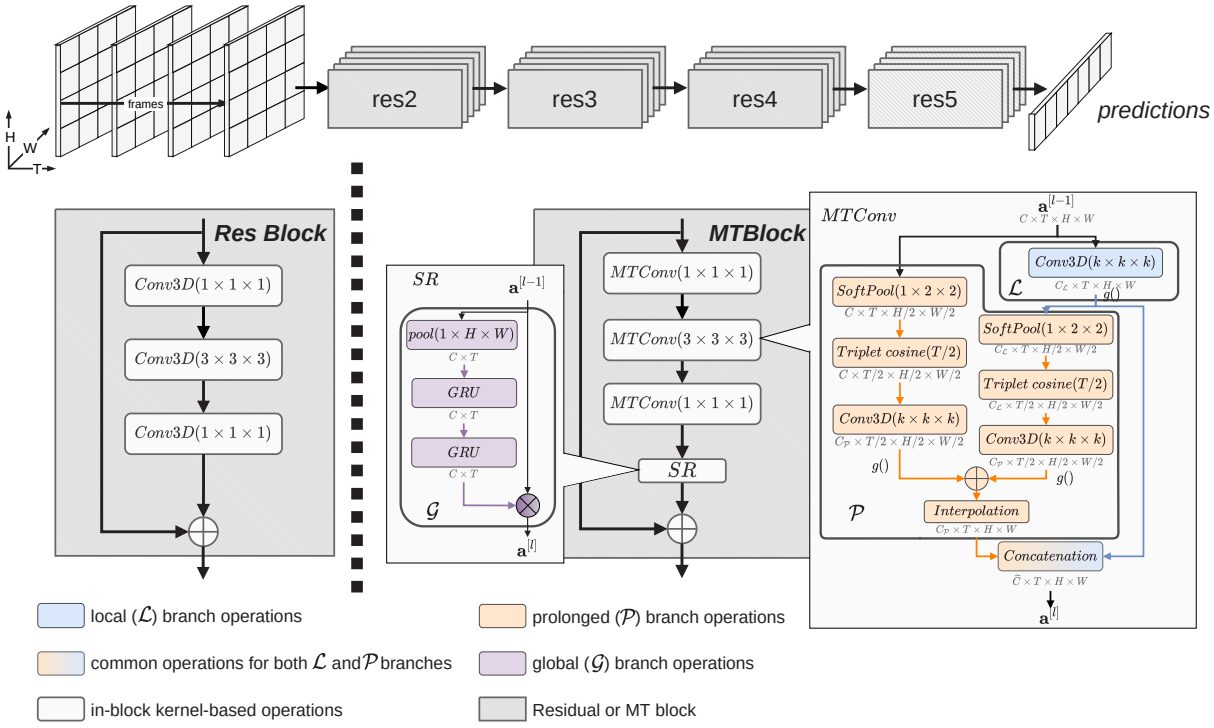


Figure 5.2. MTBlock structure. Utilising X3D [63] as backbone architectures, MTBlocks act as direct replacements for the Residual blocks (Res Block). In-block operations follow a sequence of Multi-Temporal Convolutions (MTConv) and a final Squeeze and Recursion (SR) feature alignment module as shown at the right side. We denote element-wise additions with $\oplus$ and element-wise multiplications with $\otimes$.

channels addressing the added complexity over $\mathcal{L}$. The $\mathcal{L}$ branch and $\mathcal{P}$ branch feature extraction process is summarised as seen in Equation 5.2:

$$\mathbf{a}^{[l]} = \mathcal{L}(\mathbf{a}^{[l-1]}) \frown (\mathcal{P}(\mathcal{L}(\mathbf{a}^{[l-1]}), \mathbf{a}^{[l-1]})) \tag{5.2}$$

where $\frown$ denotes the concatenation of the outputs from the two branches.

Local branch in MTConv. Short-term local motions in the input activations are extracted in the local branch. With layer input ($\mathbf{a}^{[l-1]}$) we use a 3D convolution followed by batch normalisation (BN) [105] and compute feature volume ($\mathbf{z}^{[l],\mathcal{L}}$) of $C_{\mathcal{L}}$ channels followed by non-linearity ReLU activation ($g()$). Unless otherwise stated, $g()$ refers to a ReLU activation. The final branch output takes the form of $\mathbf{a}^{[l],\mathcal{L}} = g(\mathbf{z}^{[l],\mathcal{L}})$.

Prolonged branch in MTConv. The extraction of patterns of extended duration is done over the prolonged branch. Information from the local branch ($\mathcal{L}$) and the layer input is used to extract features across larger spatio-temporal windows. The exploration of long-temporal features is done by reducing both inputs by a factor of two across their spatio-temporal dimensions. The size reduction is done by a factor of two as it provides a balanced trade-off between accuracy and computation. More aggressive strategies for spatio-temporal size reductions by larger factors might lead to significant information loss. Spatial downsampling over both inputs is done by their per-frame regional exponential maximum with *SoftPool* [246] with the activations produced being of size $T \times H' \times W'$ (where $H' = H/2$ and $W' = W/2$). The activations are then downsampled temporally

by a temporal triplet cosine frame selection to size $T' = T/2$. Detailed explanations for both methods are provided later in the section. The reduction in spatio-temporal size can increase the receptive fields of the prolonged kernels without requiring additional parameters while also reducing computational costs based on the size decrease of the activation volumes that are convolved. This inclusion of receptive fields, twice the duration of those in $\mathcal{L}$, allows for the exploration of temporal movements of larger spatio-temporal regions. Extended temporal patterns for inputs $\mathbf{a}^{[l-1]}$ and $\mathbf{a}^{[l],\mathcal{L}}$ are extracted by Conv3D operations followed by Batch Normalisation (BN). The complete process is formulated as:

$$\mathbf{a}^{[l],\mathcal{P}} = \mathcal{I}(g(\mathbf{z}^{[l],\mathcal{L}\to\mathcal{P}}) \oplus g(\mathbf{z}^{[l],\mathcal{P}})) \tag{5.3}$$

in which element-wise addition is denoted by $\oplus$ and $\mathcal{I}()$ is the spatio-temporal tri-linear interpolation of the volume from size $(T' \times H' \times W')$ to original size $(T \times H \times W)$. The feature volume $\mathbf{z}^{[l],\mathcal{L}\to\mathcal{P}}$ corresponds to the extracted patterns from the reduced input $\mathbf{a}^{[l],\mathcal{L}}$, while $\mathbf{z}^{[l],\mathcal{P}}$ corresponds to features extracted from $\mathbf{a}^{[l-1]}$:

$$\mathbf{z}^{[l],\mathcal{L}\to\mathcal{P}} = \mathcal{T}(\bar{\mathbf{a}}^{[l],\mathcal{L}}) * \mathbf{w}^{\mathcal{L}\to\mathcal{P}} \; and \; \mathbf{z}^{[l],\mathcal{P}} = \mathcal{T}(\bar{\mathbf{a}}^{[l]}) * \mathbf{w}^{\mathcal{P}} \tag{5.4}$$

with $\mathcal{T}()$ the triplet cosine frame selection for a spatially pooled volume $(\bar{\mathbf{a}})$. The convolutional weight vectors for the respective inputs are denoted as $\mathbf{w}^{\mathcal{L}\to\mathcal{P}}$ and $\mathbf{w}^{\mathcal{P}}$.

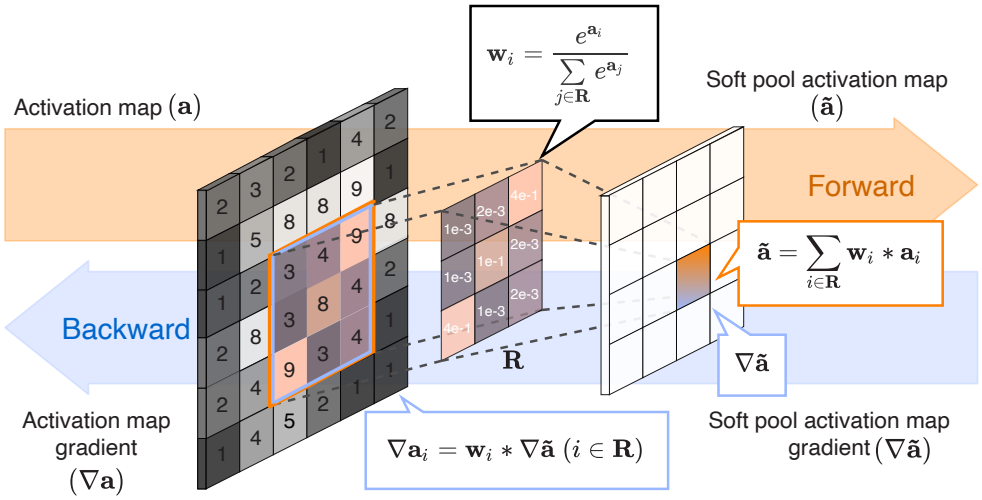


Figure 5.3. SoftPool spatial pooling. Forward operations (in orange) the kernel uses exponential maximum of each activation to produce a weighted sum of region $\mathbf{R}$. The weights are also used during the backward gradient $(\nabla\mathbf{a})$ calculations (in blue).

Prolonged branch spatial downsampling. *SoftPool* [246] uses soft-maximum approximation weighting to reduce the spatial dimensionality of the input. The method is based on the calculation of the softmax for each of the inputs within the kernel region. The softmax weights have a proportional effect to the output with higher-valued activations having a larger effect on the output than low-valued activations. A formulation based on input $\mathbf{a}$ and frame t with spatial region $\mathbf{R}$ of size $H \times W$ is shown in Equation 5.5.

$$\bar{\mathbf{a}}_{t,r} = \sum_{r\in\mathbf{R}} \frac{e^{\mathbf{a}_{t,r}} * \mathbf{a}_{t,r}}{\sum_{k\in\mathbf{R}} e^{\mathbf{a}_{t,k}}}, \; \forall \; t \in |T| \tag{5.5}$$

Prolonged branch temporal downsampling. Image-based pooling methods that have been extended for spatio-temporal data are based on the fusion of multiple frames. This fusion can degrade the quality of spatial details. The produced effects are similar to afterimages in which edges of objects within frames are less distinguishable as their cross-frame motions are joined together to a single frame. With our method being dependant on the preservation of such features, we instead use a temporal downsample method based on frame selection on the spatially-pooled activation volume ($\overline{\overline{\mathbf{a}}}$).

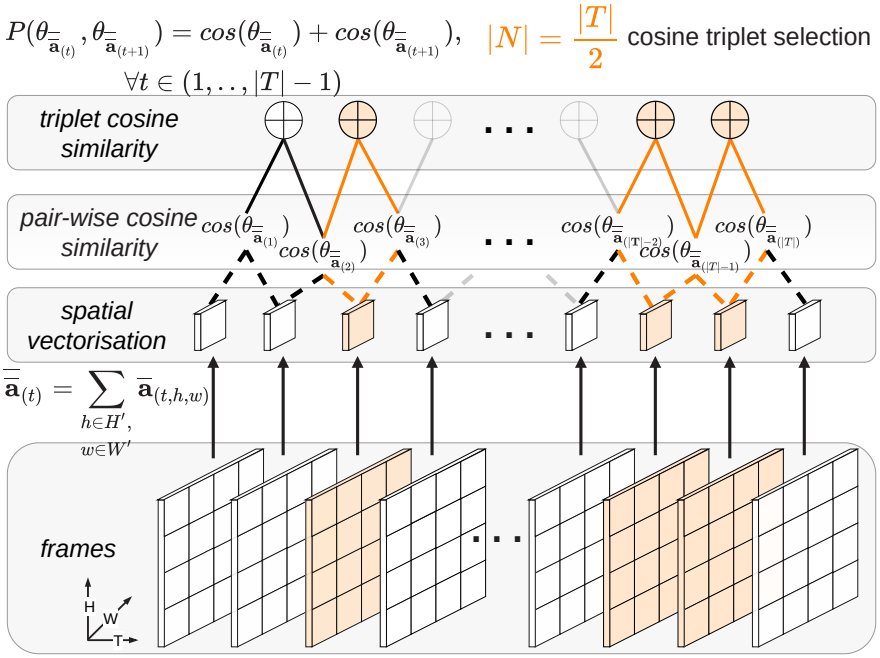


Figure 5.4. Temporal triplet cosine similarity pooling. Frames are selected based on their channel-wise sum ($\oplus$) of their per-frame pair cosine similarity ($cos(\theta_{\overline{\overline{\mathbf{a}}}_{(t)}})$) calculated from their spatially-summed volumes ($\overline{\overline{\mathbf{a}}}$)

The complete frame selection process is visualised in Figure 5.4. For each frame t the entire spatial regions is averaged to a single value for every channel. The activation volume produced ($\overline{\overline{\mathbf{a}}}$) contains frame-wise vectors that can be used to measure their per-pair, feature-wise similarity. This is done based on their magnitude and sum of their dot products:

$$cos(\theta_{\overline{\overline{\mathbf{a}}}_{(t)}}) = \frac{\sum_{c \in C} \overline{\overline{\mathbf{a}}}_{(t,c)} * \overline{\overline{\mathbf{a}}}_{(t+1,c)}}{\sqrt{\sum_{c \in C} \overline{\overline{\mathbf{a}}}^2_{(t,c)}} * \sqrt{\sum_{c \in C} \overline{\overline{\mathbf{a}}}^2_{(t+1,c)}}} \tag{5.6}$$

The cosine similarity pairs are then summed together for the creation of cosine similarity triplets ($P(\theta_{\overline{\overline{\mathbf{a}}}_{(t)}}, \theta_{\overline{\overline{\mathbf{a}}}_{(t+1)}}) = cos(\theta_{\overline{\overline{\mathbf{a}}}_{(t)}}) + cos(\theta_{\overline{\overline{\mathbf{a}}}_{(t+1)}})$). The produced triplet represents a concatenated view of how similar features in frame t are in comparison to features from the preceding frame $(t-1)$ and succeeding frame $(t+1)$. The frame selection then takes the form of selecting the frame locations (N) with the lowest triplet cosine similarity value focusing on the most informative frames:

$$\begin{gathered} \underset{\forall n \in N}{\operatorname{argmin}} P(\theta_{\overline{\overline{\mathbf{a}}}_{(n)}}, \theta_{\overline{\overline{\mathbf{a}}}_{(n+1)}}) = cos(\theta_{\overline{\overline{\mathbf{a}}}_{(n)}}) + cos(\theta_{\overline{\overline{\mathbf{a}}}_{(n+1)}}), \\ where\ N \subset T, |N| = |T/2| \end{gathered} \tag{5.7}$$

The cross-frame similarity is used for frame selection, instead of frame regions temporally fused together, the per-frame activations remain consistent over the produced temporally sub-sampled volume.

5.3.2 MTBlocks structure

Global aggregated feature importance. The concatenated activations of the local ($\mathcal{L}$) and prolonged ($\mathcal{P}$) branches are aligned based on the importance of each feature in the context of the entire video sequence. The role of the *global aggregated feature importance* branch ($\mathcal{G}$) is the creation of coherent activations based on averaged feature attention through *Squeeze and Recursion* [243] with GRU [40] recurrent cells. The branch operates over a vectorised version of the original volume pooled by its spatial dimensionality ($pool(\mathbf{a}^{[l-1]})$). The pooled volume is processed through a dual-layer recurrent sub-network for the discovery and amplification of globally-informative features. Initial refinement of salient features is done by the update gate ($\mathbf{z}_{(t)}$) that uses frame (t) input ($pool(\mathbf{a}^{[l-1]})_{(t)}$) and the previous ($t-1$) hidden state ($\mathbf{h}_{(t-1)}$) of the previous recurrent cell (for time $t-1$), through a sigmoid (σ) activation with weight $\mathbf{W}_z$ and bias $\mathbf{b}_z$:

$$\mathbf{z}_{(t)} = \{\sigma(\mathbf{W}_z * [\mathbf{h}_{(t-1)}, pool(\mathbf{a}^{[l-1]})_{(t)}] + \mathbf{b}_z)\} \tag{5.8}$$

The resulting update gate ($\mathbf{z}_{(t)}$) can be seen as a single operation that corresponds to LSTM's forget ($\mathbf{f}_{(t)}$) gate from Equation 4.1 and input ($\mathbf{i}_{(t)}$) gate from Equation 4.2.

Cell input $pool(\mathbf{a}^{[l-1]})_{(t)}$ and previous state outputs $\mathbf{h}_{(t-1)}$ also pass through a reset gate ($\mathbf{r}_{(t)}$), which uses weight ($\mathbf{W}_z$) and bias ($\mathbf{b}_z$), in order to ignore less time-consistent features:

$$\mathbf{r}_{(t)} = \{\sigma(\mathbf{W}_r * [\mathbf{h}_{(t-1)}, pool(\mathbf{a}^{[l-1]})_{(t)}] + \mathbf{b}_z)\} \tag{5.9}$$

In LSTMs this is done by computing the candidate values ($\widetilde{\mathbf{C}}_{(t)}$) in Equation 4.3 and then the cell state ($\mathbf{C}_{(t)}$) in Equation 4.4. GRUs do not distinguish between cell states ($\mathbf{C}_{(t)}$) and hidden states ($\mathbf{h}_{(t)}$), as only their hidden states $h_{(t)}$ are passed across temporal cells and used as cell outputs.

Both update and reset gates act in a complementary manner with the same inputs. A candidate hidden state is computed ($\widetilde{\mathbf{h}}_{(t)}$), based on the activations produced by the reset gate, through a *tanh* activation, and includes reduced influence from the previous state ($\mathbf{h}_{(t-1)}$) based on rate $\mathbf{r}_{(t)}$. The produced cell state is the fusion of a portion of the previous state ($\mathbf{z}_{(t)} * \mathbf{h}_{(t-1)}$) and the supplementary portion of the candidate hidden state ($(1 - \mathbf{z}_{(t)}) * \widetilde{\mathbf{h}}_{(t)}$):

$$\widetilde{\mathbf{h}}_{(t)} = tanh(\mathbf{W}_h * [\mathbf{r}_{(t)} * \mathbf{h}_{(t-1)}, pool(\mathbf{a}^{[l-1]})_{(t)}] + \mathbf{b}_h) \tag{5.10}$$

$$\mathbf{h}_{(t)} = \mathbf{z}_{(t)} * \mathbf{h}_{(t-1)} + (1 - \mathbf{z}_{(t)}) * \widetilde{\mathbf{h}}_{(t)} \tag{5.11}$$

This approach significantly simplifies the feature selection process in comparison to LSTMs in Equations 4.3 to 4.5. Figure 5.5 demonstrates the structural differences between the two cell types. The inclusion of hidden states ($\mathbf{h}$) in the activation maps ($\mathbf{a}^{[l]}$) is performed the same as for LSTMs.

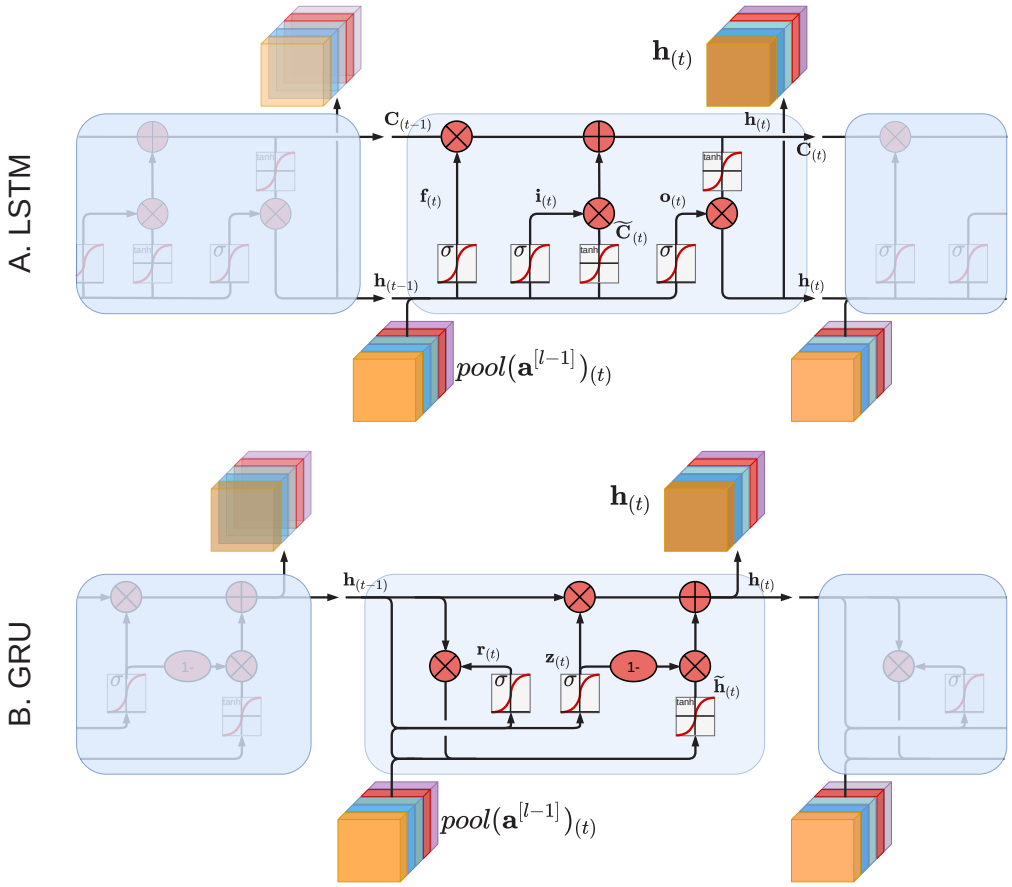


Figure 5.5. (A) LSTMs and (B) GRUs. Overview of both architectures and their sequential chain of cells. GRUs only exchange their hidden states ($\mathbf{h}_{(t)}$) across cells while LSTM compute both a cell state ($\mathbf{C}_{(t)}$) and a hidden state ($\mathbf{h}_{(t)}$).

5.3.3 Multi-Temporal Networks (MTNet)

We propose three MTNet architecture variants that use X3D [63] backbones, respectively X3D_S, X3D_M and X3D_L. These models vary in size and GFLOP usage, and we replace their Residual blocks and 3D Convs with the proposed MTBlocks and MTConvs, as shown in Figure 5.2. We denote our models as MTNet_S, MTNet_M and MTNet_L. The architectures follow a step-wise network and block expansion as recently proposed for video [63] and image-based models [203]. Details of the three proposed models in terms of the number of parameters and GFLOPs appear in Table 5.1.

5.4 Main results

The evaluation of our proposed method is done over the three MTNets of different sizes and on five action recognition benchmark datasets. We overview the experiment setting for our tests in Section 5.4.1. We compare against state-of-the-art models on Kinetics-400 in Section 5.4.2, on Moments in Time in Section 5.4.3, on Kinetics-700 (2020) in Section 5.4.4 and on HACS in Section 5.4.5. In Section 5.4.6 we perform additional ablation studies for different branch ratios (δ), recurrent cells and pooling strategies for the prolonged branches. Experiments for transfer learning on UCF-101 are presented in Section 5.4.7.

5.4.1 Environment settings

Our training environment is based on the settings that were also used by Feichtenhofer [63] for X3D networks. Some differences include the frame sampling process, the use of multigrid batch size schedule and spatial-data augmentations. We detail our choices below.

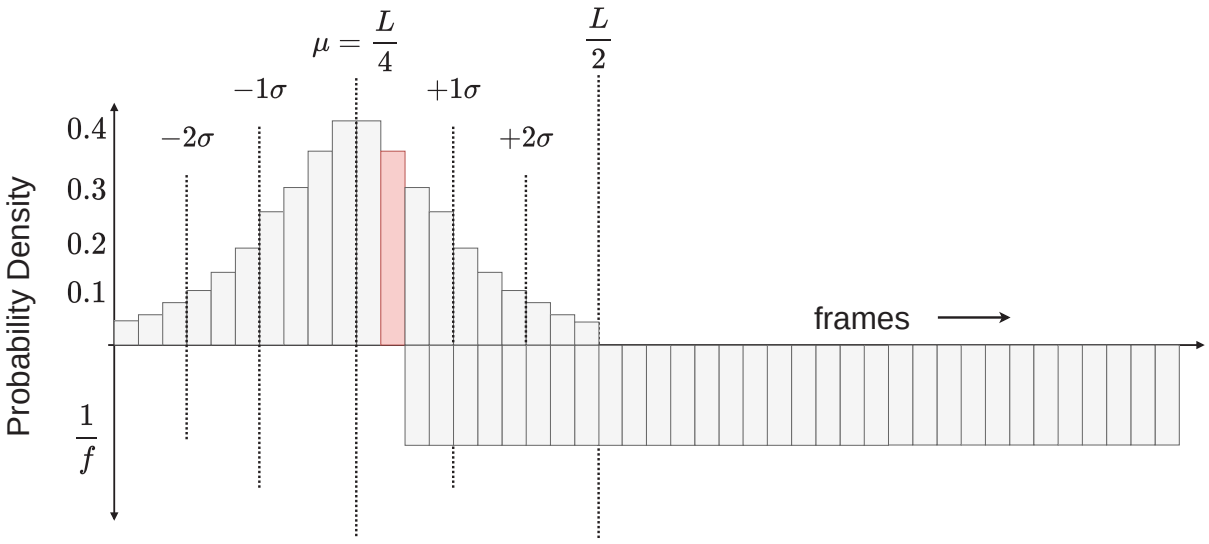


Figure 5.6. Frame sampling probability density. Top normal distribution, with mean ($\mu = \frac{L}{4}$) and standard deviation ($\sigma = \frac{L}{6}$), is used to determine the start frame location (s). The probabilities for the rest of the frames are evenly distributed based on a uniform distribution of frame probabilities $p = \frac{1}{f}$, where f is the remainder of frames $f = L - s$.

Interval selection. The frame selection is done through uniform random sampling. The selection of the starting frame is done based on the assumption of a normal probabilistic distribution between (0 - $\frac{L}{2}$) where L is the total length of the entire video sequence. This start frame initialisation is done to ensure that the majority of the middle frames will be selected, as in most datasets the primary actions are preformed in middle frames. The probability distribution of frames is shown in Figure 5.6. Based on the average clip length, for each dataset, we use equivalently sized temporal strides. The average clip length in HACS is 60 frames, and we thus use a stride of 2. For the Kinetics variants, average clip length is 250 frames so we set the temporal stride to 5. Similarly for MiT we use a temporal stride of 3 with average duration of 90 frames.

Computational inference. We report computational cost over two different measures. We first report the computational inference as the measures used in Section 4.4.1. This can express the computational costs (FLOPs) with respect to 10 clips with 3 spatial augmentations. The frame size used is 256×256. The inference time is then reported as the number of GFLOPs per spatio-temporal view (clips times crops). As our second measure, we report the inference cost in relation to latency (in msec).

Multigrid batch schedule. We use a multigrid scheme [292] for improvements in the training speeds. Similar to Section 4.4.1, mini-batch size sampling is done proportionally to the original batch size through the condition that $b \times t \times h \times w = B \times T \times H \times W$, in which (b, t, h, w) represent the scaled batch size, number of frames, height and width dimensions. Equally, (B, T, H, W) are the original dimensions. This is done in order to maintain equivalence between the computational costs of the new and previous branches.

Training parameters for MTConv. We train our models on HACS for which we use *random initialisation without pre-training ("from scratch").* We additionally set the initial mini-batch size to 16 clips per GPU (with the total mini-batch size of 64)[2]. Frame selection is done by the aforementioned frame sampling procedure. For the spatial domain, we interpolate all frames so that the shorter side is of size 320. We then perform a random crop of 256×256. We note that the input frame size difference between MTNet_S and MTNet_M is part of the architectural parameters as with X3D_S and X3D_M. In our case, inputs of 256×256 size in MTNet_S are resized to 182×182 based on the $\sim 30\%$ decrease as proposed in [63]. All experiments were performed over 400 epochs with momentum of 0.9 and weight decay of 5×10^{-5}. Unless stated otherwise, all of our models use $\delta = 0.875$. We explain this choice in Section 5.4.6.

Cosine-based learning rate schedule on X3D. Similarly to relevant works [64], as well as the suggested training recipe for X3D networks [63], we use a cosine annealing learning rate decay schedule [155]. This schedule is based on periodic decrease and increase of the learning rate through the use of a cosine function. The learning rate lr_n for iteration n is calculated as $lr_n = lr_0 * 0.5[cos(\frac{n}{n_{max}}\pi) + 1]$ in which n_{max} is the total number of iterations and lr_0 is the starting learning rate. We use $lr_0 = 1.16$ and learning rate warmup for the first 8k iterations similar to [63, 64].

Spatial data augmentation. We use the same data augmentation strategies as for SRTG in Section 4.4.1. We set the sequential data augmentation probabilities to 70% to reduce data pre-processing times. The spatial augmentations include blurring, brightness increases, contrast increase and decrease as well as geometrical transformations.

5.4.2 Results on Kinetics-400 (K-400)

We compare the proposed architectures (MTNets) in Table 5.1 with state-of-the-art results. We additionally present results for the two top-performing SRTG networks with 3D and (2+1)D convolutions. This allows for a direct evaluation in order to determine the improvements of MTConvs over the architectures introduced in Chapter 4. Compared to the current top performing X3D-XL [63], our largest $\textbf{MTNet}_L$ architecture produces comparable performance (-1% top-1 and -0.7% top-5 lower accuracies), while considerably reducing the number of multiply-addition operations (measured in GFLOPs). $MTNet_L$ can reduce the computations by a factor of $\times 2.75$ in relation to X3D-XL. In general, the performance expected from $\textbf{MTNet}_L$ is similar to that of SlowFast-101 (SF) [64] which utilised more than 12 times the number of GFLOPs compared to our model. Considering the computational requirements, $\textbf{MTNet}_L$ largely outperforms MFNet [39] with similar number of GFLOPs, by +5.3% in top-1 and +2.8% in top-5 accuracies.

The smaller $\textbf{MTNet}_M$ shows accuracy rates close to *Channel-Separated Network* (ip-CSN-101) [260], *Temporal Adaptive Module (TAM)* ResNet-50 and the ResNet 50 variant of SlowFast, while being significantly more efficient than any of the alternative architectures. Specifically, $\textbf{MTNet}_M$ is 9.4 times more efficient than ip-CSN-101, 9.7 times more than TAM and 7.5 times more efficient than SF-50. This reduction in multiply-add operations can significantly benefit the required training times in relation to the compared networks. Given its modest number of multiply-add operations (FLOPs), $\textbf{MTNet}_M$ can still outperform R(2+1)D ResNet101 [261] and *Temporal Shift Module* (TSM) [148].

[2]All experiments were performed with mixed half-precision (float16) for improved memory utilisation efficiency. Batch Normalisation is computed with single-precision (float32) for scaling stability. SoftPool is also computed with single precision to avoid gradient underflows.

Table 5.1. Comparison with K-400 state-of-the-art. For consistency with previous testing methods, we report the model complexity as the GFLOPs per single clip view × the number of clips with spatial cropping of size 256×256.

Model	Input	Backbone	top-1	top-5	GFLOPs × views	Params
R(2+1)D [261]	16×224^2	ResNet101	62.8	83.9	152×115	63.6M
I3D [31]	16×224^2	InceptionV1	71.6	90.0	$108\times N/A$	12M
MF-Net [39]	16×224^2	ResNet50	72.8	90.4	11.1×50	**8.0**M
TSM [148]	16×224^2	ResNet50	74.7	91.4	65×10	24.3M
ip-CSN-101 [260]	8×224^2	ResNet101	76.7	92.3	83.0×30	24.5M
TAM [153]	16×256^2	ResNet50	76.9	92.9	86×12	25.6M
SF-50 [64]	$(32,4)\times224^2$	ResNet50	77.0	92.6	65.7×30	34.4M
ip-CSN-152 [260]	8×224^2	ResNet152	77.8	92.8	108.8×30	32.8M
SF-101 [64]	$(32,4)\times224^2$	ResNet101	77.9	93.5	213×30	53.7M
X3D-XL [63]	16×224^2	ResNet(X3D)	**79.1**	**93.9**	48.4×30	11.0M
SRTG r3d-101 [243] **(ours)**	16×224^2	ResNet101	73.2	91.3	78.1×30	107.1M
SRTG r(2+1)d-101 [243] **(ours)**	16×224^2	ResNet101	73.8	92.0	163.1×30	105.3M
MTNet_S [246] **(ours)**	16×256^2	ResNet(X3D)	74.8	92.1	$\mathbf{5.8\times30}$	25.8M
MTNet_M [246] **(ours)**	16×256^2	ResNet(X3D)	76.6	92.5	8.8×30	25.8M
MTNet_L [246] **(ours)**	16×256^2	ResNet(X3D)	78.1	93.2	17.6×30	50.1M

Our smallest MTNet network (**MTNet**$_S$) performs on par with TSM while having the lowest number of GFLOPs from all tested networks. Despite the significant reduction of GFLOPs, **MTNet**$_S$ can still perform better than both MFNet and I3D [31] networks. We note that the proposed MTNet architectures are the only family of spatio-temporal architectures within the range of sub-20 GFLOPs that produce accuracy rates similar to that of the top-performing, and significantly more expensive, state-of-the-art models.

The tested SRTG variant (**SRTG r(2+1)d-101**) was also trained with the parameters of Section 4.4.1. The network demonstrates the ability to improve the accuracies in comparison to the originally proposed R(2+1)D [261]. The inclusion of SRTG modules for temporal channel dynamics calibration can produce accuracies similar to those of TSM and MF-Net. However, compared to the smaller MTNet_S the accuracy benefits of extracting spatio-temporal patterns over varying sized space-time locations are visible.

We additionally include in our experiments **SRTG r3d-101**. We show that the expected performance is similar to that of TSM and the previously mentioned **SRTG r(2+1)d-101**. Based on the overall simplicity of the architecture, the accuracy improvements with the inclusion of SRTG are notable. In comparison to the tested MTNet sizes, **SRTG r(2+1)d-101** shows both lower performance (1.6–4.9%) as well as substantially larger computational requirements. The number of required GFLOPs is equivalent to 4.43–13.4 times that of the MTNets variants.

We present a visual representation of the accuracies in K-400 with respect to the computational complexity of each model (GFLOPs) and the number of parameters in Figure 5.7. MTNets can better balance the trade-off between computational complexity and accuracy.

We perform additional tests to better understand how each spatio-temporal strategy

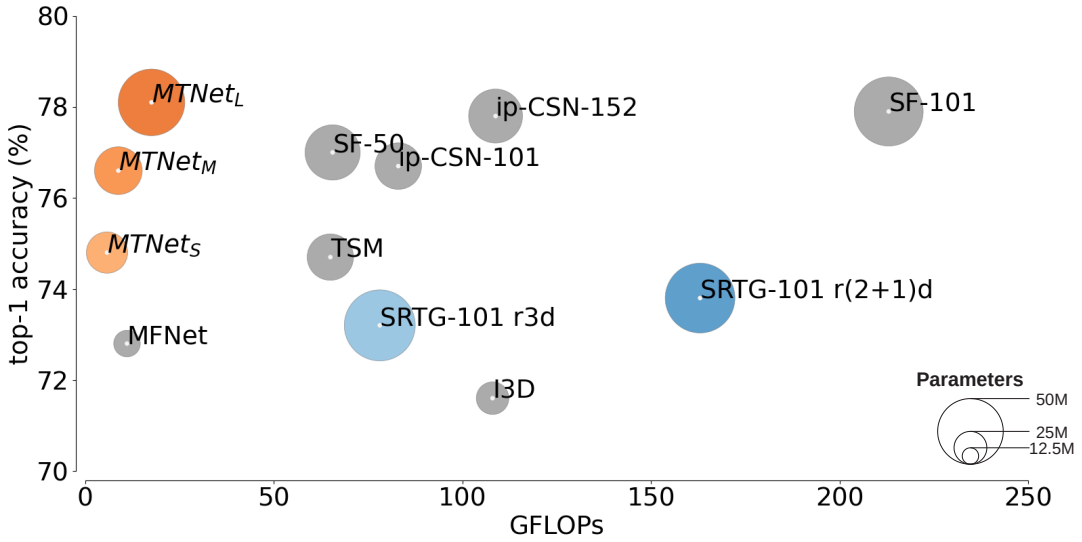


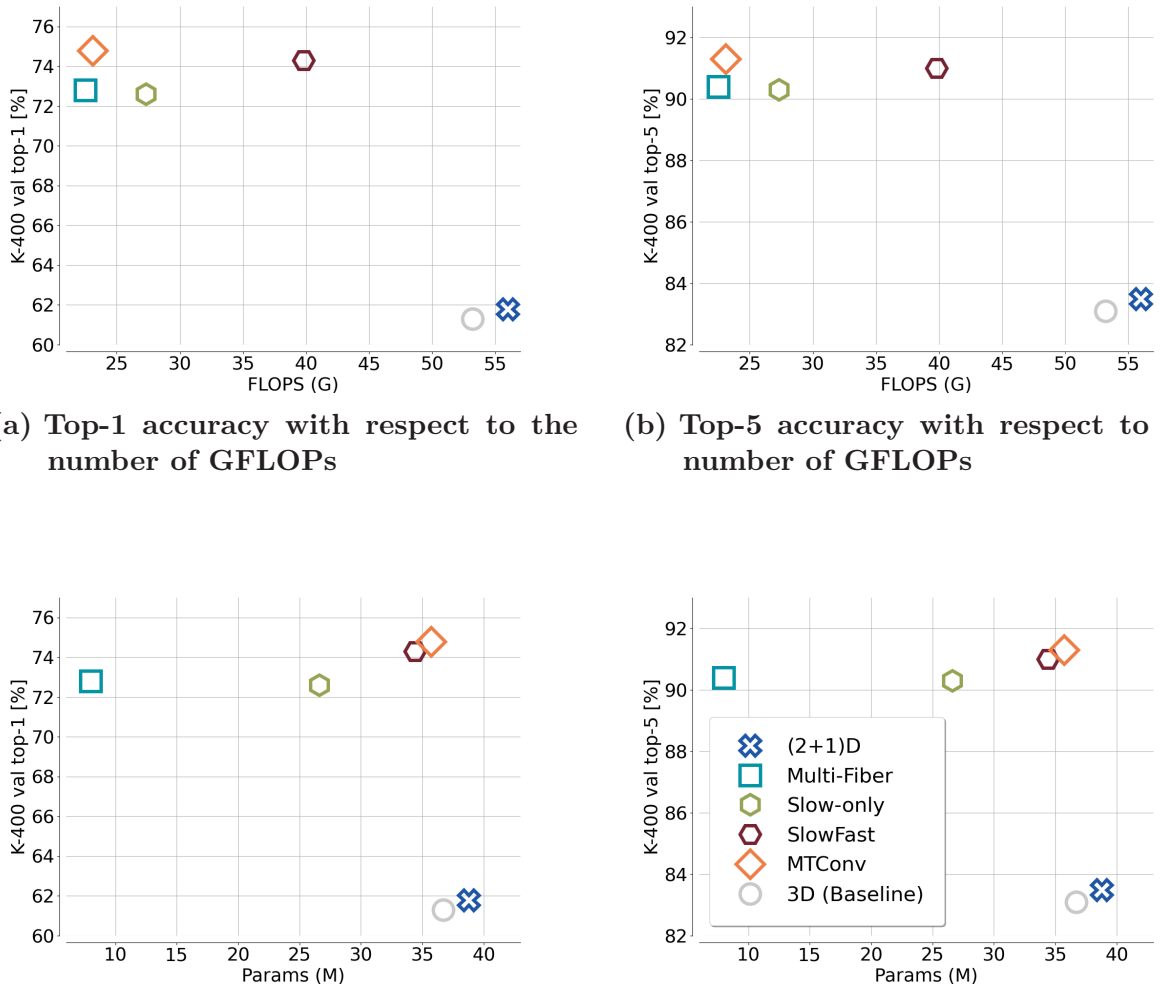
Figure 5.7. Accuracy to GFLOP/parameters comparison on K-400. The size of the blob corresponds to the number of parameters that the network uses. As shown, MTNets are significantly more efficient in comparisons to state-of-the-art counterparts while maintaining competitive accuracy results.

Table 5.2. Spatio-temporal method comparison on K-400. Accuracy rates for spatio-temporal 3D conv methods on Kinetics with ResNet-50 as backbone.

Method	top-1	top-5	FLOPs (G)	Params (M)
(Baseline) 3D [89]	61.3	83.1	53.2	36.7
(2+1)D [261]	61.8 (+0.5)	83.5 (+0.4)	56.0 (+2.8)	38.8 (+2.1)
Multi-Fiber [39]	72.8 (+11.5)	90.4 (+7.3)	**22.5** (−30.7)	**8.0** (−28.7)
Slow-only [64]	72.6 (+11.5)	90.3 (+7.2)	27.3 (−25.9)	26.6 (−10.1)
SlowFast [64]	74.3 (+13)	91.0 (+7.9)	39.8 (−13.4)	34.4 (−2.3)
MTConv **(ours)**	**74.8** (+13.5)	**91.3** (+8.2)	23.1 (−30.1)	35.7 (−1.0)

affects the accuracy. We compare all the aforementioned methods with a baseline ResNet-50 architecture in Table 5.2. The use of a common backbone architecture for all methods is done in order to provide a standardised comparison across the different spatio-temporal modelling schemes from the literature. The direct replacement of 3D convolutions with MTConvs demonstrate a significant performance improvement with +13.5% in top-1 and +8.2% in top-5 accuracies over 3D Convs. This also includes +(0.5–13)% top-1 and +(0.3–7.8)% top-5 accuracy performance increments over the compared spatio-temporal convolution schemes. MTConvs perform similar to SlowFast convolutions which follow the same trend as in Table 5.1, while performing better than the alternative Multi-Fibre or the SlowFast supplementary Slow-only approaches. For both results in the top-1 and top-5 accuracies we do not notice significant improvements between 3D and (2+1)D.

Computation-wise, MTConvs are also less complex with notable reductions in terms



(a) **Top-1 accuracy with respect to the number of GFLOPs**

(b) **Top-5 accuracy with respect to the number of GFLOPs**

(c) **Top-1 accuracy with respect to the number of parameters**

(d) **Top-5 accuracy with respect to the number of parameters**

Figure 5.8. Spatio-temporal data processing methods accuracy/complexity trade-offs on K-400 based on Table 5.2. Comparisons include the R(2+1)D model, MFNet, Slow-only path from SlowFast, SlowFast and the Proposed MTConvs. Baseline architecture is a 50-later ResNet.

of the number of multiply-addition operations with a 56% decrease in operation numbers in comparison to standard 3D convolutions. Although the performance of MTConvs and SlowFast is similar, their margin in number of GFLOPS is significant with MTConvs requiring approximately 1.7 times less multiply-add operations. Additionally, MTConvs are within the same margin as other efficient methods such as Multi-Fibre which employ group-based convolutions while outperforming them in both top-1 and top-5. As shown in Figure 5.8, the decrease in GFLOPs for MTNets is unrelated to the number of parameters.

5.4.3 Results on Moments in Time (MiT)

We demonstrate the top-1 and top-5 accuracies achieved by state-of-the-art models on the Moments in Time (MiT) dataset in Table 5.3. We denote pre-defined architectures with "Fixed" and models created by Neural Architecture Search with "NAS" [323]. NAS-based

Table 5.3. Comparison with MiT state-of-the-art. Models denoted with (†) include additional information input sources.

Model	Arch. size	top-1	top-5
EvaNet [195]	NAS[323]	31.8	N/A
AssembleNet [213]		34.3	**62.7**
MF-Net [39]	Fixed	27.3	48.2
I3D [31]		29.5	56.1
CoST [140]		32.4	60.0
SoundNet [169] †		7.6	18.0
TSN+Flow [169] †		15.7	34.7
SRTG r3d-34 [243] **(ours)**		28.5	52.3
SRTG r3d-50 [243] **(ours)**		30.7	55.6
SRTG r3d-101 [243] **(ours)**		33.6	58.5
SRTG r(2+1)d-34 [243] **(ours)**		29.0	54.2
SRTG r(2+1)d-50 [243] **(ours)**		31.6	56.8
SRTG r(2+1)d-101 [243] **(ours)**		33.7	59.1
MTNet$_M$ [244] **(ours)**		34.5	58.6
MTNet$_L$ [244] **(ours)**		**35.2**	59.3

models have optimised structures in order to fit the data better. Our best performing architecture **MTNet**$_L$ outperforms current state-of-the-art models in top-1 accuracy with 35.2% while having a slightly lower top-5 rate (-3.4%) than the AssembleNet [213]. Notably, the comparisons also include models that exploit supplementary information. Specific architectures further utilise optical flow [169] and sound-based [169] inputs. In comparison, all of our models (MTNet and SRTG-enabled r3d and r(2+1)d) are trained solely on stacked RGB frames. Interestingly, the smaller **MTNet**$_M$ performs similar to learned architectures such as AssembleNet [195] while having a pre-defined architecture.

5.4.4 Results on Kinetics-700 (K-700)

We additionally evaluate our models on the 700-variant of the Kinetics dataset. Our results achieved are presented in Table 5.4. Our best performing architecture **MTNet**$_L$ outperforms the tested models from the literature by significant margins. In comparison to the previous top-performing SlowFast-101, **MTNet**$_L$ shows an accuracy improvement of 6.0% for top-1 accuracy and 6.9% for top-5 accuracies. In relation to the similarly complex MF-Net, we show +9.0% top-1 and +10.7% top-5 accuracies. In comparison to SRTG networks, **MTNet**$_L$ demonstrates accuracy increases in the range of 1.5–6.7% for top-1 and 6.5–14.2% for top-5. We also report the accuracy rates of **MTNet**$_M$ with performance comparable to that of SlowFast-101 while also demonstrating a +1.1% improvement in the top-1 accuracy. Overall, **MTNet**$_M$ is the most lightweight architecture that we have tested while being the second best performing. In relation to SRTG networks, the accuracy

Table 5.4. Comparison with K-700 state-of-the-art. GFLOP calculation is similar to that in Table 5.1. (*) indicates reproduced models.

Model	Pre-train	GFLOPs × views	top-1	top-5
I3D [31]	K-600	$108 \times N/A$	53.0	69.2
TSM [148] *	HACS	65.0×10	54.0	72.2
MF-Net [39] *		11.1×50	54.3	73.4
ir-CSN-101 [260] *		63.6×10	54.7	73.8
SF-50 [64] *		65.7×30	56.2	75.7
SF-101 [64] *		213.0×30	57.3	77.2
SRTG r3d-34 [243] **(ours)**		26.6×30	49.1	72.7
SRTG r3d-50 [243] **(ours)**		52.7×30	53.5	74.2
SRTG r3d-101 [243] **(ours)**		78.1×30	56.5	76.8
SRTG r(2+1)d-34 [243] **(ours)**		37.8×30	49.4	73.2
SRTG r(2+1)d-50 [243] **(ours)**		83.4×30	54.2	74.6
SRTG r(2+1)d-101 [243] **(ours)**		163.1×30	56.8	77.4
MTNet_M [244] **(ours)**		8.8×30	58.4	77.6
MTNet_L [244] **(ours)**		17.6×30	**63.3**	**84.1**

margins range between 1.5–9.3% depending on the model size. Based on this, it is evident that accuracy improvements that can be achieved through MTNet as well as the additional reductions in computational costs.

5.4.5 Results on HACS

We provide results over the Human Action Clips Segments (Clips) dataset (HACS). We present the performance achieved for different models in Table 5.5. Due to the lack of results reporting on HACS because of its recency, all results are re-implemented on our own machine and thus also provide a standard benchmark. Datasets that are utilised for weight initialisation are denoted by the *Pre* column with the sources that models have been imported from described in the sub-caption. Latency times in milliseconds (msecs) are computed as the inference of a single batch during the forward and backward pass (inf_f, inf_b) divided by the number (b) of clips in the batch ($\downarrow F$ ($\frac{inf_F}{b}$), $\uparrow B$ ($\frac{inf_B}{b}$)). The inference calculations are all performed with batch sizes of 32. For the MFNet [39], TAM [153], SF-101 [64] and X3D-L [63] models, their weights are initialised based on the K-400 dataset. R3D and R(2+1)D models with their 34, 50 and 101 variants are pre-trained on K-700. Lastly, we include in our comparisons *Channel-Separated Spatio-Temporal Convolutions* (CSN) that are pre-trained on the IG65M dataset that is not publicly available. The two tested CSN networks variants are an *interaction-preserved* channel-separated network (ip-CSN) and *interaction-reduced* Channel-Separated Network (ir-CSN). The ip-CSN network replaces 3×3×3 convolutions with separable 3D convolutions,

Table 5.5. Action recognition model comparisons on HACS. Weight initialisation sources are denoted by their respective indicators.

Model	Pre	Accuracy		GFLOPs	Params	Latency
		top-1	top-5	× views	(M)	(↓F / ↑B)
MF-Net [39]†	K-400	78.3	94.6	11.1×50	8.0	32.8/236.0 [3]
TAM [153]†		82.2	95.2	86×12	25.6	42.1/165.3
SF-101 [64]†		83.7	96.8	65.7×30	53.7	39.3/125.1
X3D-L [63]†		85.8	96.1	24.8×30	**6.1**	73.6/ 457.4 [3]
r3d-34 [119]*	K-700	74.8	92.8	26.6×30	63.7	32.7/74.1
r3d-50 [119]*		78.4	93.8	52.6×30	36.7	28.2/87.7
r3d-101 [119]*		80.5	95.2	78.0×30	69.1	41.6/110.2
r(2+1)d-34 [119]*		75.7	93.8	37.8×30	61.8	40.8/152.0
r(2+1)d-50 [119]*		81.3	94.5	83.3×30	34.8	33.2/128.7
r(2+1)d-101 [119]*		82.9	95.7	163.0×30	67.2	49.9/163.6
ir-CSN-101 [260]†	IG65	83.8	93.8	63.6×10	22.1	51.4/461.2 [3]
ip-CSN-101 [260]†		84.1	93.9	63.6×10	24.5	64.3/512.6 [3]
SRTG r3d-34 [243] **(ours)**	-	78.6	93.6	26.6×30	83.8	35.2/80.6
SRTG r3d-50 [243] **(ours)**	-	80.3	95.5	52.7×30	56.9	31.8/96.9
SRTG r3d-101 [243] **(ours)**	-	81.6	96.3	78.1×30	107.1	49.2/131.6
SRTG r(2+1)d-34 [243] **(ours)**	-	80.4	94.3	37.8×30	82.1	46.3/157.0
SRTG r(2+1)d-50 [243] **(ours)**	-	83.8	96.6	83.4×30	55.0	37.6/141.5
SRTG r(2+1)d-101 [243] **(ours)**	-	84.3	96.8	163.1×30	105.3	58.9/172.2
MTNet$_S$ [244] **(ours)**	-	80.7	95.2	5.8×30	25.8	50.8/199.6 [3]
MTNet$_M$ [244] **(ours)**	-	83.4	95.9	8.8×30	25.8	62.8/216.3 [3]
MTNet$_L$ [244] **(ours)**	-	**86.6**	**96.7**	17.6×30	50.1	98.3/513.7 [3]

† models and weights from official repositories.
* re-implemented models and weights.

while ir-CSN replaces them with depth-wise 3D convolutions.

Our **MTNet**$_S$ performs similarly to both r3d-101 [119] and r(2+1)d-50 [119], while not being pre-trained on another dataset nor including their large number of parameters. Additionally, **MTNet**$_S$ shows to perform better than MF-Net which requires roughly 1.9 times more GFLOPs. The latency times of **MTNet**$_S$ are similar to those of SRTG r3d-101 and SRTG r(2+1)d-101. This is owing to the use of channel-separated convolutions which require additional $O(n \times l)$ time compared to 3D convolutions, where n is the number of groups of channels and l is the number of channel-separated convolution layers. The larger **MTNet**$_M$ shows an overall increase in performance compared to **MTNet**$_S$ with +2.7% and +0.7% top-1 and top-5 accuracies. The achieved accuracies are similar to those of the SlowFast-101 and ir-CSN-101 models with only a fraction of the parameters and compute and without pre-training. Latency times follow an increasing trend compared to **MTNet**$_S$. Compared to X3D-L, **MTNet**$_L$ shows a +0.8% for the top-1 and +0.6%

[2]Marked models are based on the use of Group/Channel-separated 3D Convolutions. X3D-L and MTNet models use the patched version of Channel-separated 3D convolutions in PyTorch [189] 1.9.x while the other implementations are based on the channel-wise $O(n \times l)$ complex implementation. At the time of writing, conversions are not possible due to weights being only available in specific versions.

top-5 accuracy improvements while having $\sim$29% fewer GFLOPs. In our tests, **MTNet**$_L$ achieved the best accuracy rates out of the tested models significantly outperforming SF-101 and ir/ip-CSN-101 without any previous kernel initialisation. We also note that partial latency overheads are also due to our proposed frame downsampling method based on cosine similarity triplets. Although the number of computations is not significantly affected, tensor slicing and indexing is computationally slow during parallelisation.

5.4.6 Ablation Studies

The scope of this section is to perform ablation studies on HACS for our proposed models. For MTConvs, we compare results from different ratios (δ) for the local ($\mathcal{L}$) and prolonged ($\mathcal{P}$) branches. We additionally evaluate the effects of different recurrent cell types on the global aggregated feature importance branch ($\mathcal{G}$) of MTBlocks. Finally, we demonstrate the resulting accuracies by employing different spatio-temporal pooling methods applied to inputs of $\mathcal{P}$.

Table 5.6. Branch channel ratio: Varying channel ratio (δ) across MTNet$_M$ and MTNet$_L$ architectures.

Net.	δ setting	top-1	top-5	FLOPs (G)	Params (M)
MTNet$_M$	$\delta = 1.0$ (No $\mathcal{P}$)	82.2	93.6	10.8	29.7
	7/8	**83.4**	**95.9**	8.8	25.8
	3/4	83.1	95.6	6.7	21.8
	5/8	81.6	93.2	4.8	19.3
	1/2	79.7	91.8	3.6	**18.6**
	3/8	78.6	89.4	2.6	19.2
	1/4	77.1	88.6	**2.1**	21.0
MTNet$_L$	$\delta = 1.0$ (No $\mathcal{P}$)	84.9	95.7	20.6	53.5
	7/8	**86.6**	**96.7**	17.6	50.1
	3/4	86.1	96.2	12.5	45.3
	1/2	83.2	95.3	7.09	**42.7**
	3/8	82.1	93.9	5.2	45.3
	1/4	80.3	92.4	**4.1**	47.8

Branch channel ratio. As one primary feature of MTConvs is the bifurcation of channels assigned to local ($\mathcal{L}$) and prolonged ($\mathcal{P}$) branches, we further explore how different ratio values (δ) affect performance accuracy-wise, and in terms of computations and parameters. Evident from Table 5.6, the best ratios (δ) in terms of accuracies, are in range of $0.875 \sim 0.75$ with changes in performance being marginal ($\pm 0.3 \sim 0.5\%$) for both the top-1 and top-5 accuracies. Compared to the use of standard 3D Convs, these ratios lead to further reductions in both multiply-add operations as well as the number of parameters. The improvements in the number of GFLOPs given these ratios are further demonstrated by a significant reduction of $25 \sim 37$ % with the use of both $\mathcal{L}$ and $\mathcal{P}$ branches in comparison to using solely the local branch ($\mathcal{L}$). The sole use $\mathcal{L}$ branch is equivalent to a single standard 3D Conv. Computation-wise, the best ratio is $\delta = 1/2$ as it demonstrates the largest balanced combined reduction in terms of GFLOPs (-66%) and number of

parameters (-37%). The attributing factor for the loss in performance when using small ratios is the strong dependency of branch $\mathcal{P}$ to branch $\mathcal{L}$. Interestingly, the decrease in local feature dimensionality corresponds to the inability of the prolonged features to encapsulate substantial video action details by themselves. We note that δ decreases do not relate directly to reductions in parameters as seen in Table 5.6, since further reductions with $\delta < 1/2$ show an increasing trend, with branch $\mathcal{P}$ employing a larger number of parameters. Based on this, the smallest number of parameters is observed when the ratio is split equally (balanced) between the two channels ($\delta = 1/2$). Lastly, we note that zero-valued ratios $\delta = 0$ are not architecturally feasible as branch $\mathcal{P}$ includes the outputs of branch $\mathcal{L}$.

Table 5.7. Recurrent cell configurations: Alternative recurrent cells for the *global aggregated feature importance branch* ($\mathcal{G}$). Branch ratio of $\delta = 7/8$.

Net	Cell type	Params (M)	FLOPS (G)	Latency (msec) ↓F	Latency (msec) ↑B	top-1	top-5
MTNet_S	RNN [211]	24.3	5.8	58	78	78.8	93.7
	LSTM [96]	26.5	5.8	61	79	79.9	94.3
	LSTM (peephole) [76]	26.5	5.8	68	85	80.1	94.5
	GRU [40]	25.8	5.8	65	80	**80.7**	**95.2**
MTNet_M	RNN [211]	24.3	8.8	84	113	82.5	94.8
	LSTM [96]	26.5	8.8	86	109	83.1	95.4
	LSTM (peephole) [76]	26.5	8.8	94	120	83.2	95.6
	GRU [40]	25.8	8.8	90	111	**83.4**	**95.9**

Recurrent cell configuration. Evaluations are done in terms of the accuracy effects based on changes of the recurrent methodology. Table 5.7 demonstrates recurrent layer method changes for $\textbf{MTNet}_S$ and $\textbf{MTNet}_M$. These changes only affect the global aggregated feature branch ($\mathcal{G}$). Latency times are reported based on the time (in msecs) required for a full forward (↓F) and backward (↑B) pass of a single $16\times256\times256$ clip to the entire network. We distinguish this from the reported latencies in Table 5.5 which are the per clip averages over entire batches of 32 clips. The use of GRUs [40] has been motivated by the (small) improvements compared to alternative recurrent cell structures of LSTMs and RNNs. Considering $\textbf{MTNet}_S$, GRUs seem to perform slightly better than regular RNN cells [211] with +1.9% top-1 and +1.5% top-5 accuracies. However, the overall simplicity of RNNs proves to be more efficient in terms of parameters, with a -8% overall network parameter reduction as well as marginally faster forward and backward latency times. This observation is also present for $\textbf{MTNet}_M$ as GRU's top-1 and top-5 accuracies improve the RNN baseline by +0.9% and +1.1% respectively. In comparison to LSTMs [96] and LSTMs with peepholes variants [76], GRUs also show marginally improved accuracy rates. However, GRUs merge LSTM's *forget* and *input* states as well as their *cell* and *hidden* states, thus simplifying the overall recurrent cell structure by requiring a smaller number of operations and weights. This is demonstrated by the reduction in number of parameters of GRUs in comparison to both LSTM variants.

Spatio-temporal pooling methodology. We include in our ablation studies accuracy changes attributed to different pooling methods for $\mathcal{P}$ branch's inputs. The methods tested can be both temporally and spatially symmetric (with the downsampling operations being performed similarly across all three dimensions) and asymmetric (with spatial pooling done separately from temporal). Our findings are reported in Table 5.8, in which we demonstrate the top-1 accuracies for different pooling configurations. Our

Table 5.8. Spatio-temporal pooling methodology: Top-1 accuracy for different pooling methods used on inputs for the *prolonged branch* ($\mathcal{P}$). Ratio is set to $\delta = 7/8$.

Net	Pooling					
	Avg	Max	Stochastic [310]	SoftPool [246]	Avg + *cos*	SoftPool [246] + *cos*
MTNet_S	77.8	75.9	76.8	77.8	80.5	**80.7**
MTNet_M	79.8	77.6	78.2	80.7	82.6	**83.4**
MTNet_L	83.8	82.1	82.9	84.2	85.9	**86.6**

Table 5.9. Transfer Learning on UCF-101: Top-1 and top-5 accuracies after pre-training.

Model	Pre-training	top-1	top-5
I3D	K-400	92.4	97.6
TSM	K-400	92.3	97.9
ir-CSN-152	IG65M	95.4	99.2
MF-Net	K-400	93.8	98.4
SF r3d-50	ImageNet	94.6	98.7
SF r3d-101	ImageNet	95.8	99.1
SRTG-101 (3D)	HACS+K-700	97.3	99.6
SRTG-101 (2+1)D	HACS+K-700	97.2	99.1
MTNet_S **(ours)**	HACS	94.2	98.0
MTNet_M **(ours)**	HACS	95.4	98.1
MTNet_L **(ours)**	HACS	**97.4**	**99.2**

frame selection method with the proposed temporal triplet cosine (*cos*) similarity produces improvements over all of the tested symmetric methods. Average pooling with triplet *cos*, shows accuracy rate improvements of +2.7% for $MTNet_S$, +2.8% on $MTNet_M$ and +2.1% for $MTNet_L$. Following the same trend, SoftPool [246] in combination with a triplet *cos* frame selection increases top-1 accuracy by +2.9%, +2.7% and +2.4% for each of the aforementioned models, respectively, in comparison to symmetric baseline SoftPool. Overall, the gap in accuracy rates between both asymmetric methods utilising average pooling or SoftPool is only considered marginal with ±0.57%. The improvement is observed with the use of SoftPool instead of average pooling. With this considered, changes in performance are primarily associated with the use of triplet *cos* in comparison to temporally-extended spatial methods. Therefore, we base our choice of temporal pooling on the significant gains in performance through the use of triplet *cos* similarity pooling in comparison to spatio-temporally symmetric pooling methods, that apply pooling operations similarly across all dimensions.

5.4.7 Feature Transferability evaluation

We compare transfer learning capabilities of MTNets with state-of-the-art video models on UCF-101 for fine-tuning. In order to compare with other methods, all tested models use the same weight initialisation as in Table 5.5. We note that $MTNet_L$ pre-trained on

HACS, achieves similar performance to that of **SRTG r(2+1)d-101** and **SRTG r3d-101** which have been pre-trained on both HACS and K-700 datasets. Both of the models perform better than the rest of the tested architectures within our settings. $MTNet_M$ demonstrates accuracy rates close to those of ir-CSN which uses the IG65M dataset sourced from Instagram [77] as well as the 101 variant of SlowFast. The smallest of the MTNet architectures, $MTNet_S$, shows accuracies above those of TSM and I3D while the overall performance is comparable to that of SlowFast-50. With this, we further demonstrate the generalisation capabilities of our varying spatio-temporal feature extraction approach as well as the dynamic temporal feature calibration module.

5.5 Discussion and conclusions

We have presented in this section the challenges that are faced when using local patterns as descriptors of actions of variant complexity and duration. Action identities of local video segment may not directly relate to the underlined action. Our aim is the extension of the temporal receptive fields of convolutions by encoding features across different temporal modalities.

Our proposed Multi-Temporal Convolutions (MTConvs) are based on the extraction of features across variant spatio-temporal windows. Our multi-temporal blocks are built on three branches. The local branch extracts motion characteristics within a short temporal location, the prolonged branch is used for motion features of extended durations that include relation to the local branch. The global aggregated feature importance branch calibrates branch information based on the feature dynamics.

We have evaluated our work on four large scale datasets Kinetics-400, Kinetics-700, Moments in Time and HACS, demonstrating competitive results to state-of-the-art architectures and in most cases outperforming them. This also comes with a significant reduction in the computational complexity signifying the overall efficiency of our method. Our ablation studies further validate our claims and motivate our design choices. Based on the obtained results, we believe that modelling variable-duration spatio-temporal patterns is a viable research direction to inspire future works in the field of video action recognition.

Chapter 6

Class-Specific Regularisation Across Time

In this chapter we consider the inclusion of class-related information during the feature extraction process. The proposed method fuses class-based information to the extracted features by amplifying activations that better correspond to the specific class. We evaluate how the proposed feature regularisation method, through activation amplification, can improve classification performance based on classes in which their overall features demonstrate relative similarities or overlap.

6.1 Introduction

In the previous chapters, we explored the modelling difficulties of motion variations and proposed an approach to directly address these through the inclusion of temporally local patterns and prolonged temporal features. The resulting multi-temporal convolutions were based on the use of a triple-branch approach and the subsequent fusion of their produced features. We additionally proposed an attention-based, feature alignment method to integrate the relationships between short temporal motion characteristics with respect to their importance in the context of the entire video sequence. Through this, temporally important features are highlighted by recurrent cells, enabling the creation of coherent activations based on the discovered averaged feature attention. Even though the proposed feature extraction process has shown significant gains over fixed-sized spatio-temporal convolution alternatives, we believe that there is still room for improvement. Specifically, we consider the uniform nature of the feature extraction process even for classes that exhibit a high degree of similarity in terms of their temporal features and movements. Therefore, in this chapter we examine the regularisation of feature activations through features of the corresponding class that they represent.

The main architectural principle of CNNs is that they include a large stack of multiple subsequent layers within a single hierarchical architecture. Based on this, the feature extraction process takes the form of applying successive convolutions alongside non-linearities over an input. This adds an additional layer of complexity through every successive operation. Kernels at early layers of the architecture are capable of extracting simple textures and patterns, while deeper layers target features with higher degrees of complexity. Through this, the feature dependency to the preceding layer's weighted neural connections becomes more apparent in later layers. Only a portion of these features and their inherited connections is specific to an individual class [11, 78]. As all of the kernels in CNNs are learned in a class-agnostic way, the main features that are descriptive of each class, and discriminative between instances, are predominantly computed at the very last layer. This not only limits the capabilities of the network, by optimising towards features

corresponding to multiple classes, but also their interpretability. This is because there is no direct association between features and a specific class. This link additionally cannot be easily discovered.

Our aim in this chapter is to study the effects of using class-specific feature activations to propagate information through the network. The proposed method termed *Class Regularisation* (CR) [247] utilises class information within the feature extraction process in convolution blocks. This class-relative information is added back to the network through the amplification and suppression of activations, with respect to the predicted classes in the batch. An additional benefit of regularising activations based on corresponding class features, is that the effects of activations are further modulated, which is a significant benefit on the application of non-linearities. Through this class-relative information fusion, the CNN can differentiate between features in terms of their relevance to a specific class. This leads to a reduction of uncorrelated features that include noise in the final fully-connected layers that are responsible for the class predictions.

The chapter is structured as follows. We first overview approaches that are based on normalisation and regularisation of convolutional features in Section 6.2. We then provide a formal description of our method in Section 6.3. Our experiments on the task of action recognition are shown in Section 6.4. We conclude and discuss possible future research pathways in Section 6.5.

6.2 Normalisation and regularisation of features

The inclusion of regularised information in CNNs is challenging given the increased level of ambiguity in terms of the model's inner workings with respect to the network depth. The two main methods used to change the distributions of activation maps are based on either activation normalisation or regularisation.

Normalisation. One of the first normalisation-based methods has been *Batch Normalisation* (BN) [105]. BN aims at addressing activation value distribution changes during updates, referred to as *Internal Convergence Shift*, by using the mean and standard deviation for each of the input activations. However, normalisation is not specifically bound to feature activations. *Weight Normalisation* (WN) [216] introduces a parameterisation of a layer's weight vector which utilises the unit's length, effectively decoupling the parameters from their directions within the weight space. Santurkar *et al.* [217] have argued that instead of reducing *Internal Convergence Shift*, BN instead produces a smoother loss landscape. This new landscape improves the overall stability of models by allowing larger learning rates and faster overall convergence. Van Laarhoven [270] has further demonstrated how both BN and WN significantly reduce the effects of *weight decay* (L_2 regularisation). This is due to the fact that BN and WN create scaling-invariant weights. This has further been explored by Ba *et al.* [4], who demonstrated that BN and WN back-propagated gradients are scale-invariant. Other normalisation methods also include the use of *Local Response Normalisation* (LRN) [107, 131, 159, 222, 309] which computes the statistics within neighbourhoods of pixels rather than creating a uniform normalisation criterion for the image in its entirety, as BN does. Works that have refrained from utilising batch-based information include *Layer Normalisation* (LN) [4], which performs mean and variance computations channel-wise for layer input activations and per example. This allows for the process to be executed similarly during both training and testing. Supplementary to this, *Instance Normalisation* (IN) [267] performs LN individually for each channel which reduces the cross-channel dependencies. Lastly, *Group Normalisation* (GN) [294] is a combination

of both LN and IN, and performs normalisation over a group of input channels. If the group number is equal to one then GN takes the form of LN, while if the number of groups is equal to the number of input channels, it is similar to IN.

Regularisation. A smaller number of works have also considered BN as a method that can regularise information [81, 254, 312]. However, the two most popular regularisation techniques in the literature have been the inclusion of random noise in the data augmentation process [132, 208] and the use of Dropout [238] with the utilisation of a generalised linear model [275].

As described, works in terms of regularisation of activation data have been scarce. In addition, current normalisation and regularisation techniques take the form of general solutions in CNNs, and a specific method for spatio-temporal data does not exist. In addition, to our knowledge no regularisation-based technique has been proposed that can bridge the general nature of convolutional feature extracted and the corresponding descriptive features for classes specifically. Our proposed method named Class Regularisation fills this void as a general solution that can be used in any architecture with minimum computational cost and regularises the feature distribution by including each feature's proportional importance to the target class.

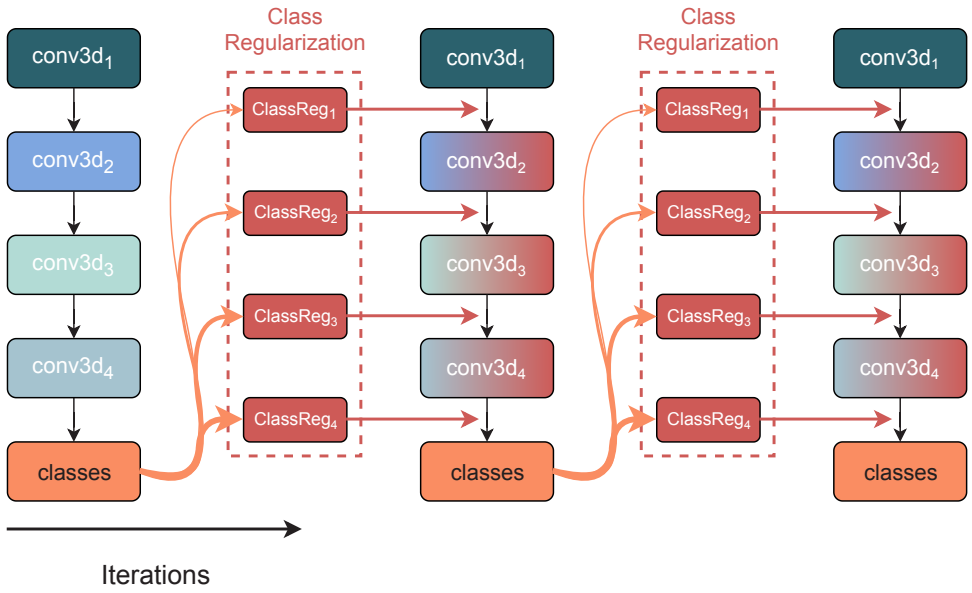


Figure 6.1. Iterative Class Regularisation. Pathway creation between class weights and intermediate layer features. An overview of the in-block operations is presented in Figure 6.2.

6.3 Regularisation over convolution blocks

In this section we first formulate Class Regularisation in Section 6.3.1 and explain how class features are used alongside extracted convolution features. We then provide a description of the parameter update process in Section 6.3.2.

6.3.1 Layer and class feature fusion

Class Regularisation is built on the main notion that different kernels focus on spatio-temporal patterns that appear in different classes. We use a standard notation based

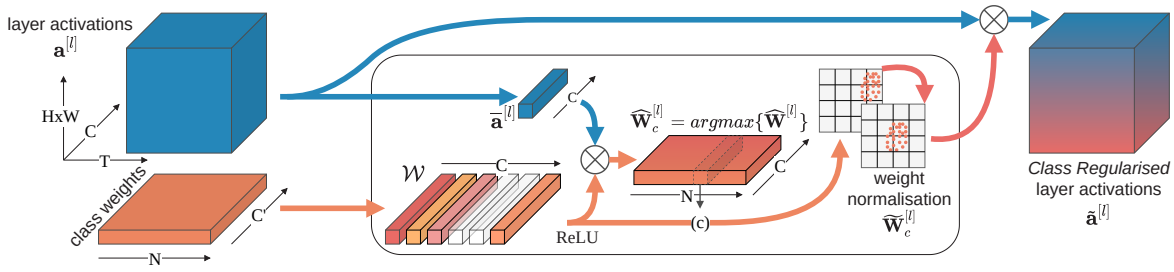


Figure 6.2. Class Regularisation operations. Class to feature correlations are encoded in weight vector $\widetilde{\mathbf{W}}^{[l]}$ with the class (c) vector with the maximum correspondence being selected ($\widehat{\mathbf{W}}_c^{[l]}$) and scaled based on the layer used to $\widetilde{\mathbf{W}}_c^{[l]}$. The weights are then applied to the input activation maps creating Class Regularised activations ($\tilde{\mathbf{a}}^{[l]}$). ($\otimes$) denotes channel-wise multiplications.

on which activation maps of layer l are denoted by $\mathbf{a}^{[l]}$ and have a size of C channels, T temporal extent and H and W spatial dimensionality. We further denote the number of classes as N with the weight vector ($\mathbf{W}$) in the final fully-connected layer for predictions of size $N \times C'$, where C' is the number of channels.

A visualisation of our approach can be found in Figure 6.2. When considering the inclusion of class-related features to a layer $[l]$, one of the main problems is the representation of weights $\mathbf{W}$. Class weights discover the feature relevance to classes, in feature space C'. However, layer $[l]$ features are based on feature space C. This task comes with information loss due to the *curse of dimensionality* when traversing from one feature space to another. In order to create a single differentiable mapping of the class weights over feature space C', we use a single point-wise convolution function that acts as a transformation [115], remapping activations in space C to corresponding values in C'. The weight vector used is denoted as $\mathcal{W}$ and is learned over training with the parameters being added to the optimiser. We use a *ReLU* non-linear function to discover the final weight representations and to perform updates based on the calculated gradients during back-propagation. The new weight is formulated as $\mathbf{W}^{C' \rightarrow C}$.

Although the size of the created embedding space of $\mathbf{W}^{C' \rightarrow C}$ is equal to that of $\mathbf{a}^{[l]}$ ($|W^{C' \rightarrow C}| = |a^{[l]}|$), their embeddings do not necessarily conform to the same representations. Therefore, in order to make the new weights $\mathbf{W}^{C' \rightarrow C}$ compatible to the representation space of activation maps, we first create a sub-sampled volume of $a^{[l]}$ through pooling over its spatio-temporal extent. The new volume $\overline{\mathbf{a}}^{[l]}$ encapsulates a spatio-temporal average feature activation of the original volume and provides a concatenated view of the activations. Based on the averaged feature map, we perform a point-wise multiplication of $\overline{\mathbf{a}}^{[l]}$ over each class ($n \in N$) location in weight vector $\mathbf{W}^{C' \rightarrow C}$. The resulting class weights $\widehat{\mathbf{W}}^{[l]}$ contain weights based on classes with respect to the current layer activations. This is similar to the application of class weights in the final layer with the exception of the transformation between feature spaces $C' \rightarrow C$.

In order to associate a class with the related features of the layer, we select the vectors with the highest feature activations. Therefore, the selection of the corresponding class that is best described by the averaged global features of vector $\overline{\mathbf{a}}^{[l]}$ takes the form of discovering the maximum activation instance within vector $\widehat{\mathbf{W}}^{[l]}$. This is again similar to the top-1 selection in the final class prediction layer in the network. The selection is done to discover the index (c) of the maximum activation instance $\widehat{\mathbf{W}}_c^{[l]}$. We refrain from using the volume

Algorithm 1: Class Regularisation computational overview

Data: Layer $[l]$ activation map ($\mathbf{a}^{[l]}$).
Class weights ($\mathbf{W}$) for N classes.
Result: Class-regularised activations ($\widetilde{\mathbf{a}}^{[l]}$).

$\overline{\mathbf{a}}^{[l]} \leftarrow pool(\mathbf{a}^{[l]})$
for n *in* N **do**
  $\mathbf{W}_n^{C' \rightarrow C} \leftarrow max\{0, \mathcal{W} * \mathbf{W}_n\}$
  $\widehat{\mathbf{W}}_n^{[l]} \leftarrow \mathbf{W}_n^{C' \rightarrow C} \otimes \overline{\mathbf{a}}^{[l]}$
end
$c \leftarrow 0$
for n *in* $/N/$ **do**
  if $\widehat{\boldsymbol{W}}_n^{[l]} > \widehat{\boldsymbol{W}}_c^{[l]}$ **then**
    $c \leftarrow n$
  end
end
$\widetilde{\mathbf{W}}_c \leftarrow \lambda \frac{(\mathbf{W}_c^{[C' \rightarrow C]} - min\{\mathbf{W}_c^{[C' \rightarrow C]}\}) * (1-\lambda)}{max\{\mathbf{W}_c^{[C' \rightarrow C]}\} - min\{\mathbf{W}_c^{[C' \rightarrow C]}\}}$
$\widetilde{\mathbf{a}}^{[l]} \leftarrow \mathbf{a}^{[l]} \otimes \widetilde{\mathbf{W}}_c$

$\widehat{\mathbf{W}}_c^{[l]}$ directly, as it represents the maximum activations based on classes rather than the feature weights that correspond to the respective classes. Instead, the index c is used to select the class weights from weight vector $\mathbf{W}^{C' \rightarrow C}$.

Given that each layer l is at a different location inside the network, the descriptive capabilities of their features, in terms of the classes that they best correspond to, are significantly impacted. As the feature complexity increases in the later network layers, this connection of features to classes becomes more apparent. Therefore, as the feature quality with respect to the target classes can vary across the architecture, we introduce an *affection rate* (λ) term. The main aim of the term is to scale weights, and correspondingly their effect over the activation maps, based on the depth of the regularised layer. Through this, weights are then normalised given their maximum and minimum values and they are shifted accordingly.

The final normalised class weights ($\widetilde{\mathbf{W}}$) are applied over the spatio-temporal layer activations ($\mathbf{a}^{[l]}$), through a point-wise multiplication similar to the use of weights. Based on the selected class (c) specific features in the activation volume are amplified or reduced. A full overview of the execution sequence of the processes performed by Class Regularisation are shown in Algorithm 1. We use (*) to simplify convolution denomination from Boyce and DiPrima [21]. We further denote ReLU activations as the maximum of a given operation ($f(x)$) over a given volume (x) as the maximum between the produced function value and zero ($max\{0, f(x)\}$).

A possible question that may arise during the formulation is based on the calculation of a layer-feature inclusive weight vector $\widehat{\mathbf{W}}^{[l]}$ and the re-calculation of class-based activations with $\widetilde{\mathbf{a}}^{[l]} \leftarrow \mathbf{a}^{[l]} \otimes \widetilde{\mathbf{W}}_c$. We make this distinction in our method as $\widehat{\mathbf{W}}^{[l]}$ utilises a spatio-temporally pooled vector of feature activations. This disregards the feature localities that are represented in $\mathbf{a}^{[l]}$ and thus does not allow for the corresponding class and feature inclusive volume to be directly used. Instead, the normalised weights $\widetilde{\mathbf{W}}_c$ are applied over the original input and thus can discover class-relative features over regions in the activation

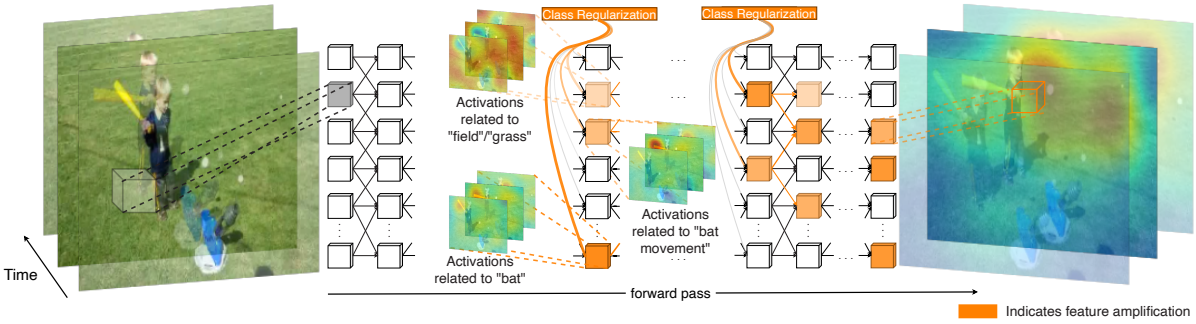


Figure 6.3. Visualisation of feature amplification. Class informative spatio-temporal features are intensified during iterations. The effect of amplification is propagated to later layers through the layer connections in which Class Regularisation is applied.

map. It is also possible to apply the weights $\mathbf{W}^{C' \to C}$ directly to $\mathbf{a}^{[l]}$ and decrease the overall complexity. However, this will result in a significant increase in both computations and memory requirements as it will require the discovery of the maximum class feature activations from the created spatio-temporal activations.

6.3.2 Updating class regularisation weights

As our proposed method utilises learnable parameters, there is a requirement of integrating the used weights ($\mathcal{W}$) as part of the learning procedure. We distinguish between the weights ($\mathcal{W}$) in our method that are used for dimensionality matching and the prediction layer's weights ($\mathbf{W}$) they are applied to. The weights in the prediction layer are updated as normal based on the gradients calculated given the probabilistic class distribution. In order to re-use their values for Class Regularisation, we decouple them from the original computation graph and include them as a supplementary input as shown in Figure 6.2. As the class prediction weights are used from the previous iteration, this decoupling of the weights has no effect on the class prediction weights in the current iteration.

The Class Regularisation parameters are updated based on the probabilistic loss calculated from the final prediction layer and thus are not required for the creation of an individual criterion. Dissimilarities in the feature representation space between layer activations $a^{[l]}$ and the transformed class weights $\mathbf{W}^{C' \to C}$ are discovered based on the error corresponding to the class predictions. The alignment of the two feature spaces can also be achieved through a distance minimisation cost function (similar to an autoencoder style loss [130, 210, 255]). However, we refrain from using a distance minimisation criterion for each Class Regularisation, or a concatenated version of it, at all locations that Class Regularisation is applied to, in order to limit the backwards traversal of information and the subsequent number of operations and FLOPs that are required. In addition, specific loss functions would mean that multiple gradients would be calculated for the remainder of the network parameters. Bifurcating the gradients and performing more than one step for each parameter during each iteration step has the potential of leading to sub-optimal parameter space directions and locations. Instead, by integrating the Class Regularisation parameters in the prediction task, the task of matching between the two created spaces is accumulated to the main classification task.

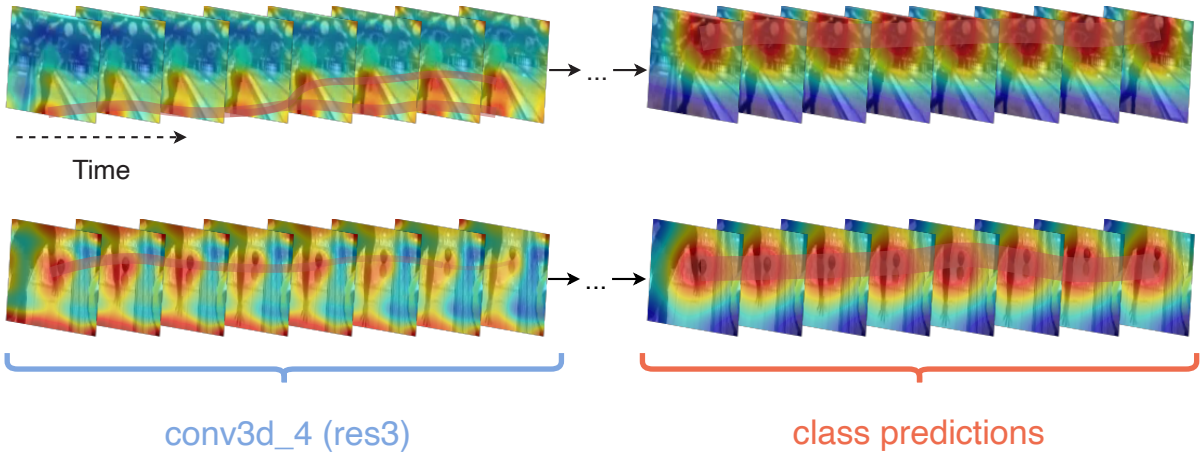


Figure 6.4. Layer-wise class to feature correlation. We use the features for (*res3*) in wide-ResNet-50 [308] with 3D convolutions. Examples are drawn randomly from Kinetics-400 [120] and are both from class *"bowling"*.

6.3.3 Class Regularisation for visual explanations

Class Regularisation focuses on the inclusion of class-relative information inside the extracted features. This results in the creation of a direct correspondence between classes and features for each convolutional block that the method is applied to. An additional attribute, of this representation of class features in a feature space relative to the layer activations, is the improvement in terms of the overall explainability capabilities of the model. The curse of dimensionality problem has led to the dependence of back-propagating from the predictions to a particular layer. By actively alleviating it for the visualisation of relevant features, direct feature to class visual explanations can be created. Since the classes are represented in the same feature space as the activation maps, spatio-temporal regions that are informative over multiple network layers, can be discovered. To the best of our knowledge, this is the first method that enables such a direct feature correspondence.

We extend the proposed regularisation method to enable visualisation with the inclusion of *Saliency Tubes* [241] at each block that it is applied to. Based on this, we create representations of features with the highest activations with respect to the selected class. We show two visual examples of the class *"bowling"* in Kinetics-400 to demonstrate class activations in different layers of the network (Figure 6.4). As observed, in both cases features in the early layers lack in terms of their deterministic capabilities for their feature to class correspondence. Most of the features are targeted at the distinction between foreground and background. In later layers however, this focus shifts from the actor to the background in the predictions layer in the top clip. As the bowling ball and bowling pins are part of a wallpaper, the overall focus shifts which demonstrates that a strong spatial-based visual signal is favoured over certain cases during the feature extraction process. In the bottom clip, the main field of focus of the network in the last layers is more limited to the area following the ball's trail. We note that the experimented architectures fuse spatio-temporal features together in the last convolution layers of the network, thus only the spatial extent can be visualised for the final class predictions.

The amplification of layer features based on corresponding classes is additionally presented in Figure 6.3. In Figure 6.3, the top-3 kernels to be amplified for an example of class *"baseball hit"* can have a correspondence to the spatio-temporal features, such as the appearance of the bat, the field and the movement of the bat during a swing. In

addition, these amplifications are also propagated to deeper layers in the network through the connections of the most informative kernels.

6.4 Experiments

We present the merits of our proposed Class Regularisation for the task of action recognition on ResNet-based architectures [90, 89, 261, 308] as well as on MTNets [244]. The ResNet architectures chosen include both 3D and (2+1)D convolution variants as well as differences in their channel dimension size with wide-ResNets [308]. We include wide Residual Networks in our tests in order to have a greater understanding of the effects of larger feature dimension sizes. We do not explicitly include experiments with SR [243] networks from Chapter 4, as MTNets include SR as part of their architectures.

6.4.1 Architectural overview based on Class Regularisation

An overview of the architectures alongside how Class Regularisation is applied over the networks can be viewed in Table 6.1. We first present the ResNet architectures in Table 6.1a alongside the progressive change in λ. Our choice of values is based on the number of layers with $\lambda = 1$ denoting that Class Regularisation will have no effect over the volume that it is applied to based on the regularisation shift. Instead, when $\lambda \to 0$ the effect of Class Regularisation increases over the volume that it is applied to. Based on this, we progressively decrease λ as the feature complexity increases. In Table 6.1b we show how the MTNets that use a X3D backbone are structured. For simplicity, we do not elaborate on the processes performed in a MTConv. Therefore, for example, a $3\times3\times3$ MTConv with 54 channels will correspond to a $3\times3\times3$ 3D Conv for the local ($\mathcal{L}$) branch with 47 channels alongside a $3\times3\times3$ 3D Conv for the prolonged ($\mathcal{P}$) branch with 7 channels and a single point-wise convolution for transforming the local branch feature space to that of the prolonged $47 \to 7$. The primary change between MTNet$_S$ and MTNet$_M$ is in their input size similarly to the X3D backbones that they employ [63]. MTNet$_S$ uses inputs of size 13×160^2, while MTNet$_M$ uses inputs of size 16×224^2. The affection rate λ follows a scaling similar to the ResNets.

6.4.2 Experiment setup

We demonstrate our proposed method on two benchmark action recognition datasets. For our experiments we use the widely popular Kinetics-400 and HACS for baseline comparisons.

Training is only performed for the weights associated with Class Regularisation. The rest of the network weights are from pre-trained models and no updates are performed over them. On Kinetics-400 (K-400) and HACS we train the Class Regularisation weights from scratch while using a standard Kaiming initialisation [91]. The initial mini-batch is set to 16 with 4 clips per GPU. For all of our experiments we use a SGD optimiser with momentum [252] which we set to 0.9. We use weight-decay of value 1e-5. We use spatio-temporal data augmentations and image crops similar to the previous chapter. All models were trained over a total of 120 epochs as no further improvements were observed afterwards. Based on this, we further apply a step-wise learning rate reduction every 50 epochs to one tenth of the previous learning rate value. The initial learning rate is set to 0.1. For fine-tuning we decrease the learning rate to 1e-3 and set the step-wise learning rate reduction to every 70 steps.

Table 6.1. Architectures of 3D convolutions with and without Class Regularisation. The convolution kernels are denoted first followed by the number of channels and the number of blocks. Throughout all the models, Class Regularisation accounts for a small number of parameters.

(a) ResNet-based models with and without Class Regularisation

layer name	3D ResNet50 (w/o CR)	(2+1)D ResNet50 (w/o CR)	3D Wide ResNet 50 (w/o CR)	3D ResNet101 (w/o CR)
$con3d_1$	$7 \times 7 \times 7, 64$ conv			
$con3d_2$	$\begin{bmatrix} 1\times1\times 1 \\ 3\times3\times3 \\ 1\times1\times1 \end{bmatrix} (\times 64) \times 3$	$\begin{bmatrix} 1\times1\times1 \\ 1\times3\times3 \\ 3\times1\times1 \\ 1\times1\times1 \end{bmatrix} (\times 64) \times 3$	$\begin{bmatrix} 1\times1\times1 \\ 3\times3\times3 \\ 1\times1\times1 \end{bmatrix} (\times 128) \times 3$	$\begin{bmatrix} 1\times1\times1 \\ 3\times3\times3 \\ 1\times1\times1 \end{bmatrix} (\times 64) \times 3$
$ClassReg_1$	- / $\lambda = 0.9$	- / $\lambda = 0.9$	- / $\lambda = 0.9$	- / $\lambda = 0.9$
$con3d_3$	$\begin{bmatrix} 1\times1\times1 \\ 3\times3\times3 \\ 1\times1\times1 \end{bmatrix} (\times 128) \times 4$	$\begin{bmatrix} 1\times1\times1 \\ 1\times3\times3 \\ 3\times1\times1 \\ 1\times1\times1 \end{bmatrix} (\times 128) \times 4$	$\begin{bmatrix} 1\times1\times1 \\ 3\times3\times3 \\ 1\times1\times1 \end{bmatrix} (\times 256) \times 4$	$\begin{bmatrix} 1\times1\times1 \\ 3\times3\times3 \\ 1\times1\times1 \end{bmatrix} (\times 128) \times 4$
$ClassReg_2$	- / $\lambda = 0.8$	- / $\lambda = 0.8$	- / $\lambda = 0.8$	- / $\lambda = 0.8$
$con3d_4$	$\begin{bmatrix} 1\times1\times1 \\ 3\times3\times3 \\ 1\times1\times1 \end{bmatrix} (\times 256) \times 6$	$\begin{bmatrix} 1\times1\times1 \\ 1\times3\times3 \\ 3\times1\times1 \\ 1\times1\times1 \end{bmatrix} (\times 256) \times 6$	$\begin{bmatrix} 1\times1\times1 \\ 3\times3\times3 \\ 1\times1\times1 \end{bmatrix} (\times 512) \times 6$	$\begin{bmatrix} 1\times1\times1 \\ 3\times3\times3 \\ 1\times1\times1 \end{bmatrix} (\times 256) \times 23$
$ClassReg_3$	- / $\lambda = 0.7$	- / $\lambda = 0.7$	- / $\lambda = 0.7$	- / $\lambda = 0.7$
$con3d_5$	$\begin{bmatrix} 1\times1\times1 \\ 3\times3\times3 \\ 1\times1\times1 \end{bmatrix} (\times 512) \times 3$	$\begin{bmatrix} 1\times1\times1 \\ 1\times3\times3 \\ 3\times1\times1 \\ 1\times1\times1 \end{bmatrix} (\times 512) \times 3$	$\begin{bmatrix} 1\times1\times1 \\ 3\times3\times3 \\ 1\times1\times1 \end{bmatrix} (\times 1024) \times 3$	$\begin{bmatrix} 1\times1\times1 \\ 3\times3\times3 \\ 1\times1\times1 \end{bmatrix} (\times 512) \times 3$
$ClassReg_4$	- / $\lambda = 0.6$	- / $\lambda = 0.6$	- / $\lambda = 0.6$	- / $\lambda = 0.6$
predictions	global average pool, softmax unit group			

(b) MTNets with and without Class Regularisation.

layer name	MTNet_S (w/o CR)	(2+1)D MTNet_M (w/o CR)	MTNet_L (w/o CR)
$con3d_1$	$1 \times 3 \times 3, 24$ conv		
$con3d_2$	$\begin{bmatrix} 1\times1\times1 \\ 3\times3\times3 \\ 1\times1\times1 \end{bmatrix} \times \begin{bmatrix} 54 \\ 54 \\ 24 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1\times1 \\ 3\times3\times3 \\ 1\times1\times1 \end{bmatrix} \times \begin{bmatrix} 54 \\ 54 \\ 24 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1\times1 \\ 3\times3\times3 \\ 1\times1\times1 \end{bmatrix} \times \begin{bmatrix} 54 \\ 54 \\ 24 \end{bmatrix} \times 5$
$ClassReg_1$	- / $\lambda = 0.9$	- / $\lambda = 0.9$	- / $\lambda = 0.9$
$con3d_3$	$\begin{bmatrix} 1\times1\times1 \\ 3\times3\times3 \\ 1\times1\times1 \end{bmatrix} \times \begin{bmatrix} 108 \\ 108 \\ 48 \end{bmatrix} \times 5$	$\begin{bmatrix} 1\times1\times1 \\ 3\times3\times3 \\ 1\times1\times1 \end{bmatrix} \times \begin{bmatrix} 108 \\ 108 \\ 48 \end{bmatrix} \times 5$	$\begin{bmatrix} 1\times1\times1 \\ 3\times3\times3 \\ 1\times1\times1 \end{bmatrix} \times \begin{bmatrix} 108 \\ 108 \\ 48 \end{bmatrix} \times 10$
$ClassReg_2$	- / $\lambda = 0.8$	- / $\lambda = 0.8$	- / $\lambda = 0.8$
$con3d_4$	$\begin{bmatrix} 1\times1\times1 \\ 3\times3\times3 \\ 1\times1\times1 \end{bmatrix} \times \begin{bmatrix} 216 \\ 216 \\ 96 \end{bmatrix} \times 11$	$\begin{bmatrix} 1\times1\times1 \\ 3\times3\times3 \\ 1\times1\times1 \end{bmatrix} \times \begin{bmatrix} 216 \\ 216 \\ 96 \end{bmatrix} \times 11$	$\begin{bmatrix} 1\times1\times1 \\ 3\times3\times3 \\ 1\times1\times1 \end{bmatrix} \times \begin{bmatrix} 216 \\ 216 \\ 96 \end{bmatrix} \times 25$
$ClassReg_3$	- / $\lambda = 0.7$	- / $\lambda = 0.7$	- / $\lambda = 0.7$
$con3d_5$	$\begin{bmatrix} 1\times1\times1 \\ 3\times3\times3 \\ 1\times1\times1 \end{bmatrix} \times \begin{bmatrix} 432 \\ 432 \\ 192 \end{bmatrix} \times 7$	$\begin{bmatrix} 1\times1\times1 \\ 3\times3\times3 \\ 1\times1\times1 \end{bmatrix} \times \begin{bmatrix} 432 \\ 432 \\ 192 \end{bmatrix} \times 7$	$\begin{bmatrix} 1\times1\times1 \\ 3\times3\times3 \\ 1\times1\times1 \end{bmatrix} \times \begin{bmatrix} 432 \\ 432 \\ 192 \end{bmatrix} \times 15$
$ClassReg_4$	- / $\lambda = 0.6$	- / $\lambda = 0.6$	- / $\lambda = 0.6$
predictions	global average pool, softmax unit group		

6.4.3 Results on HACS

In our comparisons on the HACS [316] dataset we include Class Regularisation in the current top-performing MTNet [244] models and compare with other state of the art architectures in Table 6.2. In order to maintain comparison standards with the rest of the works, we use the same cropping and sampling size as in [63, 244], which results in 30-view spatio-temporal input clips. For each architecture we report, from left to right, the

Table 6.2. Action recognition model comparisons on HACS. Weight initialisation sources are denoted by their respective indicators.

Model	Pre	top-1	top-5	GFLOPs × views	Params
MF-Net [39]†	K-400 [120]	78.3	94.6	11.1×50	8.0M
TAM [62]†		82.2	95.2	86×12	25.6M
SF-101 [64]†		83.7	96.8	65.7×30	53.7M
X3D-L [63]†		85.8	96.1	24.8×30	**6.1**M
ir-CSN-101 [260]†	IG65 [77]	83.8	93.8	63.6×10	22.1M
ip-CSN-101 [260]†		84.1	93.9	63.6×10	24.5M
r3d-34 [119]*	-	74.8	92.8	26.6×30	63.7M
r3d-50 [119]*		78.4	93.8	52.6×30	36.7M
r3d-101 [119]*		80.5	95.2	78.0×30	69.1M
r(2+1)d-34 [119]*		75.7	93.8	37.8×30	61.8M
r(2+1)d-50 [119]*		81.3	94.5	83.3×30	34.8M
r(2+1)d-101 [119]*		82.9	95.7	163.0×30	67.2M
SRTG r3d-34 [243]	-	78.6	93.6	26.6×30	83.8M
SRTG r3d-50 [243]		80.3	95.5	52.7×30	56.9M
SRTG r3d-101 [243]		81.6	96.3	78.1×30	107.1M
SRTG r(2+1)d-34 [243]		80.4	94.3	37.8×30	82.1M
SRTG r(2+1)d-50 [243]		83.8	96.6	83.4×30	55.0M
SRTG r(2+1)d-101 [243]		84.3	96.8	163.1×30	105.3M
MTNet$_S$ [244]	-	80.7	95.2	5.8×30	25.8M
MTNet$_M$ [244]		83.4	95.9	8.8×30	25.8M
MTNet$_L$ [244]		86.6	96.7	17.6×30	50.1M
MTNet$_S$ (CR)	-	81.8	96.1	5.8×30	26.5M
MTNet$_M$ (CR)		84.7	96.7	8.9×30	26.5M
MTNet$_L$ (CR)		**87.5**	**97.4**	17.7×30	51.2M

† models and weights from official repositories.
* re-implemented models.

pre-training datasets (Pre) that the models used to initialise their weights following the average achieved top-1 and top-5 accuracy rates on the validation set. Models that do not include a pre-trained dataset are trained from scratch. For the last two metrics, we report the inference cost expressed as the required GFLOPs per single clip times the number of clips and their spatio augmentations (denoted as "views"). We finally report the overall number of parameters for each network.

In comparison to the top-preforming models on HACS, enabling Class Regularisation can increase the overall performance. Notably, it also comes at a negligible computational overhead when added over MTNet ($< +1\%$). More specifically, considering the overall efficiency of the MTNets in addition to the proposed methods, our MTNet$_S$ with CR can achieve accuracy rates similar to those of ResNet-101 with 3D or ResNet-50 with (2+1)D convolutions [261], as well as TAM [62] and SRTG r3d-50 [243]. However, the GFLOP requirements are very low, similar to the original MTNet$_S$ with only an additional > 0.1 GFLOPs that account for less than 10% of the total 5.8 GFLOPs. We additionally do

not notice a significant increase in terms of the computational overhead for MTNet$_M$ and MTNet$_L$ with the inclusion of Class Regularisation. For MTNet$_M$ (CR), we note a performance increase of +1.3% in the top-1 and +0.8% in the top-5 compared to the original network without CR. These accuracy rates are similar to both of the larger SRTG r3d-101 [243] and Channel-Separated Networks (ip-CSN-101) [260]. This increase in performance does not come at a cost in terms of the computational inference or the number of parameters with the additional number of parameters accounting for $\sim 10\%$ of the total number of network parameters. The larger MTNet$_L$ (CR) outperforms all other tested models by a margin of at least +0.9% in the top-1 and +0.7% (from MTNet$_L$) while maintaining almost identical computational complexity as MTNet$_L$. Considering the fact that the tested models were not fine-tuned, with their weights previously initialised from a significantly larger dataset, the performance benefits of Class Regularisation are evident.

6.4.4 Results on Kinetics-400

The results presented on the K-400 dataset include experiments on pre-trained state-of-the art architectures alongside pair-wise comparisons for networks with weights that have been randomly initialised similarly to the experiments on HACS. The distinction between the two is made in order to demonstrate the direct impact of Class Regularisation across different training environments. The first one explores the capabilities of the proposed method on pre-trained models while the other is done *from scratch*, similar to HACS.

Main accuracy results. We demonstrate in Table 6.3 the main accuracy rates achieved through the inclusion of Class Regularisation in the top-performing MTNet architectures. The accuracy rates follow a similar trend as those that have been presented on HACS, with MTNet$_L$ with Class Regularisation (CR) being the top-performing architecture. This further decreases the gap between the significantly larger X3D-XL and MTNet$_L$ to only 0.2% for the top-1 accuracy. However, the computational complexity of the MTNet$_L$ architecture still remains small, corresponding to approximately 36% of that of X3D-XL. Through this we show that the inclusion of CR comes with no additional computational costs, thus being a straightforward approach for lightweight architectures to further improve their classification accuracies. The smaller MTNet$_M$ also shows competitive results to those of the top performing models when also including CR. The top-1 accuracy is drawn closer to networks such as SlowFast-101 and MTNet$_L$ while only including a fraction of the computational requirements. In terms of efficiency, it still retains its overall efficiency with computations being reducing further from the lightweight MFNet and TSM. The only two more efficient models are the MTNet$_S$ variants with and without Class Regularisation. The smallest tested architecture, MTNet$_S$ (CR), shows a +1.3% improvement on the top-1 accuracy in comparison to the baseline model MTNet$_S$ without CR. This comes with less than 0.1 GFLOPs of additional computations. In terms of additional parameters, across all of the tested architectures the parameter increments with CR remain constant at approximately +0.76M parameters. This increase is more noticeable to small architectures, such as MTNet$_S$ in which the number of parameters is already limited. In larger architectures, such as MTNet$_L$ the additional parameters are proportionally less significant.

Pair-wise comparisons. In Table 6.4 we present pair-wise accuracies over models that are not pre-trained on larger datasets and instead are trained with their weights being randomly initialised. The accuracies of the models that do not include CR are re-calculated to ensure the same training setting. Over all of the four tested models, architectures that include CR outperform their counterparts without. The largest margins are observed for wide-r3d-50 with a 1.3% increase in the top-1 accuracy and r3d-101 with +1.3% increase

Table 6.3. Comparison with K-400 state-of-the-art. For consistency with previous testing methods, we report the model complexity as the GFLOPs per single clip view × the number of clips with spatial cropping of size 256×256.

Model	Input	Backbone	top-1	top-5	GFLOPs × views	Params
I3D [31]	16×224^2	InceptionV1	71.6	90.0	$108\times N/A$	12M
TSM [148]	16×224^2	ResNet50	74.7	91.4	65×10	24.3M
R(2+1)D [261]	16×224^2	ResNet101	62.8	83.9	152×115	63.6M
ip-CSN-101 [260]	8×224^2	ResNet101	76.7	92.3	83.0×30	24.5M
ip-CSN-152 [260]	8×224^2	ResNet152	77.8	92.8	108.8×30	32.8M
MF-Net [39]	16×224^2	ResNet50	72.8	90.4	11.1×50	8.0M
SF-50 [64]	$(32,4)\times224^2$	ResNet50	77.0	92.6	65.7×30	34.4M
SF-101 [64]	$(32,4)\times224^2$	ResNet101	77.9	93.5	213×30	53.7M
X3D-XL [63]	16×224^2	ResNet(X3D)	**79.1**	**93.9**	48.4×30	11.0M
TAM [62]	16×256^2	ResNet50	76.9	92.9	86×12	25.6M
SRTG r3d-101 [243]	16×224^2	ResNet101	73.2	91.3	78.1×30	107.1M
SRTG r(2+1)d-101 [243]	16×224^2	ResNet101	73.8	92.0	163.1×30	105.3M
MTNet_S [244]	13×182^2	ResNet(X3D)	74.8	92.1	$\mathbf{5.8\times30}$	25.8M
MTNet_M [244]	16×256^2	ResNet(X3D)	76.6	92.5	8.8×30	25.8M
MTNet_L [244]	16×256^2	ResNet(X3D)	78.1	93.2	17.6×30	50.1M
MTNet_S (CR)	13×182^2	ResNet(X3D)	76.1	92.4	5.8×30	26.5M
MTNet_M (CR)	16×256^2	ResNet(X3D)	77.7	92.9	8.9×30	26.5M
MTNet_L (CR)	16×256^2	ResNet(X3D)	78.9	93.6	17.7×30	51.0M

for the top-5 accuracy. In contrast, for smaller models such as r3d-50 the improvements show to be less with +0.6% for the top-1 and top-5 accuracies. This seems to suggest that larger models such as r3d-101 or models with an increased number of channels per layer, such as wide-r3d-50, benefit more from the proposed regularisation method. The added floating-point operations (FLOPs) remain constant as the method is not affected by the increase in the number of convolutional layers per block. However, an increase is observed for wide-r3d-50, in which the number of channels per block is expanded. This corresponds to the latent space of the feature representations per convolution being enlarged which in turn also affects the representation of class-relative features in that space as well. Similar tendencies are also observed for the number of parameters, with the only exception being wide-r3d-50 and with the increased convolutional feature space.

As shown in Figure 6.5, with the inclusion of Class Regularisation, overall accuracy improvements across the majority of the K-400 classes can be observed. The largest margins are shown for instances and classes in which the execution can be descriptive. Examples of such cases include the *"bench pressing"* , *"high jump"* and *"jogging"* classes with increases in their accuracies in the range of 8.9% to 15.6%. In contrast, classes that are more likely to contain significant feature variations are more likely to be wrongly classified. For example, considering the *"parkour"* class, no standard features can be established as there are significant variations across examples. This also includes classes such as *"garbage*

Table 6.4. Pair-wise comparisons for K-400. We compare with popular residual architectures trained from scratch with and without Class Regularisation on Kinetics.

Model	CR	Input	Accuracy (%) top-1	Accuracy (%) top-5	FLOPs (G)	Params (M)
r3d-50 [247]	✗	16×112^2	63.6	84.5	52.6	36.7
	✓		**64.2** (+0.6)	**85.1** (+0.6)	53.8 (+1.2)	37.6 (+0.9)
r(2+1)d-50 [247]	✗		64.5	85.2	83.3	34.8
	✓		**65.2** (+0.7)	**86.4** (+1.2)	84.5 (+1.2)	35.7 (+0.9)
wide-r3d-50 [247]	✗		64.0	85.4	168.6	140.9
	✓		**65.3** (+1.3)	**86.1** (+0.7)	171.3 (+2.7)	143.4 (+2.5)
r3d-101 [247]	✗		65.2	86.3	78.0	69.1
	✓		**67.7** (+2.5)	**87.6** (+1.3)	79.2 (+1.2)	70.0 (+0.9)

collection", which is performed either mechanically (top), by a single person (mid) or by multiple people (bottom). Factors such as the oscillations in *"pumping gas"* or contextual information in *"sniffing"* also effect the descriptive capabilities of features.

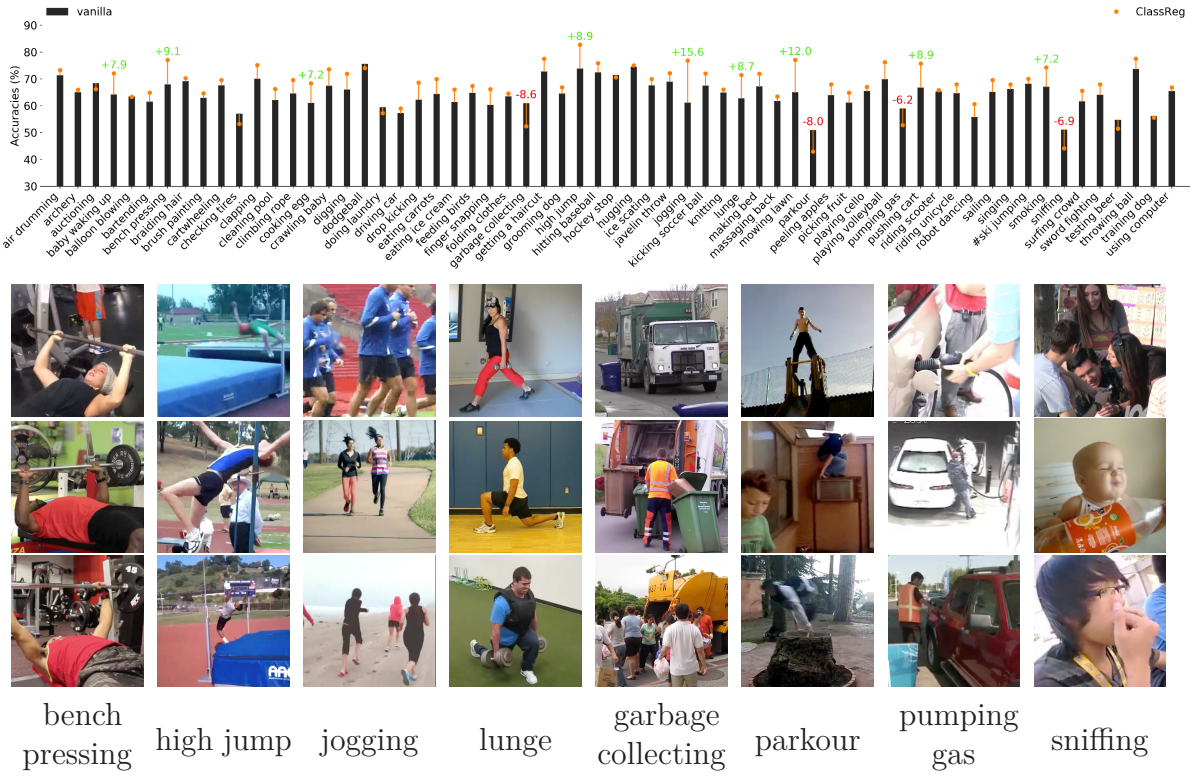


Figure 6.5. Per-class performance for ResNet-101 with and without Class Regularisation with illustrative examples of classes with large performance gains or losses in K-400.

6.4.5 Statistical significance

In the previous sections we have overviewed the accuracy rates achieved over large-scale datasets. Initial experiments included top-performing models with the inclusion of Class Regularisation in comparison to state-of-the-art architectures. In addition, we have

demonstrated pair-wise results, within the same training setting, for architectures with and without Class Regularisation. In both cases, the effect of Class Regularisation to the accuracy rates has been visible with a uniform increase in classification performance across both tasks and tested architectures. However, this only provides a partial understanding of the overall improvements that the proposed method can provide as it remains unclear how statistically different the predictions made were to those of the original models.

As the proposed method is an additional module that can be added to existing architectures, and does not perform any architectural changes at the network level, a reflection on the rates achieved is required. Therefore, there is an additional need for a testing measure in order to conclude if the resulting difference in performance does not indeed fall within the expected error margins. This can be expressed through a null hypothesis (h_0) such that the resulting accuracies are indeed similar. The statistical significance level (ρ) can be then defined as the probabilistic confidence that the null hypothesis (h_0) does hold true. Alternatively, based on a threshold value (a), probabilities smaller than the threshold value signify that the null hypothesis should be rejected as the homogeneity between results is small. This threshold value is commonly set to 0.05, as motivated by Fisher [67], and Caparo [47]. In our experiments we set the null hypothesis (h_0) as the achieved top-1 performance of a base model without Class Regularisation and the same model with Class Regularisation being statistically equivalent and within the margin of error.

As the hypothesis is based on pairs of model accuracies, we use a standard McNemar's significance test [163] to study the predictive accuracies of the models. The test highlights the proportions of examples that the two models disagree in terms of the predictions.

Table 6.5. Contingency table for the calculation of the χ^2 statistic.

		Model 1		
		Correct	Incorrect	Total
Model 2	Correct	a	b	a+b
	Incorrect	c	d	c+d
	Total	a+c	b+d	n

More specifically, considering Table 6.5, the null hypothesis holds true if the marginal probabilities of $p_a + p_b \approx p_a + p_c$ and $p_c + p_d \approx p_b + p_d$. This corresponds to $h_0 : p_b \approx p_c$. The test statistic is then calculated as:

$$\chi^2 = \frac{(|b - c| - 1)^2}{(b + c)} \tag{6.1}$$

which uses the Edwards continuity correction [59] and has a Chi-squared distribution with a single degree of freedom. Based on the threshold value (a), we can compute the probability (ρ) that the resulting differences in accuracies are indeed due to h_0. Probabilities lower than the defined threshold ($a = 0.05$) reject the null hypothesis of statistical similarity in the top-1 accuracy rates of the two models.

Our statistical significance experiments were preformed over the K-400 validation set and were based on the provided models from Tables 6.3 and 6.4. This demonstrates both settings with models trained from scratch shown in Tables 6.6a to 6.6d and state-of-the-art models that have been pre-trained on HACS shown in Tables 6.6e to 6.6g.

From scratch models. The architectures presented in Table 6.4 are further explored based on their statistical significance. The four ResNet variants show that the improvements

Table 6.6. McNemar's statistical significance test on K-400.

CR \ Original	Correct	Incorrect	Total
Correct	8112	659	8771
Incorrect	576	4314	4890
Total	8688	4973	13661

statistic (χ^2) = 5.445
p-Val (ρ) = 1.9e^{-2}

(a) ResNet50 3D

CR \ Original	Correct	Incorrect	Total
Correct	8253	654	8904
Incorrect	557	4197	4754
Total	8810	4851	13661

statistic (χ^2) = 7.610
p-Val (ρ) = 5.8e^{-3}

(b) ResNet50 (2+1)D

CR \ Original	Correct	Incorrect	Total
Correct	8216	703	8919
Incorrect	527	4215	4742
Total	8743	4918	13661

statistic (χ^2) = 24.898
p-Val (ρ) = 6.0e^{-7}

(c) wide-ResNet50 3D

CR \ Original	Correct	Incorrect	Total
Correct	8775	473	9248
Incorrect	134	4279	4413
Total	8909	4752	13661

statistic (χ^2) = 188.211
p-Val (ρ) = 7.8e^{-43}

(d) ResNet-101 3D

CR \ Original	Correct	Incorrect	Total
Correct	9516	878	10394
Incorrect	707	2560	3267
Total	10223	3438	13661

statistic (χ^2) = 18.233
p-Val (ρ) = 1.9e^{-5}

(e) MTNet$_S$

CR \ Original	Correct	Incorrect	Total
Correct	9931	678	10609
Incorrect	538	2514	3052
Total	10469	3192	13661

statistic (χ^2) = 15.889
p-Val (ρ) = 6.7e^{-5}

(f) MTNet$_M$

CR \ Original	Correct	Incorrect	Total
Correct	10176	614	10790
Incorrect	496	2375	2871
Total	10672	2989	13661

statistic (χ^2) = 12.332
p-Val (ρ) = 4.4e^{-4}

(g) MTNet$_L$

achieved with the inclusion of Class Regularisation is not within the margin of statistical error. All of the probabilities of statistical homogeneity between the model accuracies are significantly below the Fisher threshold of $a = 0.05$. We observe that based on the contingency Tables 6.6a to 6.6d, $\rho_i < 0.98, \forall i \in M$, where M is the set of the four tested models trained from scratch. The largest probability margins from a are also shown to

correlate with either networks with larger number of channels per block, such as wide-ResNet-50 or an increased number of layers (ResNet-101). This follows the notion that there is dependency on both the feature complexity as well as the feature size as more complex or a larger number of features can be more discriminative for the recognition of a class. This therefore also impacts the application of Class Regularisation and meaningful feature and class feature correlations can be better explored in more complex spaces.

Pre-trained models. Statistical significance results, on models that have their weights initialised from pre-training on HACS, show similar probabilities to the ones trained with randomly initialised weights. Across the three MTNet variants, the improvements by including Class Regularisation are determined to not correspond to statistical error with a confidence $\gg 99\%$. This enforces the notion that increases in accuracy rates, in both training from scratch and pre-trained settings, are indeed because of CR.

6.5 Discussion and conclusions

In CNNs, feature extraction takes the form of a sequential application of kernels over an input volume. Features are learned in a hierarchical order. Based on this hierarchy, only a small fraction of these features and cross-layer connections would correspond to a specific class. This leaves room for improvement as all of the extracted features are considered uniformly during the final class probability distribution. In this section we have introduced a method named Class Regularisation which can strengthen or weaken layer activations based on the batch of videos that are processed.

Although the regularisation of activations has been explored as a general calibration method for activation distributions in CNNs layers [105, 294], there is a lack of techniques that can utilise class-based information as part of the regularisation process. This relates to features being calibrated individually of its overall descriptive capabilities, in terms of a class. Our proposed Class Regularisation can discover the features that best address a specific class and incorporate this discovered association as part of the training process. Its general application is not bound by the architecture type and utilises information from previous training iterations. We use as inputs both the class weights extracted from the previous iterations, and the activation feature maps of the layer that is to be regularised. The method can discover the class to feature correspondence and apply it over the input activation map creating *Class Regularised* activations.

We evaluate the proposed method over two large-scale datasets: HACS and Kinetics-400, and over two additional training environments. We show that, with the inclusion of Class Regularisation, further improvements on top-performing models can be achieved with minimal additional computations or parameters. These observations hold true for both models that have been trained with their weights being initialised with random values (from scratch), as well as for models previously trained on a different dataset and their weights being initialised from them. We additionally demonstrate that performance difference to the baseline models due to CR is statistically significant.

Additionally, Class Regularisation aids in improving the explainability of 3D-CNNs. Through the direct connections between classes and features, we can visualise the spatio-temporal features that correlate to specific classes and thus provide further insights in terms of the specific patterns that classes best relate to.

We believe that the direct application of Class Regularisation, regardless of the model used, alongside its minimum additional computations can provide significant benefits, both the accuracies and well as the understanding of spatio-temporal CNNs.

Chapter 7

Spatio-Temporal Feature Interpretation

In this chapter we present explainable representations of spatio-temporal features extracted from 3-dimensional convolutions. We argue that the creation of a methodology to provide visual interpretations for spatio-temporal features can aid in the explanation of failure cases given an architecture. For example, visualisations can be used to understand cases of confusion between classes through the analysis of misclassified outputs. [1]

7.1 Introduction

In the previous chapters, we have shown the creation of robust spatio-temporal descriptors through convolutional receptive fields of flexible temporal size. The proposed method achieved this with the introduction of a triple branch approach, addressing both local spatio-temporal features and prolonged patterns of extended durations. The computed features are then aligned based on their overall temporal importance of features in the third branch. We have detailed the principals of our feature alignment method utilising attention discovered with recurrent-based sub-networks over time. We have provided an overview of our gating mechanism based on soft-nearest neighbour and cyclic consistency distance to achieve information fusion between the temporally aligned and the originally discovered patterns. Through this gate, feature fusion is either enabled or disabled based on the features exhibiting cyclic consistency. Our extensive experimentation on both large-scale and smaller datasets has showed significant improvements over fixed-sized alternatives while also further reducing computation costs.

Subsequently, we have presented a direction for improvement as the uniform extraction of features can be limiting for action and human interaction classes that are similar in terms of their overall motion features. The regularisation method that we have presented uses feature amplification over activations in order to incorporate class-based information within the extracted features. Relationships that form between specific features and classes during training can be further exploited in order to strengthen their association and increase intra class variance.

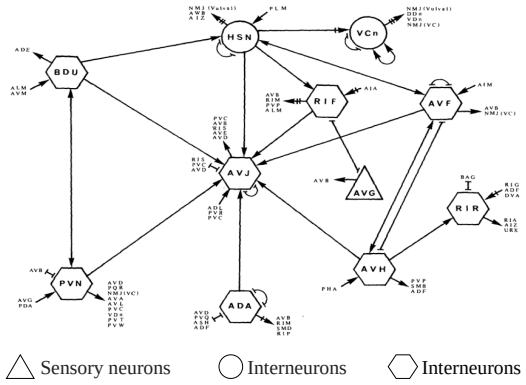
Throughout the experiments in previous chapters, we have shown score-based quantitative improvements across different models. Although quantitative measures such as validation set accuracies can be indicative of the overall performance, they do not provide a qualitative measure to present the effects of each network's selected architectural aspects. The need

[1] The code for *Saliency Tubes* (†) is available at https://git.io/JmXIH and for *Class Feature Pyramids* (‡) is available at https://git.io/fjDCW

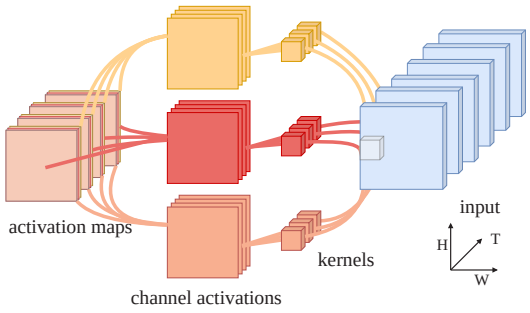
† ‡

for such a measure is further evident by the overall limited transparency of CNNs as models are typically regarded as *black boxes*.

Even though there is only a weak connection between current artificial neurons and biological equivalents, there are commonalities in terms of the methods that aim to understand them. A significant portion of current works in modern neuroscience studies biological neural connectomes mappings [289]. A Connectome mapping can present information paths of the synaptic connections between different neurons and neural types in a visually comprehensible format. However, these representations are limited in a sense that they can only provide neuroscientists with a visual description of the synaptic connections between neurons. They do not provide any information about the magnitude of a synaptic connection nor if the connection excites or inhibits. This is in direct contrast to artificial neural networks where not only it is possible to discover their non-zero weights but also the produced feature activation values. We compare the information mapping in two examples of biological and artificial neural processes in Figure 7.1, where the neuronal circuit of a roundworm's egg-laying process and forward information pass in a spatio-temporal (3D) convolution are shown. Despite their obvious structural differences, a key aspect of artificial neurons in comparison to biological neurons, that benefits their overall explainability is their sequential forward-to-backward information flow. Inversely, gradient computations are calculated in a backward-to-forward manner. The chain rule of backpropagation provides a direct connection between classes and their associated features across different layers.



(a) Caenorhabditis elegan (Roundworm) egg-laying circuitry



(b) Spatio-temporal feature extraction in convolutional layer

Figure 7.1. Examples of Biological neural circuits and convolutional feature extraction. The biological neural circuit (a) represents round-worm's egg-laying process. The spatio-temporal convolution kernels (b) are responsible for the extraction of space-time features. Image (a) is sourced from [289].

Given the hierarchical connection between classes and their layer features, we propose two novel spatio-temporal visual attention methods. We first explore the visual interpretations in the form of the spatio-temporal clip regions that are found to be feature rich for a specific class. Our approach, named Saliency Tubes, can uncover the regions and frames that 3D CNNs rely on, class predictions by utilising the weights of a selected class and their produced feature mask that corresponds to the regional attention of the network.

Based on this method of representing spatio-temporal attention, we further study hierarchically the dependencies of network features. In the extended method named Class Feature Pyramids, we use a cross-layer exploration method, termed *back step*, which constructs an association between high-level features of layers in greater architectural

depths and low-level features in earlier layers. Through back step the entire network can be traversed while capturing the causalities of feature activations between the discovered most informative features of the preceding and succeeding layers. The resulting few-to-many connections across adjacent network layers can be best described as a pyramid-like structure.

In Section 7.2 we overview approaches in the literature for the visualisation of spatial patterns. Advances in spatio-temporal network interpretation methods are discussed in Section 7.3. We overview the main methodology of Saliency Tubes in Section 7.4. The extension to Class Feature Pyramids is then presented in Section 7.5. We conclude in Section 7.6.

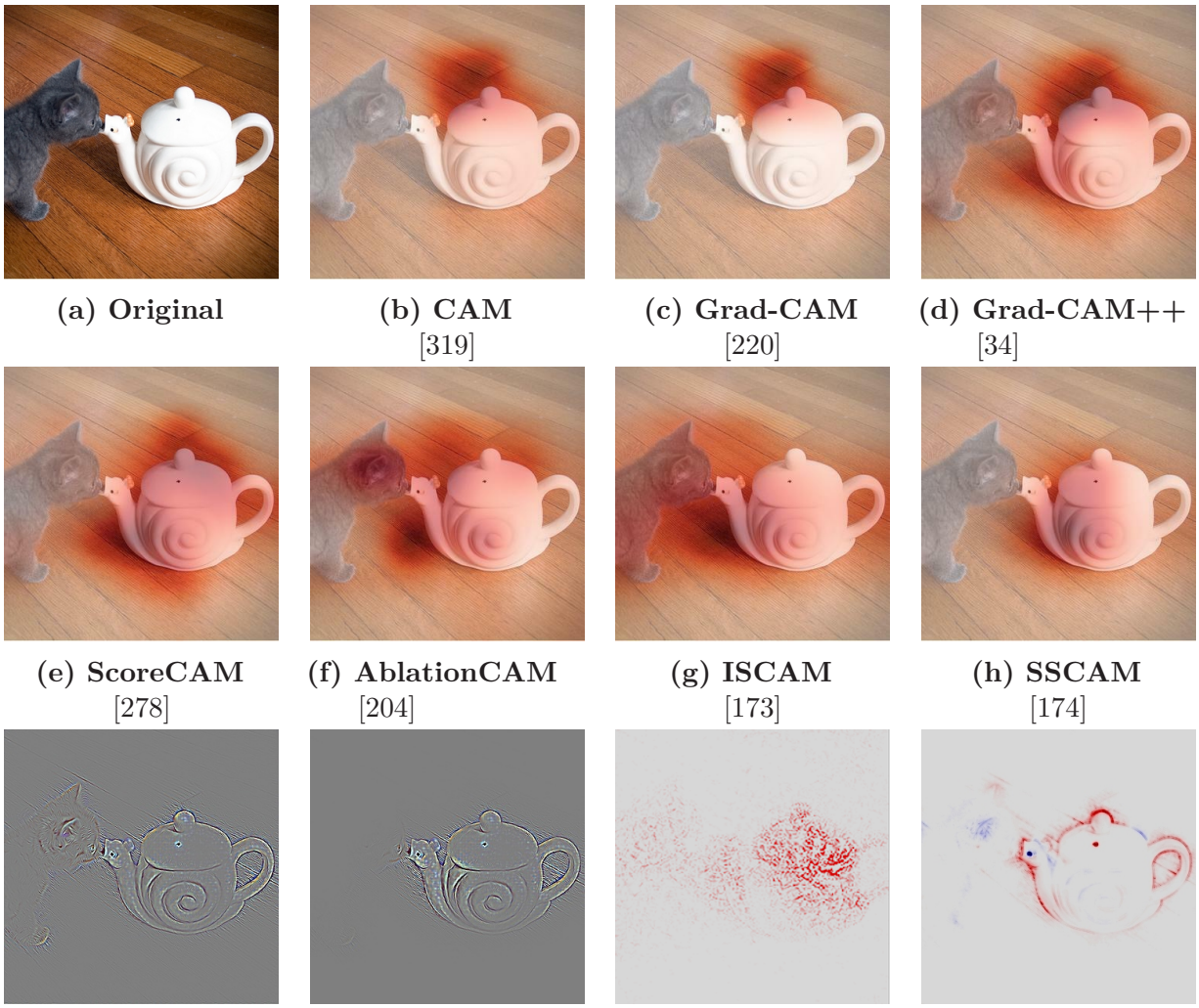
7.2 Spatial convolutional feature interpretations

We focus on visual interpretation approaches for spatial data visualisations. Due to the high popularity of the image-based domain, spatial feature visualisation techniques have been explored in greater depth than techniques for spatio-temporal data. We therefore first overview spatial feature explanation methods. Based on the visualisation tasks, we divide them into three categories. The first set of methods is based on the utilisation of feature activations aimed at providing descriptions for the causes resulting in a certain activation. The second set is for the creation of visualisations that correspond to features based on layer and class weights. The resulting explanations demonstrate the characteristics and features that specific layer weights correspond to. The last set of methods is based on neural attribution. Attributions aim at exploring the influence between neurons of adjacent layers. Examples of activation-based and neural attribution methods are presented in Figure 7.2.

7.2.1 Activation-based visualisations

One of the earliest activation-based methods, *class activation maps (CAM)* [319], was based on localised salient regions in CNN inputs through the combination of feature regions that correspond to features of a specific class. Approaches have further been based on the creation of masks in order to provide a visual representation of a model's class focus [69], while other mask-based methods such as *RISE* [193] use randomly masked versions of inputs to discover an activation correspondence. Class Activation Maps have been generalised with *GradCAM* [220] which instead uses convolutional features that relate to a specific class. This creates a salient mask over the input image. *GradCAM++* [34] introduced a pixel-wise weighting based on spatial positions in the final convolution maps. *Ablation-CAM* [204] uses a similar weighting procedure for feature maps that is based on feature map importance with respect to a specific class. Supplementary approaches have also been used to obtain feature maps during the forward information pass of the original inputs masked by feature activation maps [278]. This results in a score-based weighting of the activation maps combined with the input images (Score-CAM).

Other approaches have considered a *network dissection* [10] in which a measure is created between the alignment of neural units though their produced activations and concepts of objects. This is done with the use of the neural activations as a segmentation mask, measured over an intersection-over-union score for the discovered regions. Further works have also considered the decomposition of classes in terms of their features alongside their respected regions [320]. Similarly, visual attention measures have also led to works



(a) Original | **(b) CAM** [319] | **(c) Grad-CAM** [220] | **(d) Grad-CAM++** [34]

(e) ScoreCAM [278] | **(f) AblationCAM** [204] | **(g) ISCAM** [173] | **(h) SSCAM** [174]

(i) GB [236] | **(j) GB-CAM** [220] | **(k) IG** [250] | **(l) LPR** [6]

Figure 7.2. Feature visualisations. Presented methods are considering activation-based visualisations (top two rows) and neural attribution methods (bottom row).

that include dual modalities [186] with the incorporation of textual justifications for the choices made based on the discovered features. Works have also focused on the creation of supplementary learnable models through sub-modular optimisation with *Local Interpretable Model-agnostic Explanations (LIME)* [207]. Zintgraf *et al.* [322] have visualised both positive and negative feature correlations between image regions and classes.

7.2.2 Extracted weight feature representation

Weight visualisations can provide insights in terms of the type of features that a CNN extracts and associates with a specific class or object category. A popularised approach considers the parametrisation of inputs and subsequently their updates, in order to maximise a specific neuron that corresponds to a class [61]. Although this creates a visual correspondence between classes and their respective features, the representations can often be less intuitive as they highly depend on the values used during initialisation. Zeiler and Fergus [309] proposed a de-convolution approach to address this problem.

Their approach aimed towards the approximation of features. Simonyan *et al.* further explored maximising activations as a visualisation technique where the parameterised image optimised by capturing class features is learned in a supervised manner. Works of Nguyen *et al.* [176] have shown how convolutional features can demonstrate high correspondence to unrealistic visual features based on the vast space of possible images and patterns that are similar to conventionally extracted patterns. Part of this phenomenon is based on *feature entanglement* [182] as convolutional features may not correspond to singular semantic concepts which can be visually understood. Recent work addresses the issues presented by *feature entanglement* through the inclusion of Gaussian filters [305, 277], jitter effect [171] and creating centre-biased gradients [177]. Other recent works have also been based on the creation of a multi-objective task [239] of activation maximisation and activation distance minimisation, (example shown in Figure 7.3) in order to reduce the search space or representations.

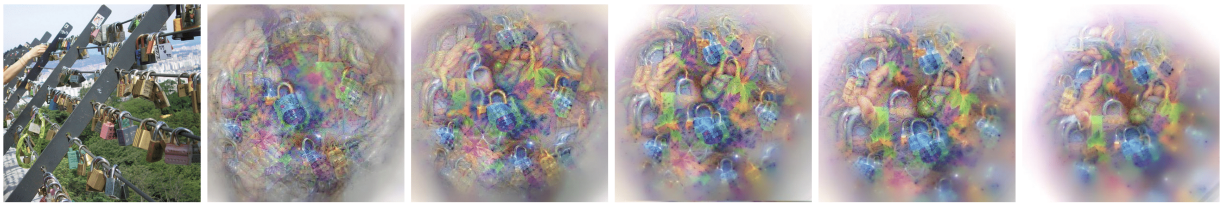


Figure 7.3. Top-30 feature visualisations through weight-excitation.'layer6.12.conv3' from wide-ResNet-101 [308] is used to create the visualisations. Image sourced from [239].

To improve the realism in images, some approaches have considered the use of adversarial networks to synthesise visual features [12]. Based on the approach of utilising adversarial training, Nguyen *et al.* [175] proposed a deep image generator network (DGN) without priors of hidden distributions as in generative models [85, 123]. Although there are improvements in the visual quality of visualisations of generative models, representations produced by generator models lack learned feature causalities [97] as they introduce an additional ambiguity step with the inclusion of the generator sub-network.

7.2.3 Neural attributions

In order to discover the influence of neurons across layers, works of Springenberg *et al.* [236] studied how additional signals from higher layers can be added within the propagation process. The resulting *guided backpropagation* method includes zeroing-out negative gradients and thus creates a mask for the regional features that only display a positive influence. The visualisation of relevant information was also performed by decomposing the resulting class feature vectors into pixel relevance scores that reflect each pixel's contribution to the final class prediction with *Layer-wise Relevance propagation (LRP)* [6]. This approach has been extended to include non-linearities. Such non-linearities include normalisation, which provides additional difficulties in terms of standardising a technique for the backpropagation of relevant information [16]. Similarly, Shrikumar *et al.* [227] proposed a reference state for neurons based on their inputs, creating scores between obtained and reference states. Other approaches considered attributions that include the creation of integrated gradients through the integral of gradients along the information path [250].

7.3 Spatio-temporal features

Despite the large number of visualisation methods for image-based features, there is only a small selection of methods that addresses the video domain. Early works of Karpathy *et al.* [117] were aimed at the creation of visual representations for LSTM recurrent cells. Based on this, Bargal *et al.* [9] have explored visual explanations in action recognition with models that incorporate 2D Convolutions and LSTMs, with their main method being an extension of *Excitation Backpropagation* [313]. However, both works focus on the decisions of recurrent cells within the context of action recognition with 2D convolutions being used as per-frame feature extractors.

Visualisations of spatio-temporal features in action recognition have proven challenging. Early works by Chattopadhay *et al.* [34] have extended class activation maps for object recognition. Further works also included extensions of Layer-wise Relevance Propagation (LRP) [237] by indicating the spatio-temporal locations that are deemed more important for each video class. Similar works [94] have also extended Deep Taylor Decomposition (DTD) [170], showing positive and negative gradient relevance in inputs. Hiley *et al.* [93] have also demonstrated relevance through DTD in a spatial-only and temporal-only context for 3D convolutions. Li *et al.* [146] have utilised Extremal Perturbations [68, 69] which are based on a learned mask that specifies the region that maximises the class probability. Further works of Price and Damen [199] have used a frame-wise Shapley value [224] to determine the contribution of each frame to the final class prediction made by the model.

In our proposed methods we study spatio-temporal convolutions in a hierarchical manner. With the proposed *Saliency Tubes* as a method tailored for the representation of spatio-temporal informative regions and the proposed *back-stepping* technique in *Class Feature Pyramids*, we can effectively traverse multiple network layers and generate a class dependency graph based on the most informative features.

7.4 Saliency Tubes feature visualisation

Figure 7.4 outlines the proposed approach for spatio-temporal salient region localisation. We denote activation maps with a, while layers are indexed by $[l]$. The activations ($\mathbf{a}^{[l]}$) for the final convolution layer ($[l]$) include activation maps of size $C' \times \mathbf{R}$, where C' is the number of corresponding channels, and $\mathbf{R}$ is the spatio-temporal region of T' temporal extent, height H' and width W' (with size $|\mathbf{R}| = T' \times H' \times W'$). We note that C' is also based on the total number of convolutions that are performed in layer l. The tensor in the final fully-connected layer that is responsible for class predictions is denoted by $\mathbf{y}$. Each class ($i \in \{0, .., N\}$) from the total N classes can then be selected as $\mathbf{y}_i$. Each channel ($j \in \{0, ..., C'\}$) in a class's (i) prediction vector is formulated as $\mathbf{y}_{i,j}$ and relates to a specific feature of the network's activations map $\mathbf{a}^{[l]}$ and designates how informative that specific activation map is towards a correct prediction for an example of class i. To relate the importance of feature i to spatio-temporal input regions, we propagate back to activation maps ($\mathbf{a}^{[l]}_{j \times T \times H \times W}$) and multiply ($\otimes$) all of the map elements by the equivalent prediction weight vector $\mathbf{y}_{i,j}$. The normalised class weighted operation is the formulated as $\mathbf{z}_{i \to j}$:

$$\mathbf{z}_{i \to j} = \frac{f(\mathbf{a}_j) - \min\limits_{\mathbf{a}_j \in \mathbf{R}} f(\mathbf{a}_j)}{\max\limits_{\mathbf{a}_j \in \mathbf{R}} f(\mathbf{a}_j) - \min\limits_{\mathbf{a}_j \in \mathbf{R}} f(\mathbf{a}_j)} \quad where\ f(\mathbf{a}_j) = \mathbf{y}_{i,j} \otimes \mathbf{a}_j \tag{7.1}$$

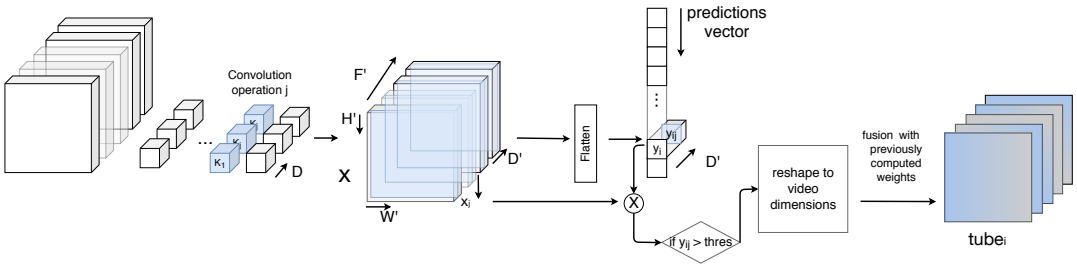


Figure 7.4. Saliency Tubes main process pipeline. Feature importance of class i is defined based on the corresponding class prediction values in feature vector $\mathbf{y}_i$. Individual features are visualised through the fusion of the activation maps in the last step of the saliency tubes. The figure presents a simplified case for a single feature in the final convolution layer for easier interpretability.

Considering the large number of features (C') and in order to limit the effect of low-information regions, we introduce a threshold argument τ. Based on it, only significantly contributing activations are selected. Values that fall below that threshold are defined as parts of set $\mathbf{E}$. We thus consider that the activation weight operation ($f(a_j)$) for channel j also includes the condition that $f(\mathbf{a}_j) = \mathbf{y}_{i,j} \otimes \mathbf{a}_j \; \forall j \in \{0, ..., C'\} \notin \mathbf{E}$.

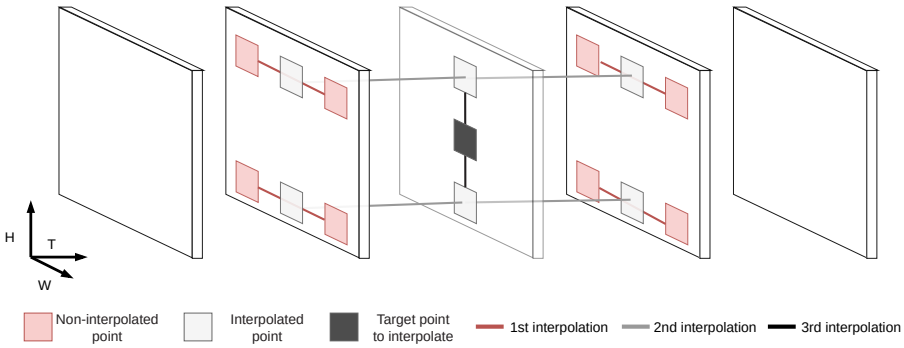


Figure 7.5. Spatio-temporal interpolation. Original points are denoted with red and interpolated points are shown as light and dark grey.

As the produced normalised class-weighted activations ($\mathbf{z}_{i \to j}$) are of spatio-temporal size $T' \times H' \times W'$, we use third-order spline interpolation in order to approximate the activation mask in an extended size of $T \times H \times W$ to fit the input. Formally, we define each 3-dimensional point of the class activations $z_{i \to j,t,h,w}$ as a point ("knot") for a non-continuous (g) function. Each point is included in the function with C^2 smoothness that has a continuously differentiable first derivative. We visualise how interpolation is performed over space and time in Figure 7.5, where original/non-interpolated points are denoted with red. The interpolation process is first done spatially with the approximation of new points (light grey between red boxes) and the subsequent increase of the spatial dimensionality. The second set of interpolation operations is performed in order to discover points across time based on the knots from the first interpolation (grey points in the middle frame). The final target point is then discovered by the third interpolation sequence.

The final spatio-temporal attention volume, named Saliency Tubes, ($\mathbf{tube}_i = \sum_{j \in C'} (z_{i \to j})^2$) is created by the channel-wise summation of the squared, normalised class-weighted activations ($\mathbf{z}_{i \to j}$). We use the square of the class-weighted activations as a straightforward amplification of salient regions to improve visualisation quality. We present visual examples in Figures 7.6 and 7.7.

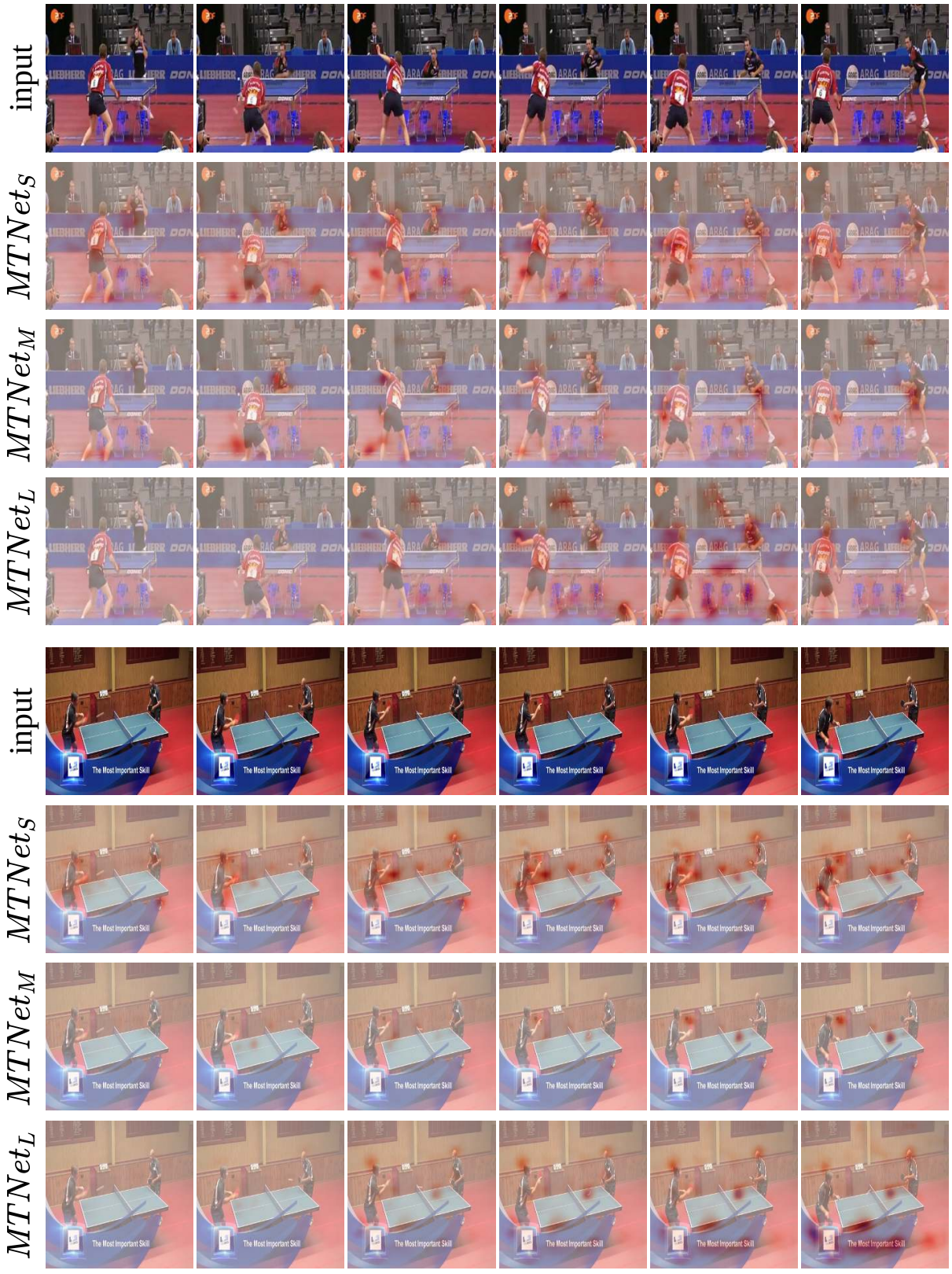


Figure 7.6. Saliency Tubes on MTNets [247] for HACS 'table tennis' class. Both examples have been randomly sampled from HACS.

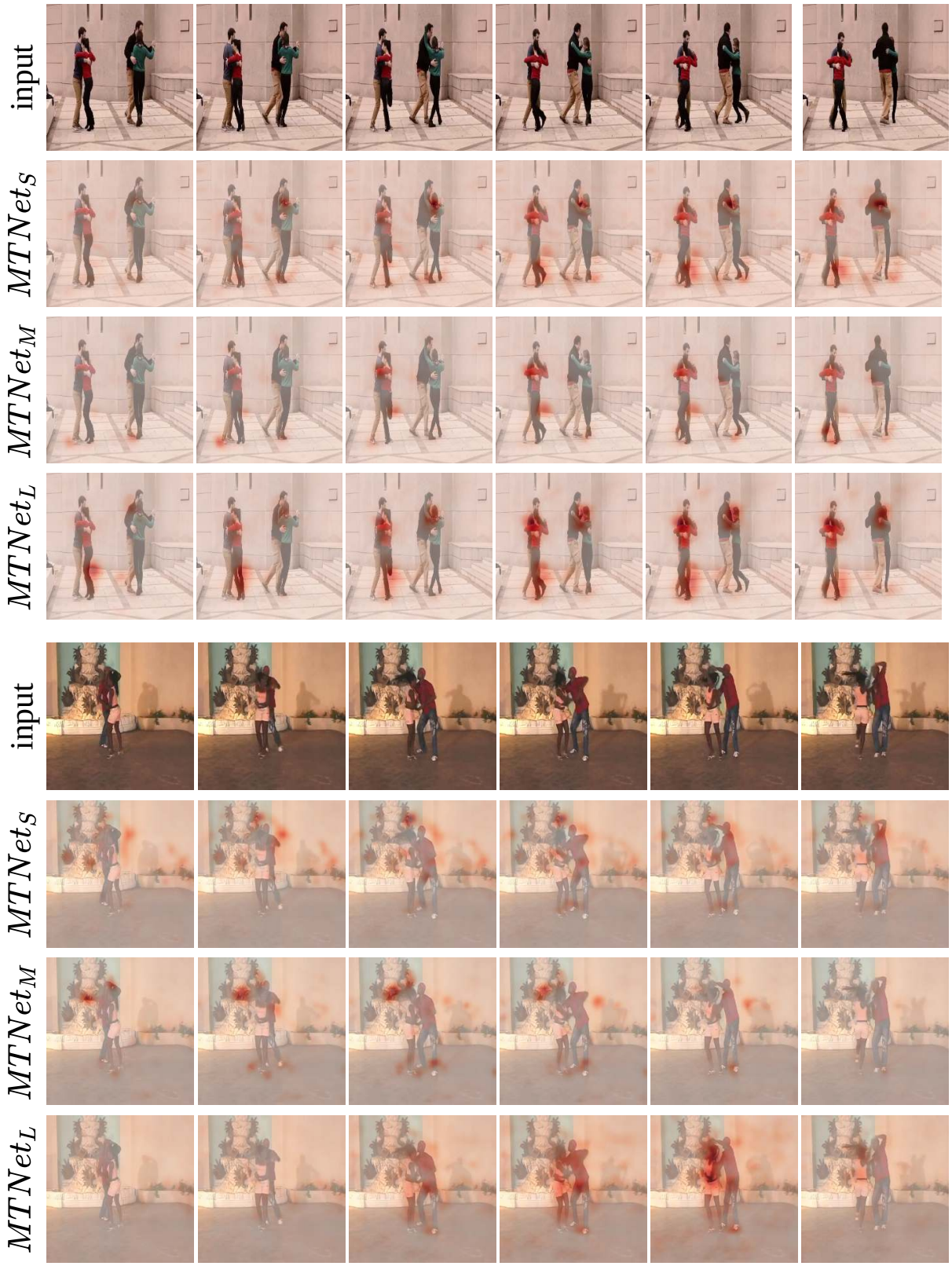


Figure 7.7. Saliency Tubes on MTNets [247] for HACS *"tango"* class. Both examples have been randomly sampled from HACS.

7.5 Class Feature Pyramids

While Saliency Tubes provide a straightforward method for visualising class-informative feature spatio-temporal regions, they also include certain limitations. Their most notable restriction is their constraints to only providing visual interpretations for the final convolutional layer due to the fact that the channel dimensionality directly relates to the number of channels of the class prediction vector. This can be informative to an extent, as the features captured by later network layers include a greater degree of complexity. However, they do not provide a broader view for the entire network. The second shortfall of Saliency Tubes is the fact that visually salient classes also display similar salient regions as they share the majority of class feature activations. Distinctions between the two classes are mostly present over a small number of activations that are specific to each class. However, these cannot be visualised without a greater exploration of features across the entire network architecture.

In our proposed extension of Saliency Tubes, we aim to address the issues of visualising class-specific features of different convolutional layers. Our Class Feature Pyramids (CFP) enable a hierarchical traversal over network layers motivated by their linear process of kernel application over activation maps. Through our method we reverse this process to enable the propagation of information over the network from the prediction layers towards earlier network layers. We term this approach back-step as it corresponds to the discovery of earlier layer kernels that produce feature maps associated with a specific class.

7.5.1 Class feature association

We first consider the class weights of the final prediction layer ($\mathbf{w}^{[p]}$) with ($[p]$) denoting the prediction layer. In order to identify the maximum probability class (c) we apply a standard softmax activation and select the corresponding class-weight vector ($\mathbf{w}_c^{[p]}$). This can be seen as the right-side red coloured tensor in Figure 7.8. The discovery of individual effects that each feature of the class weight vector has over the activation map ($\mathbf{a}^{[l]}$) of the final convolution layer ($[l]$) is performed similarly to Equation 7.1. This is achieved with the multiplication of each class weight vector channel and activation map. Based on this, we create a class-based activation map ($\mathbf{a}_c^{\star[p]}$) that has the same dimensionality as the initial input activation map. However, each of the channels represents the relationship between channel C' and the selected class c:

$$\mathbf{a}_c^{\star[p]} \overset{(7.1)}{=\!=} \{\mathbf{z}_{c\rightarrow j}\}\ \forall j \in C' \tag{7.2}$$

With the aggregation of class-based normalised information in $\mathbf{a}_c^{\star[p]}$, we then explore the channel-wise dependencies of the class and features extracted in a specific layer. This is done to address probabilistic distributions of small and broader scales. In contrast to the one-hot selection of classes, multiple features affect a specific class, thus the selection process needs to include multiple values. To this end, we apply a monotonic shifted logistic sigmoid function with a user-defined threshold value (θ):

$$\mathbf{feats}_i = \{i\ :\ \mathbf{F}_i^{[p]} > 0\}\ where\ \mathbf{F}^{[p]} = \frac{1}{1 + e^{-(\mathbf{a}_c^{\star[p]} + \theta)}} \tag{7.3}$$

The discovered most dominant feature indices ($\mathbf{feats}_i$) are identified from Equation 7.3. These indices have a direct correspondence to the feature locations from the previous layer ($[l]$) outputs, which in turn also relate to respective kernels ($\mathbf{k}^{[l]}$). Through this sigmodial

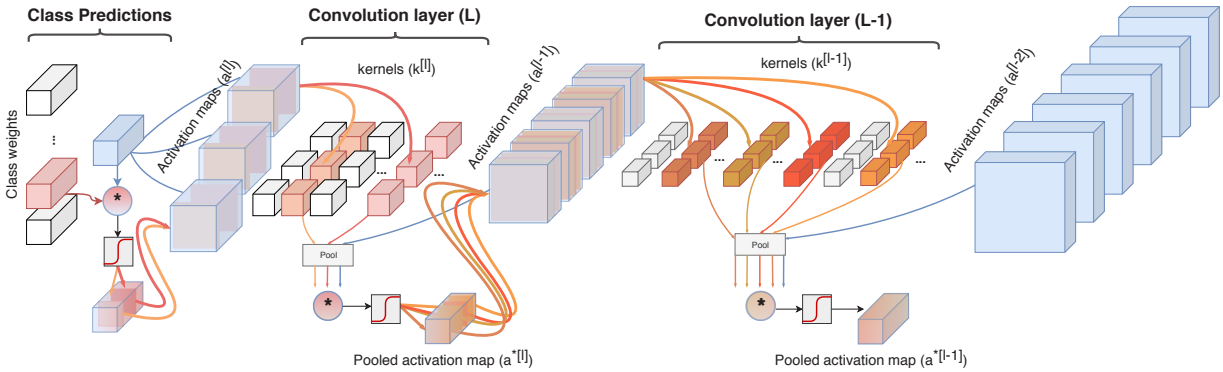


Figure 7.8. Back-step process. Averaged pooled versions of layer weights ($\overline{\mathbf{k}'}^{[l]}$) and their input activation maps ($\overline{\mathbf{a}}^{[l-1]}$) are utilised for the channel-wise creation of global space-time vector representation. Through this, the locality of convolutions is alleviated. With the element-wise multiplication ($\otimes$) of the kernels and activations, pooled class-based activation maps ($\mathbf{a}^{\star[l]}$) are created that only correspond the selected features. The highest feature activations that relate to the class are selected through a sigmoid applied over the volume. The same process is applied iteratively to discover each high activation in the current layer.

selection process, the relationship of each channel in the activation volume $\mathbf{a}^{\star[p]}$ and kernels from previous layer ($[l]$) can be directly discovered. The influence of each of the kernels in the layer ($\mathbf{k}^{[l]}$) to the final class prediction is based on the aforementioned logistic function which is visualised as the orange cross-layer connections in Figure 7.8.

7.5.2 Layer dependencies

In comparison to propagating class-related information directly, the propagation of class-related information through corresponding features in the previous layer includes some difficulties. As the spatio-temporal extent of activations changes across layers alongside it's number of channels, a direct correlation measure is not straightforwardly achievable. This is associated with the curse of dimensionality problem as high-dimensional signals (e.g., deeper layer activations) cannot be linearly mapped to lower-dimensional spaces (e.g., earlier network layers) and vice versa. Even without considering the spatio-temporal size difference between activations from different layers, their convolution operations are followed by non-linearities and with that, the problem of representing cross-layer information also persists. In addition, the hierarchical architecture of CNNs includes a strict locality of the operations performed in each layer. As the input volume decreases in spatio-temporal size, the space-time patches that are used by each set of kernels increase in size, making the discovery of a cross-layer relationship difficult.

Class Feature Pyramids deal with the above issues by considering class information in a global manner, for the entirety of the activation maps. The localities of kernels, as well as their respective activations, are transformed to include accumulated information over the entire spatio-temporal regions. Specifically, given the discovered feature indices $\mathbf{feats}_i$ from Equation 7.3 and in order to back-step from layer $[l]$ to $[l-1]$, we first select the respective kernels ($\mathbf{k}'^{[l]}$). Channels (C) for layer $[l]$ are denoted with $C^{[l]}$. Feature traversal is then performed based on the average pooled version of activations from the previous layer ($\overline{\mathbf{a}}^{[l-1]}$) and selected layer kernels ($\overline{\mathbf{k}'}^{[l]}$):

$$\mathbf{a}^{\star[l]} = \overline{\mathbf{a}}^{[l-1]} \otimes \overline{\mathbf{w}}_j^{[l]},\ \forall\, j \in \mathbf{feats}_i \tag{7.4}$$

$$where,\ \overline{\mathbf{a}}^{[l-1]} = Pool(\mathbf{a}^{[l-1]})\ and\ \overline{\mathbf{k'}}^{[l]} = Pool(\mathbf{k'}^{[l]}) \tag{7.5}$$

Through Equation 7.4 we create a relationship between class activation maps ($\mathbf{a}^{\star[l]}$) of layer $[l]$ and the previous layer's ($[l-1]$) activations ($\mathbf{a}^{[l-1]}$) with the absence of feature locality. Our approach differs from the standard convolutional procedure where created activations use the dot product of the previous layer activations and the current layer kernels. By replacing the dot-product with a single element-wise multiplication, we can solve the channel-dimensionality reduction problem and discover cross-layer feature dependencies. The main functions of Equations 7.2 to 7.4 are used for every adjacent layer pair until a user-specified *back-stepping* depth that is searched. Previous layer activations ($\mathbf{a}^{[l-1]}$) then follow the same normalisation as in Equations 7.1 and 7.2 to produce a kernel-based activation map ($\mathbf{a}^{\star[l-1]}$). The threshold (θ) and feature indices selection ($\mathbf{feats}_i$) processes remain as previously defined.

7.5.3 Feature and layer-wise associations

An elemental part of a network's explainability is the understanding of the effect of kernels ($k^{[l]}$) at layer ($[l]$), individually or in a group, to previous layer ($[l-1]$) feature activations. These requirements become detrimental in earlier layers of lower complexity as specific activation maps have multiple strong associations with feature extractors in deeper layers. This increase in meaningful connections in earlier layers prevents the creation of an overall coherent channel-wise dependency graph. For this reason, and in order to ensure that an intuitive measure for channel selection is used during the back-step process, channel relationships in different layers can be explored either feature- or layer-wise. In the feature-wise approach, the correspondence of singular channels over the previous layer activations are explored. In the layer-wise process all discovered informative channels of a layer are grouped together and their averaged relationship to the previous layer studied.

Feature-wise relevance back-step. Individually, kernels from layer ($[l]$) are discovered based on the list of informative feature indices ($\mathbf{feats}_i$) from Equation 7.3. Each kernel can correspond to a specific activation map from Equation 7.2 (($\mathbf{a}_k^{\star[l]}$, where $k \in \mathbf{k'}_j^{[l]}$)). The *back-step* process is then applied individually for each of the features (k) and, respectively, the maximum corresponding channels are selected. A possible limitation of using a feature-wise propagation approach, especially in networks with a large number of channels, is the overall feature indices duplication. As features are explored individually, channels in the previous layers are more likely to form multiple strong connections with multiple features of the preceding layer. For this reason, feature-wise relevance is better suited for cases where the *back-step* depth remains small.

Layer-wise relevance back-step. Similarly to the singular kernel-oriented approach, layer-wise feature exploration is performed based on the list of informative features ($\mathbf{feats}_i$). However, instead of a per-feature activation map, a concatenated version of the activations is created. Apart from limiting the number of feature duplications on the *back-step* process with the compact feature representation, layer-wise propagation includes an additional advantage. Kernel selection in the previous layer is done based on the average of multiple features, providing a more robust representation in terms of the region that a layer finds salient. In addition, as multiple features are considered at once (many-to-many), the

activations in the following layer also present higher values than those discovered by singular features (one-to-many).

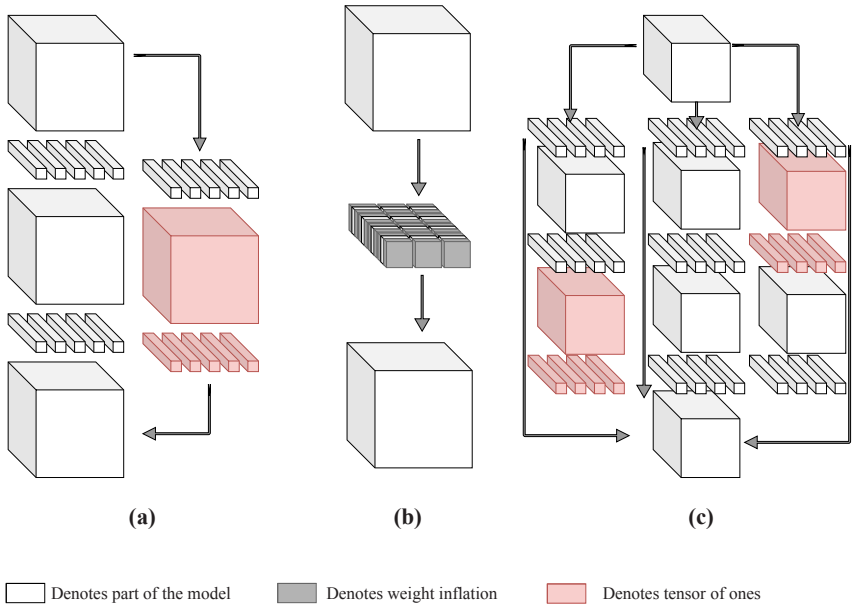


Figure 7.9. Back-step on different convolution block types. (a) **Residual connections** [90], created activations and kernels of ones are in red. (b) **Grouped convolutions** [297], inflated kernels are in grey. (c) **Convolutions in branches** [253]. The branch with the maximum number of convolutions is selected as the base with tensors of ones added to the other branches.

7.5.4 Addressing convolutional block structures

As CNNs are architecturally different in terms of their blocks, we extend our method to address the traversal within popular block variants. We discuss how back-step can be used over convolutions applied in activations in parallel and with varying kernel and channel dimensions performed at the same time. Our extension focuses towards the inclusion of these three convolution processes which exhibit a high degree of complexity in terms of how information is connected and how convolutions are performed. We provide an illustration of our approaches in Figure 7.9.

Residual connections. One of the most widely used architectural building blocks in CNNs is based on the use of residual connections [90]. These blocks are also included in a large number of works by the video recognition community e.g., [53, 64, 63, 89, 202, 261]. In the bottleneck variant, *back-step* is more complex as information is divided between two paths. Subsequently, this does not permit a hierarchical discovery of feature dependencies across the network structure. Our adaptation to the current method for such cases is the inclusion of one-valued weight and activation tensors. Value of one is preferred given its multiplication property and our use of element-wise multiplication ($\otimes$). Through this, all discovered feature indices in the previous layer with high activations are passed to the next layer directly. Therefore, activation indices are shared between previous network layers in the residual branch in the same manner as those in the main pathway of the block.

Grouped convolutions. Convolutions can also be performed over groups of features [297]. Works such as *Channel Separated Networks* (CSN) [260] use grouped 3D convolutions

for action recognition. These convolutional types demonstrate significant challenges during *back-step* as their kernel sizes are composed of a sub-set of channels. This does not allow for an immediate correspondence between the kernel channels and the entire activation volume from the previous layer. We deal with this by explicitly inflating each of the grouped kernels to the same dimensional space as the activation maps by using values of zero to cancel out any effect. We can thus simulate the channel-wise convolution process in the channel dimensional space of the input.

Convolutions in branches. This block decouples information into multiple branches [253]. The branch-based approach has also been extended with success for video classification e.g. [31, 39, 64, 247, 283]. The main mindset is the application of convolution or pooling operations within the block, over multiple pathways originating from a single activation map. Variations to the type of operations and their number add an additional degree of ambiguity for constructing such blocks in a hierarchical manner. In these cases, back-step through branches and pathways are accomplished with the addition of kernels with values of one and activation maps that act as small sub-structures that pad all branches to the same length.

7.5.5 Inference calculation

As Class Feature Pyramids are based on layer traversal, we study the times required by architecturally different spatio-temporal networks to traverse over the network and identify the kernels that produce the highest activations for a specific class.

We report inference times on nine architectures. Our evaluation includes the number of GFLOPs alongside the total running times for feature discovery. We summarise the results Table 7.1. The choice of threshold values (θ) is empirically discovered based on the produced number of informative features for each of the networks. Apart from the network architecture and the threshold value, latency also depends on the number of layers that class features are back-stepped to.

Network	GFLOPS	Back-step time (msec)	# layers	θ
Multi-FiberNet [39]	22.70	24.43	3	0.6
I3D [31]	55.79	23.21	1 + mixed5c	0.65
ResNet50-3D [89]	80.32	21.39	3	0.55
ResNet101-3D [89]	110.98	39.48	3	0.6
ResNet152-3D [89]	148.91	31.06	3	0.6
ResNeXt101-3D [89]	76.96	70.49	3	0.6
MTNet$_S$ [247]	11.30	26.32	3	0.67
MTNet$_M$ [247]	14.12	32.17	3	0.67
MTNet$_L$ [247]	20.82	34.65	3	0.67

Table 7.1. Inference times and GFLOPs. Threshold value (θ) is determined by the model complexity. All architectures are back-stepped for three layers. For I3D, this corresponds to the class filters and the last *mixed* block.

As shown in Table 7.1, the proposed method can traverse across layers at reasonable speeds without significant additional operations over normal forward information passes. In most cases, times are similar or faster to those of simple feed-forward operations. This is primarily attributed to the speed-up gained by using a global representation instead of the dot product of local convolutional operations.

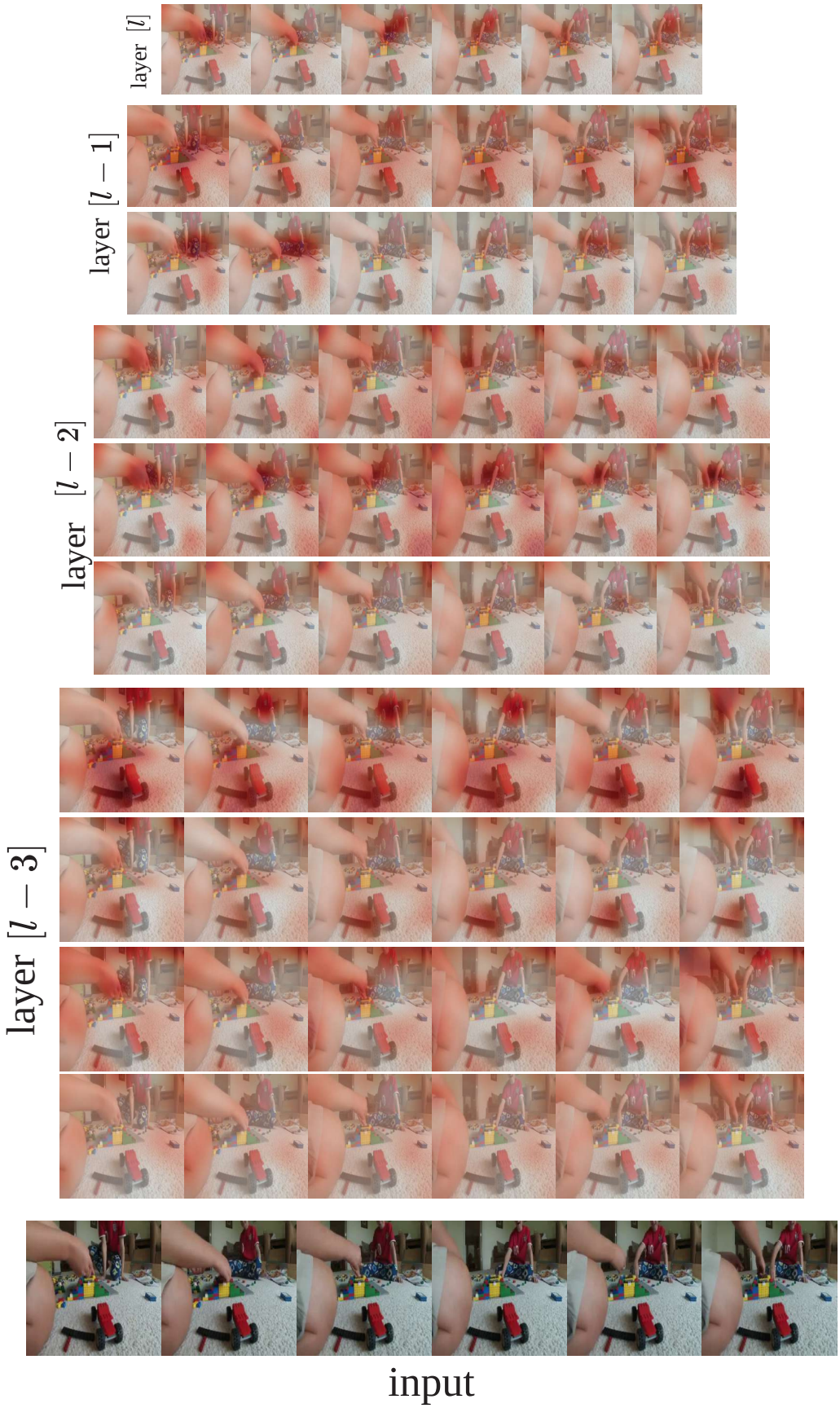


Figure 7.10. Kinetics *"building lego"* class. MTNet_L [246] is used to produce the visualisations.

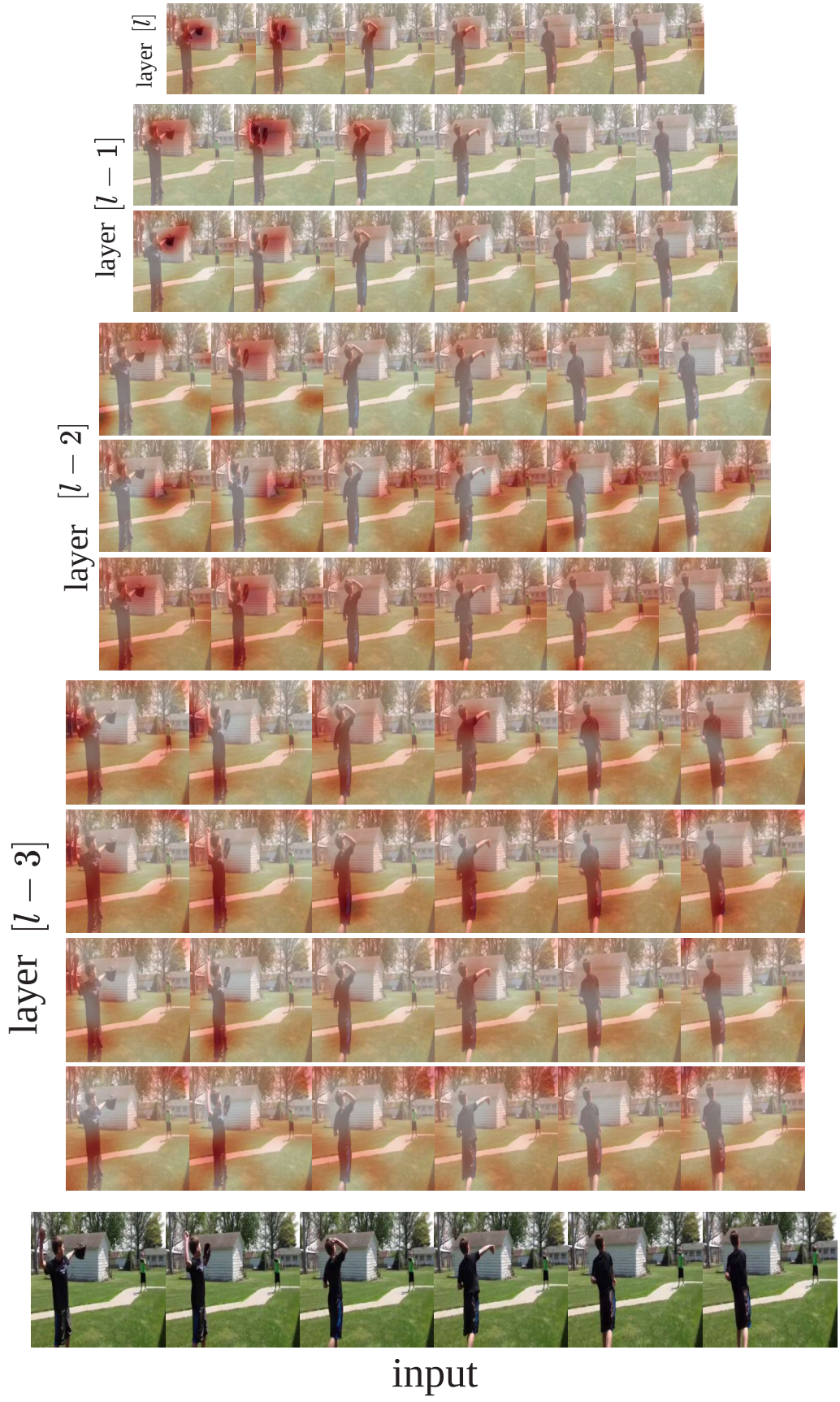


Figure 7.11. Kinetics *"throwing or catching baseball"* class. MTNet_L [246] is used to produce the visualisations.

7.5.6 Visualisation results

In this section we visualise the resulting salient regions for different layer features of MTNet_L based on feature-wise relevance *back-step*. Visualisations in Figures 7.10 and 7.11 have been produced with the same threshold value as in Table 7.1. Additional results on layer features based on the layer-wise approach are shown in Figure 7.12 where we demonstrate the concatenated salient features of the final three layers of five architectures.

Based on the produced layer-wise feature activation visualisations in Figure 7.12, for the Kinetics-700 *"playing field hockey"* class, salient regions are shown to vary significantly between different networks. Given this, layers that incorporate temporal information of different scales within their architecture, such as SlowFast [64] and MTNets [247], focus on more specific spatio-temporal regions. Evident by the salient regions of MTNet_L layers in Figure 7.12f, the network demonstrates that an association is created between the field hockey ball, the ball's movement in the field and the players in the scene with the specific class. Through this hierarchical representation of information, we can further visualise the layers where associations to appearance cues are made. For example, both layers $[l-2]$ of MFNet and SlowFast in Figures 7.12d and 7.12e show that the field's grass also attributes to the class prediction if in comparison to the *"ice hockey"* class.

7.6 Discussion

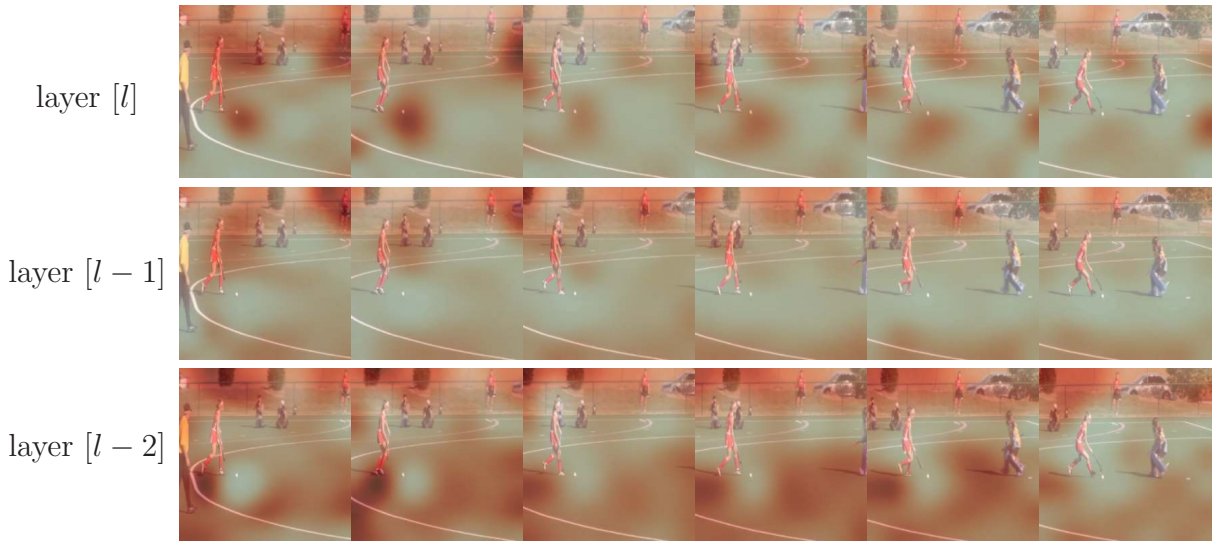
Research in the domain of human action recognition has seen tremendous progress with the introduction of spatio-temporal (3D) convolutions. Still, as the complexity of CNN architectures increases, so does the demand to understand the strength and limitations of these networks in terms of the spatio-temporal patterns that they can model. In this section, we have proposed a method that can provide visual interpretations of class-based features of 3D CNNs and overviewed its extension for features over multiple layers.

The representation of salient regions based on extracted features has been a widely popularised method in image-based CNNs [220, 319]. Their main role is to provide a visual description of the regions which causes a specific feature activation. In our proposed Saliency Tubes, we adapt this method in order to create such representations over space-time volumes. Through the interpolated class-based activations, the most informative spatio-temporal regions in the input can be presented over the original input clip. However, similar to other activation-based visualisation methods, the representations are only specific to the features based on which class predictions are made. This does not provide a higher understanding of the entire feature extraction process within the entire network.

We address these issues through an extension of Saliency Tubes to visualise class features over previous layers. The proposed method named Class Feature Pyramids can hierarchically traverse a network backwards through the discovery of feature dependencies across adjacent layers. This process is termed back-step and enables the exploration of features in any network depth. The proposed method is applicable to various convolutional blocks through invariance with implementations that can be used for residual connections [90], channel-separated (grouped) convolutions [297], and convolutions over branches [253].

We believe that the current limited research on the explainability of 3D-CNNs restricts a high-level understanding of the feature types that are learned. Our proposed methods aim to uncover features, providing a level of transparency that can aid the comprehension of extracted 3D-CNN patterns. We believe that visual interpretation methods can significantly benefit the action recognition field and uncover new research directions.

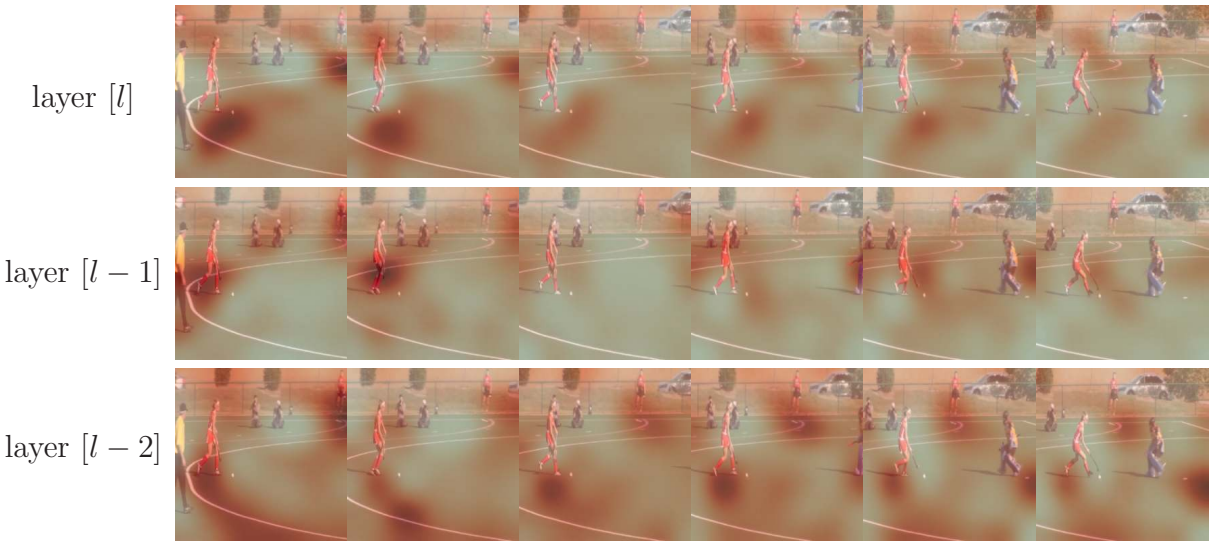
(a) R3D-50



(b) R(2+1)D-50

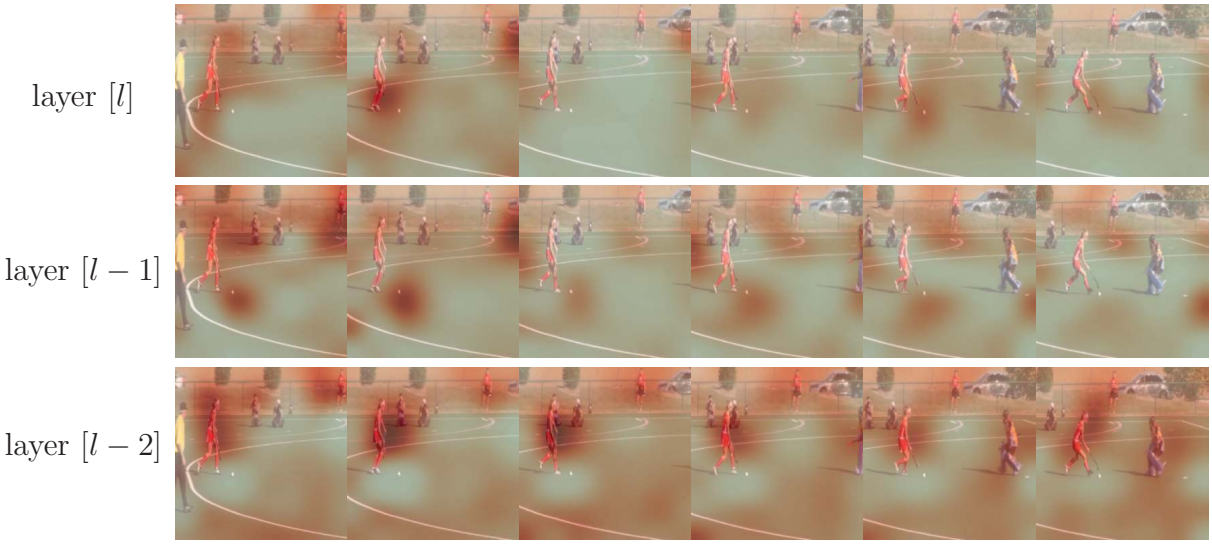
(c) I3D

Figure 7.12. Class Feature Pyramids visualisations.



(d) MFNet

(e) SlowFast 50



(f) MTNet$_L$

Figure 7.12. Class Feature Pyramids visualisations.

(g) Original clip

Figure 7.12. Class Feature Pyramids visualisations. Each model architecture is denoted by the respective sub-caption. Rows represent the layer-wise visualisations produced for each layer. The sample is drawn from *"playing field hockey"* from Kinetics-700. The theta (θ) values for each of the presented networks are [0.6, 0.6, 0.7, 0.65, 0.65, 0.67]. Layer [l] denotes the final layer for each of the architectures.

Chapter 8

Discussion and Future Research Directions

In this thesis, we have investigated the variance in human action performance. In particular, we have addressed the temporal variations from different performances of actions through the extraction and fusion of features over different spatio-temporal sizes (Chapters 4 and 5). We additionally studied the hierarchical correlation between classes and extracted features. We have enforced this connection between features and classes, through the regularisation of features based on their correspondence to the target class in Chapter 6. We finally provided visual interpretation methods for a qualitative assessment of the features and the attention regions that are found to be descriptive of specific action sequences in Chapter 7.

In this final section we summarise the contributions of our works. We then address the most prominent aspects and directions that can motivate future works in the field of human action and interaction recognition.

8.1 Summary

In this section we provide a summary for each of the works that we have presented in the chapters of this thesis.

8.1.1 Squeeze and Recursion Temporal Gates

In Chapter 4 we have studied the problem of temporal variations in human actions. As there is no typical duration length or number of frames for an action, the heavy dependence of 3D kernels in CNNs to extract temporally fixed-sized features can be improved. We have created an alignment between patterns of short motions and the movements that are performed over the entire video. The proposed SRTG method can model the relevant local feature information across the entire video through the use of recurrent cells. Recursion is performed by 2-layer LSTMs over the spatially averaged vector of feature activations. The fusion to the spatio-temporal activation volume is achieved based on the temporal cyclic-back consistency between the LSTM outputs, that include global motion information, and the vectorised convolutional features that were used. This process can achieve a degree of similarity between the purely local and global attention-based activations. Results on five benchmark datasets have shown notable improvements between SRTG and non-SRTG architectures, while the effects of including SRTG in terms of the computational complexity are minimal.

8.1.2 Multi-Temporal convolutions

In Chapter 5 we have further worked on addressing temporal disparities in the performance of human actions. We showed how the sets of local motions do not necessarily correspond to or accurately describe the action that is performed. Through variations in the temporal size of these action sequences, better descriptions can be provided. We therefore proposed Multi-Temporal convolutions (MTConv) that discover feature descriptors of both short local patterns and spatio-temporal features of different durations. The proposed method utilises three branches. The local branch focuses on spatio-temporal patterns that are performed over short space-time windows similar to standard 3D convolutions. The prolonged branch models spatio-temporal features of extended durations and spatial sizes. This is achieved by sub-sampling the activation volume to decreased dimensions. Features of both branches are aligned in the global aggregated feature importance branch based on their temporal motion dynamics across the entire video. The branch uses Squeeze and Recursion to discover the temporal attention of features from the other two branches and capture the local feature dependencies within the global scale of the entire video. Based on these convolution blocks, we have introduced MTNets that include MTConvs in a X3D backbone. MTNets achieve comparable or, in many cases, higher classification accuracy than state-of-the-art models on widely used action recognition benchmark datasets. With MTNets a notable reduction in terms of computation costs can be achieved.

8.1.3 Class Regularisation

The work described in Chapter 6 addressed the discriminative nature of CNNs that is limited to the final fully-connected layers. We studied the effects of enforcing the propagation of class-specific activations throughout the network. We proposed a method named Class Regularisation that relates class information to extracted features through direct connections between the class prediction layer and the convolutional layers. Class Regularisation also benefits the non-linearities of the network by modelling the effects of activations. To avoid the vanishing gradient problem, and the possibility of negatively influencing activations, the weights are normalised in a range given an affection rate value. The method is evaluated on standard benchmark action recognition datasets and with six models demonstrating results with and without Class Regularisation. Models across all datasets showed improvements when including Class Regularisation with minimal additional computational costs over the original architectures. In addition, Class Regularisation can aid in the creation of explainable 3D CNNs. Qualitative visualisations reveal which spatio-temporal features are strongly correlated to specific classes. Such analysis can be made for specific layers and, as such, provide insight into the discriminative patterns that specific features represent.

8.1.4 Spatio-temporal feature visualisation

In Chapter 7 we have overviewed the current limitations of research methods in explaining the spatio-temporal features of 3D CNNs. This is partly due to a scarcity of proper visualisation methods that specifically address the temporal dimension of space-time video data. For this we have proposed two methods. The first method, Saliency Tubes, visualises the feature activation of 3D CNNs with relation to a class of interest. Saliency Tubes were built upon 2D CNN methods relating to network interpretability, and extended for 3D convolution to represent salient spatio-temporal regions. These regions correspond to

the most discriminative class features of the network. We then extended Saliency Tubes to Class Feature Pyramids (CFP) that capture and present hierarchically informative features over different layers. CFPs are independent of the network type and can be employed regardless of the type of 3D convolution operation. They additionally enable the visualisation of activations in layer-wise, group-wise or kernel-wise formats. Our method is therefore suitable to visualise and, consequently, to better understand what kinds of features are learned to identify a specific class.

8.2 Limitations and Future directions

The past years have seen an exponential growth in the potential of video understanding systems and models for tasks, such as intelligent video indexing to smart surveillance. The inclusion of temporal information has shown great improvements in terms of performance for action recognition. However, in contrast to appearance features, movements are not specific to fixed durations or standardised sets of motions. Actions that are semantically similar may be visually very different, with the reverse of this also being possible. This means that, although the action class may remain the same, the perception and goals of the actor are bound to be different, thus producing highly varying results. Our works in Chapters 4 and 5 addressed these time-varying features from different executions as well as discovered relationships between the short motions that are performed. Based on this, we believe that subsequent research should aim towards three main directions.

The first direction is to improve the modelling of temporal motion variations. As dataset sizes are growing, to improve upon the large level of diversity of real world data, the representation of complex motion patterns and sequences becomes even more crucial. One notable issue that arises from the task of increasing temporal modelling capabilities of networks is the impact in computational requirements. A key element that needs to be considered is temporal order. To encode temporal information within local patterns, information only from the preceding temporal locations affect the current frame. Causal convolutions [184] have developed this notion as part of their main functionality in which multiple layers with ascending dilation sizes are used to ensure temporal ordering. Similarly, non-local blocks [285] were based on temporal attention to capture the temporal feature dependencies. In Chapter 4 we proposed an alignment of local features based on their importance. The study of local features within the global context of a video has shown to be a promising direction towards utilising feature relevance. An approach that can create spatio-temporal feature descriptors that encode information from preceding temporal locations is therefore favourable. Such approaches are also beneficial for online action recognition and action predictions, as early action predictions are feasible given that the dependency of the local features to the entire sequence is modelled explicitly.

Similarly, this dependency of fixed-sized features to the global motions in videos can be addressed architecturally. The approach in Chapter 5 of sacrificing part of the spatial feature resolution has shown promise. However, this requirement should not come at the cost of reduced spatial modelling capabilities but rather as a balanced approach of addressing temporal variation without a decline in spatial pattern extraction. A direction that could provide such a balancing approach could be through the use of reinforcement learning and evolutionary algorithms. As they aim at creating robust network architectures (e.g., [323]). The reliance to appearance-based or temporally-based can be learned through optimisation within different parts of the network. The search space parameterisation can be also extended to the temporal kernel receptive fields or even the number of different

temporal kernels that are used and how they are aggregated. As the temporal characteristics relating to the visual aspects are explored in a greater extent, we find such approaches promising.

The second research direction is towards the finer-grained labelling of human actions and interactions. A key aspect in the performance of actions is how the actor(s) identify an action. Most of the current practices only consider an action as belonging to a single class. But human behaviour is often more open to subjectivity. Therefore, a less strict separation into classes could be beneficial for their generalisation. Works based on overlapping labels or behaviour hierarchies (e.g., [71, 302]) can facilitate the focus on distinctive patterns at different levels of granularity. The inclusion of semantic meaning of actions (e.g., [49, 223]) has also shown great promise. Complex actions can be categorised to smaller and less complex sub-actions creating finer-grained labels. This can motivate the creation of larger datasets or the re-evaluation of current benchmark datasets.

Actions have been predominantly classified directly based on an input. Works have also considered semantic mid-level representations based on classes (e.g., [134, 219, 126, 257]). These methods show that they can provide some invariance to the features learned and that they relate to a specific class. They can facilitate modelling of actions through spatio-temporal patterns of these mid-level features that are connected to classes. In Chapter 6 we have proposed a way of relating the extracted convolutional features to classes and modelling their importance as part of their activations. This attention to features that are specific to a class can be crucial to distinguishing between subtly different classes. Spatio-temporal feature representations should take into account the correlation of features to classes. The creation of either, a hierarchy of general-to-finer features in layers, or the early prediction of classes across different network depths, appear promising in that respect.

Finally, we believe that the use of qualitative methods apart from accuracy should be adopted by the video understanding community. The use of a single score to determine the capabilities of the model limits the ability to determine the cases of features that are found to be more easy or challenging to classify. In order to understand and be able to address weaknesses of networks, uncovering the internal states of models has potential as an approach. Through the discovery of the patterns and features that are found to be descriptive, an assessment about the model's perception of the world can be made. Addressing skewed or incorrect perceptions is to be prioritised, since by understanding the moment-to-moment internal state, we are able to understand the effect of past states and possible future states that can be expected.

A direct application of using attention can be on the automated analysis of actions and interactions focusing on their detection. Understanding human behaviour can benefit from linking actors and the class that the action is assigned to. We believe that there is great potential on leveraging detection for the understanding of human behaviour when looking for deviations from common practices or anomalies in the sequence of movements performed. For example, sequences A and C in Figure 8.1 show examples of interactions and actions in which their classification would be difficult as actor intent does not (necessarily) correspond to the performance of an action. In addition, using finite definitions of actions without incorporating the relations to people or objects can only describe a finite number of scenarios. Considering sequence B in Figure 8.1, without a standard class specifically for tightroping whilst on a bicycle, the sequence can be easily misclassified. However, if the bicycle is detected and the action of tightroping is also discovered an association between the two can be learned instead. Based on this, a longer-term analysis of the actors and their roles or relations to other actors, as well as knowledge of the social and cultural

Figure 8.1. Ambiguous action and interaction examples. Sequence A shows that there is no interaction between the two main actors in the scene. There is however an intent for a fist-bump. Sequence B shows action ambiguity in terms of the action label to describe the activity performed. tightroping while on a bicycle is not an action that can be (easily) standardised within a dataset. In sequence C, although a golf swing is performed, the actor misses the ball and thus the action is not performed in full.

norms, can help in providing an improved understanding of observed social behaviours. Understanding the intentions of a person can help to analyse what a person is doing rather than focusing on how that is achieved.

We have just scratched the surface when it comes to the automatic understanding of human actions and interactions. This thesis has provided targeted approaches for problems that have been identified, and proposed new promising directions of research to further address the current limitations. We believe that the presented work and the identified research directions are substantial motives for future works in the automatic understanding of human actions and interactions.

Bibliography

[1] URL: http://www.cbsr.ia.ac.cn/english/Action%5C%20Databases%5C%20EN.asp.

[2] Abu-El-Haija, Sami, Kothari, Nisarg, Lee, Joonseok, Natsev, Paul, Toderici, George, Varadarajan, Balakrishnan, and Vijayanarasimhan, Sudheendra. "Youtube-8m: A large-scale video classification benchmark". In: *arXiv preprint arXiv:1609.08675* (2016).

[3] Asadi-Aghbolaghi, Maryam, Clapes, Albert, Bellantonio, Marco, Escalante, Hugo Jair, Ponce-López, Vıéctor, Baró, Xavier, Guyon, Isabelle, Kasaei, Shohreh, and Escalera, Sergio. "A survey on deep learning based approaches for action and gesture recognition in image sequences". In: *Automatic Face & Gesture Recognition (FG)*. IEEE. 2017, pp. 476–483.

[4] Ba, Jimmy Lei, Kiros, Jamie Ryan, and Hinton, Geoffrey E. "Layer normalization". In: *arXiv preprint arXiv:1607.06450* (2016).

[5] Baccouche, Moez, Mamalet, Franck, Wolf, Christian, Garcia, Christophe, and Baskurt, Atilla. "Sequential deep learning for human action recognition". In: *International Workshop on Human Behavior Understanding (HBU)*. Springer. 2011, pp. 29–39.

[6] Bach, Sebastian, Binder, Alexander, Montavon, Grégoire, Klauschen, Frederick, Müller, Klaus-Robert, and Samek, Wojciech. "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation". In: *PloS one* vol. 10, no. 7 (2015), e0130140.

[7] Bagautdinov, Timur, Alahi, Alexandre, Fleuret, François, Fua, Pascal, and Savarese, Silvio. "Social Scene Understanding: End-to-End Multi-Person Action Localization and Collective Activity Recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2. IEEE. 2017.

[8] Barekatain, Mohammadamin, Martıé, Miquel, Shih, Hsueh-Fu, Murray, Samuel, Nakayama, Kotaro, Matsuo, Yutaka, and Prendinger, Helmut. "Okutama-action: An aerial view video dataset for concurrent human action detection". In: *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2017, pp. 28–35.

[9] Bargal, Sarah Adel, Zunino, Andrea, Kim, Donghyun, Zhang, Jianming, Murino, Vittorio, and Sclaroff, Stan. "Excitation backprop for RNNs". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2018, pp. 1440–1449.

[10] Bau, David, Zhou, Bolei, Khosla, Aditya, Oliva, Aude, and Torralba, Antonio. "Network dissection: Quantifying interpretability of deep visual representations". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6541–6549.

[11] Bau, David, Zhu, Jun-Yan, Strobelt, Hendrik, Zhou, Bolei, Tenenbaum, Joshua B, Freeman, William T, and Torralba, Antonio. "Visualizing and understanding generative adversarial networks". In: *arXiv preprint arxiv:1901.09887* (2019).

[12] Baumgartner, Christian F, Koch, Lisa M, Tezcan, Kerem Can, Ang, Jia Xi, and Konukoglu, Ender. "Visual feature attribution using wasserstein gans". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2018, pp. 8309–8319.

[13] Bengio, Yoshua. "Deep learning of representations for unsupervised and transfer learning". In: *International Conference on Machine Learning Workshops (ICML)*. 2012, pp. 17–36.

[14] Bengio, Yoshua, Bergeron, Arnaud, Boulanger–Lewandowski, Nicolas, Breuel, Thomas, Chherawala, Youssouf, Cisse, Moustapha, Erhan, Dumitru, Eustache, Jeremy, Glorot, Xavier, and Muller, Xavier. "Deep learners benefit more from out-of-distribution examples". In: *International Conference on Artificial Intelligence and Statistics (ICAIS)*. 2011, pp. 164–172.

[15] Bilen, Hakan, Fernando, Basura, Gavves, Efstratios, Vedaldi, Andrea, and Gould, Stephen. "Dynamic image networks for action recognition". In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2016, pp. 3034–3042.

[16] Binder, Alexander, Montavon, Grégoire, Lapuschkin, Sebastian, Müller, Klaus-Robert, and Samek, Wojciech. "Layer-wise relevance propagation for neural networks with local renormalization layers". In: *International Conference on Artificial Neural Networks*. Springer. 2016, pp. 63–71.

[17] Bins Filho, José Carlos. "CAVIAR: ontext Aware Vision using Image-based Active Recognition". In: (2004).

[18] Blank, Moshe, Gorelick, Lena, Shechtman, Eli, Irani, Michal, and Basri, Ronen. "Actions as space-time shapes". In: *International Conference on Computer Vision (ICCV'05) Volume 1*. Vol. 2. IEEE. 2005, pp. 1395–1402.

[19] Blunsden, Scott and Fisher, RB. "The BEHAVE video dataset: ground truthed video for multi-person behavior classification". In: *Annals of the BMVA* vol. 4, no. 1-12 (2010), p. 4.

[20] Bourdev, Lubomir, Maji, Subhransu, Brox, Thomas, and Malik, Jitendra. "Detecting People Using Mutually Consistent Poselet Activations". In: *European Conference on Computer Vision (ECCV)*. Springer. 2010, pp. 168–181.

[21] Boyce, William E. and DiPrima, Richard C. *Elementary differential equations and boundary value problems*. Wiley, 1986.

[22] Buades, Antoni, Coll, Bartomeu, and Morel, J-M. "A non-local algorithm for image denoising". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2005, pp. 60–65.

[23] Caba Heilbron, Fabian, Barrios, Wayner, Escorcia, Victor, Mettes, Pascal, Snoek, Cees, Ghanem, Bernard, and Niebles, Juan Carlos. *ActivityNet Large Scale Activity Recognition Challenge*. 2016. URL: http://activity-net.org/challenges/2016/index.html.

[24] Caba Heilbron, Fabian, Carlos Niebles, Juan, and Ghanem, Bernard. "Fast temporal activity proposals for efficient detection of human actions in untrimmed videos". In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2016, pp. 1914–1923.

[25] Caba Heilbron, Fabian, Escorcia, Victor, Ghanem, Bernard, and Carlos Niebles, Juan. "Activitynet: A large-scale video benchmark for human activity understanding". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 961–970.

[26] Cao, Yu, Barrett, Daniel, Barbu, Andrei, Narayanaswamy, Siddharth, Yu, Haonan, Michaux, Aaron, Lin, Yuewei, Dickinson, Sven, Mark Siskind, Jeffrey, and Wang, Song. "Recognize human activities from partially observed videos". In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2013, pp. 2658–2665.

[27] Cao, Yue, Xu, Jiarui, Lin, Stephen, Wei, Fangyun, and Hu, Han. "Gcnet: Non-local networks meet squeeze-excitation networks and beyond". In: *International Conference on Computer Vision Workshops (ICCVW)*. 2019, pp. 1971–1980.

[28] Carreira, Joao, Agrawal, Pulkit, Fragkiadaki, Katerina, and Malik, Jitendra. "Human pose estimation with iterative error feedback". In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2016, pp. 4733–4742.

[29] Carreira, Joao, Noland, Eric, Banki-Horvath, Andras, Hillier, Chloe, and Zisserman, Andrew. "A short note about kinetics-600". In: *arXiv preprint arXiv:1808.01340* (2018).

[30] Carreira, Joao, Noland, Eric, Hillier, Chloe, and Zisserman, Andrew. "A short note on the Kinetics-700 human action dataset". In: *arXiv preprint arXiv:1907.06987* (2019).

[31] Carreira, Joao and Zisserman, Andrew. "Quo vadis, action recognition? A new model and the Kinetics dataset". In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2017, pp. 4724–4733.

[32] Caruana, Rich. "Multitask learning". In: *Learning to learn*. Springer, 1998, pp. 95–133.

[33] Chaquet, Jose M, Carmona, Enrique J, and Fernández-Caballero, Antonio. "A survey of video datasets for human action and activity recognition". In: *Computer Vision and Image Understanding* vol. 117, no. 6 (2013), pp. 633–659.

[34] Chattopadhay, Aditya, Sarkar, Anirban, Howlader, Prantik, and Balasubramanian, Vineeth N. "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks". In: *Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2018, pp. 839–847.

[35] Chen, Liang-Chieh, Yang, Yi, Wang, Jiang, Xu, Wei, and Yuille, Alan L. "Attention to scale: Scale-aware semantic image segmentation". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 3640–3649.

[36] Chen, Long, Zhang, Hanwang, Xiao, Jun, Nie, Liqiang, Shao, Jian, Liu, Wei, and Chua, Tat-Seng. "Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 5659–5667.

[37] Chen, Yunpeng, Fan, Haoqi, Xu, Bing, Yan, Zhicheng, Kalantidis, Yannis, Rohrbach, Marcus, Yan, Shuicheng, and Feng, Jiashi. "Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution". In: *International Conference on Computer Vision (ICCV)*. IEEE. 2019, pp. 3435–3444.

[38] Chen, Yunpeng, Kalantidis, Yannis, Li, Jianshu, Yan, Shuicheng, and Feng, Jiashi. "A^2-Nets: double attention networks". In: *International Conference on Neural Information Processing Systems (NeurIPS)*. 2018, pp. 350–359.

[39] Chen, Yunpeng, Kalantidis, Yannis, Li, Jianshu, Yan, Shuicheng, and Feng, Jiashi. "Multi-Fiber networks for Video Recognition". In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 352–367.

[40] Cho, Kyunghyun, Merriënboer, Bart van, Gulcehre, Caglar, Bahdanau, Dzmitry, Bougares, Fethi, Schwenk, Holger, and Bengio, Yoshua. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1724–1734.

[41] Cho, Nam-Gyu, Park, Se-Ho, Park, Jeong-Seon, Park, Unsang, and Lee, Seong-Whan. "Compositional Interaction Descriptor for Human Interaction Recognition". In: *Neurocomputing* (2017).

[42] Choi, Wongun and Savarese, Silvio. "Understanding collective activities of people from videos". In: *Transactions on Pattern Analysis and Machine Intelligence* vol. 36, no. 6 (2014), pp. 1242–1257.

[43] Chollet, Francois. "Xception: Deep Learning with Depthwise Separable Convolutions". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2017, pp. 1800–1807.

[44] Choutas, Vasileios, Weinzaepfel, Philippe, Revaud, Jérôme, and Schmid, Cordelia. "Potion: Pose motion representation for action recognition". In: *Conference on Computer Vision and Pattern Recognition*. IEEE. 2018, pp. 7024–7033.

[45] Chung, Jihoon, Wuu, Cheng-hsin, Yang, Hsuan-ru, Tai, Yu-Wing, and Tang, Chi-Keung. "HAA500: Human-Centric Atomic Action Dataset with Curated Videos". In: *arXiv preprint arXiv:2009.05224* (2020).

[46] Chung, Junyoung, Gulcehre, Caglar, Cho, KyungHyun, and Bengio, Yoshua. "Empirical evaluation of gated recurrent neural networks on sequence modeling". In: *arXiv preprint arXiv:1412.3555* (2014).

[47] Craparo, Robert M. "Significance level". In: *Encyclopedia of measurement and statistics* vol. 3 (2007), pp. 889–891.

[48] Damen, Dima, Doughty, Hazel, Farinella, Giovanni Maria, Fidler, Sanja, Furnari, Antonino, Kazakos, Evangelos, Moltisanti, Davide, Munro, Jonathan, Perrett, Toby, Price, Will, et al. "Scaling egocentric vision: The epic-kitchens dataset". In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 720–736.

[49] Damen, Dima, Doughty, Hazel, Farinella, Giovanni Maria, Furnari, Antonino, Kazakos, Evangelos, Ma, Jian, Moltisanti, Davide, Munro, Jonathan, Perrett, Toby, Price, Will, et al. "Rescaling egocentric vision". In: *arXiv preprint arXiv:2006.13256* (2020).

[50] Delaitre, Vincent, Laptev, Ivan, and Sivic, Josef. "Recognizing human actions in still images: a study of bag-of-features and part-based representations". In: *British Machine Vision Conference (BMVC)*. (BMVA). 2010.

[51] Deng, Jia, Dong, Wei, Socher, Richard, Li, Li-Jia, Li, Kai, and Fei-Fei, Li. "Imagenet: A large-scale hierarchical image database". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2009, pp. 248–255.

[52] Deng, Zhiwei, Vahdat, Arash, Hu, Hexiang, and Mori, Greg. “Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition”. In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2016, pp. 4772–4781.

[53] Diba, Ali, Fayyaz, Mohsen, Sharma, Vivek, Mahdi Arzani, M, Yousefzadeh, Rahman, Gall, Juergen, and Van Gool, Luc. “Spatio-temporal channel correlation networks for action classification”. In: *European Conference on Computer Vision (ECCV)*. (Springer). 2018, pp. 284–299.

[54] Diba, Ali, Fayyaz, Mohsen, Sharma, Vivek, Paluri, Manohar, Gall, Jürgen, Stiefelhagen, Rainer, and Van Gool, Luc. “Large Scale Holistic Video Understanding”. In: *European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 593–610.

[55] Diba, Ali, Sharma, Vivek, and Van Gool, Luc. “Deep temporal linear encoding networks”. In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2017, pp. 2329–2338.

[56] Dollár, Piotr, Rabaud, Vincent, Cottrell, Garrison, and Belongie, Serge. “Behavior recognition via sparse spatio-temporal features”. In: *International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance VS-PET*. IEEE. 2005, pp. 65–72.

[57] Donahue, Jeffrey, Anne Hendricks, Lisa, Guadarrama, Sergio, Rohrbach, Marcus, Venugopalan, Subhashini, Saenko, Kate, and Darrell, Trevor. “Long-term recurrent convolutional networks for visual recognition and description”. In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2015, pp. 2625–2634.

[58] Dwibedi, Debidatta, Aytar, Yusuf, Tompson, Jonathan, Sermanet, Pierre, and Zisserman, Andrew. “Temporal cycle-consistency learning”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2019, pp. 1801–1810.

[59] Edwards, Allen L. “Note on the “correction for continuity” in testing the significance of the difference between correlated proportions”. In: *Psychometrika* vol. 13, no. 3 (1948), pp. 185–187.

[60] Edwards, Michael, Deng, Jingjing, and Xie, Xianghua. “From pose to activity: Surveying datasets and introducing CONVERSE”. In: *Computer Vision and Image Understanding* vol. 144 (2016), pp. 73–105.

[61] Erhan, Dumitru, Bengio, Yoshua, Courville, Aaron, and Vincent, Pascal. “Visualizing higher-layer features of a deep network”. In: (2009).

[62] Fan, Quanfu, Chen, Chun-Fu, Kuehne, Hilde, Pistoia, Marco, and Cox, David. “More is less: Learning efficient video representations by big-little network and depthwise temporal aggregation”. In: (2019).

[63] Feichtenhofer, Christoph. “X3D: Expanding Architectures for Efficient Video Recognition”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2020, pp. 203–213.

[64] Feichtenhofer, Christoph, Fan, Haoqi, Malik, Jitendra, and He, Kaiming. “SlowFast networks for video recognition”. In: *International Conference on Computer Vision (ICCV)*. IEEE. 2019, pp. 6202–6211.

[65] Feichtenhofer, Christoph, Pinz, Axel, and Wildes, Richard. “Spatiotemporal residual networks for video action recognition”. In: *International Conference on Neural Information Processing Systems (NIPS)*. 2016, pp. 3468–3476.

[66] Felzenszwalb, Pedro F, Girshick, Ross B, McAllester, David, and Ramanan, Deva. "Object detection with discriminatively trained part-based models". In: *Transactions on Pattern Analysis and Machine Intelligence* vol. 32, no. 9 (2010), pp. 1627–1645.

[67] Fisher, Ronald Aylmer. "Statistical methods for research workers". In: *Breakthroughs in statistics*. Springer, 1992, pp. 66–70.

[68] Fong, Ruth, Patrick, Mandela, and Vedaldi, Andrea. "Understanding deep networks via extremal perturbations and smooth masks". In: *International Conference on Computer Vision (ICCV)*. IEEE. 2019, pp. 2950–2958.

[69] Fong, Ruth C and Vedaldi, Andrea. "Interpretable explanations of black boxes by meaningful perturbation". In: *International Conference on Computer Vision (ICCV)*. 2017, pp. 3429–3437.

[70] Fouhey, David F., Kuo, Weicheng, Efros, Alexei A., and Malik, Jitendra. "From Lifestyle VLOGs to Everyday Interactions". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[71] Frosst, Nicholas and Hinton, Geoffrey. "Distilling a Neural Network Into a Soft Decision Tree". In: *arXiv preprint arXiv:1711.09784* (2017).

[72] Gao, Chenqiang, Yang, Luyu, Du, Yinhe, Feng, Zeming, and Liu, Jiang. "From constrained to unconstrained datasets: an evaluation of local action descriptors and fusion strategies for interaction recognition". In: *World Wide Web* vol. 19, no. 2 (2016), pp. 265–276.

[73] Garcia, Nuno, Morerio, Pietro, and Murino, Vittorio. "Modality Distillation with Multiple Stream Networks for Action Recognition". In: *European Conference on Computer Vision (ECCV)*. (Springer). 2018, pp. 106–121.

[74] Gemeren, Coert van, Poppe, Ronald, and Veltkamp, Remco C. "Spatio-temporal detection of fine-grained dyadic human interactions". In: *International Workshop on Human Behavior Understanding (HBU)*. Springer. 2016, pp. 116–133.

[75] Gemeren, Coert van, Tan, Robby T, Poppe, Ronald, and Veltkamp, Remco C. "Dyadic interaction detection from pose and flow". In: *International Workshop on Human Behavior Understanding (HBU)*. Springer. 2014, pp. 101–115.

[76] Gers, Felix A and Schmidhuber, Jürgen. "Recurrent nets that time and count". In: *International Joint Conference on Neural Networks (IJCNN)*. Vol. 3. IEEE. 2000, pp. 189–194.

[77] Ghadiyaram, Deepti, Tran, Du, and Mahajan, Dhruv. "Large-Scale Weakly-Supervised Pre-Training for Video Action Recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2019.

[78] Gilpin, Leilani H, Bau, David, Yuan, Ben Z, Bajwa, Ayesha, Specter, Michael, and Kagal, Lalana. "Explaining explanations: An overview of interpretability of machine learning". In: *International Conference on data science and advanced analytics (DSAA)*. IEEE. 2018, pp. 80–89.

[79] Girdhar, Rohit, Carreira, Joao, Doersch, Carl, and Zisserman, Andrew. "Video Action Transformer Network". In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2019.

[80] Girdhar, Rohit and Ramanan, Deva. "Attentional pooling for action recognition". In: *International Conference on Neural Information Processing Systems (NIPS)*. 2017, pp. 33–44.

[81] Gitman, Igor and Ginsburg, Boris. “Comparison of batch normalization and weight normalization algorithms for the large-scale image classification”. In: *arXiv preprint arXiv:1709.08145* (2017).

[82] Gkalelis, Nikolaos, Kim, Hansung, Hilton, Adrian, Nikolaidis, Nikos, and Pitas, Ioannis. “The i3dpost multi-view and 3d human action/interaction database”. In: *Conference for Visual Media Production*. IEEE. 2009, pp. 159–168.

[83] Gkioxari, Georgia, Girshick, Ross, and Malik, Jitendra. “Contextual action recognition with R* CNN”. In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2015, pp. 1080–1088.

[84] Goldberger, Jacob, Hinton, Geoffrey E, Roweis, Sam T, and Salakhutdinov, Russ R. “Neighbourhood components analysis”. In: *International Conference on Neural Information Processing Systems (NIPS)*. 2005, pp. 513–520.

[85] Goodfellow, Ian J, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. “Generative adversarial networks”. In: (2014).

[86] Goyal, Priya, Dollár, Piotr, Girshick, Ross, Noordhuis, Pieter, Wesolowski, Lukasz, Kyrola, Aapo, Tulloch, Andrew, Jia, Yangqing, and He, Kaiming. “Accurate, large minibatch SGD: training imagenet in 1 hour”. In: *arXiv preprint arXiv:1706.02677* (2017).

[87] Goyal, Raghav, Ebrahimi Kahou, Samira, Michalski, Vincent, Materzynska, Joanna, Westphal, Susanne, Kim, Heuna, Haenel, Valentin, Fruend, Ingo, Yianilos, Peter, Mueller-Freitag, Moritz, et al. “The" something something" video database for learning and evaluating visual common sense”. In: *International Conference on Computer Vision (ICCV)*. 2017, pp. 5842–5850.

[88] Gu, Chunhui, Sun, Chen, Ross, David A, Vondrick, Carl, Pantofaru, Caroline, Li, Yeqing, Vijayanarasimhan, Sudheendra, Toderici, George, Ricco, Susanna, Sukthankar, Rahul, et al. “Ava: A video dataset of spatio-temporally localized atomic visual actions”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 6047–6056.

[89] Hara, Kensho, Kataoka, Hirokatsu, and Satoh, Yutaka. “Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet?” In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2018, pp. 18–22.

[90] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. “Deep residual learning for image recognition”. In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2016, pp. 770–778.

[91] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification”. In: *International Conference on Computer Vision (ICCV)*. IEEE. 2015, pp. 1026–1034.

[92] Herath, Samitha, Harandi, Mehrtash, and Porikli, Fatih. “Going deeper into action recognition: A survey”. In: *Image and vision computing* vol. 60 (2017), pp. 4–21.

[93] Hiley, Liam, Preece, Alun, Hicks, Yulia, Chakraborty, Supriyo, Gurram, Prudhvi, and Tomsett, Richard. “Explaining motion relevance for activity recognition in video deep learning models”. In: *arXiv preprint arXiv:2003.14285* (2020).

[94] Hiley, Liam, Preece, Alun, Hicks, Yulia, Marshall, David, and Taylor, Harrison. "Discriminating spatial and temporal relevance in deep Taylor decompositions for explainable activity recognition". In: (2019).

[95] Hoai, Minh and Zisserman, Andrew. "Talking heads: Detecting humans and recognizing their interactions". In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2014, pp. 875–882.

[96] Hochreiter, Sepp and Schmidhuber, Jürgen. "Long short-term memory". In: *Neural computation* vol. 9, no. 8 (1997), pp. 1735–1780.

[97] Holzinger, Andreas, Langs, Georg, Denk, Helmut, Zatloukal, Kurt, and Müller, Heimo. "Causability and explainability of artificial intelligence in medicine". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* vol. 9, no. 4 (2019), e1312.

[98] Howard, Andrew G, Zhu, Menglong, Chen, Bo, Kalenichenko, Dmitry, Wang, Weijun, Weyand, Tobias, Andreetto, Marco, and Adam, Hartwig. "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861* (2017).

[99] Hu, Jie, Shen, Li, Albanie, Samuel, Sun, Gang, and Vedaldi, Andrea. "Gather-excite: Exploiting feature context in convolutional neural networks". In: *International Conference on Neural Information Processing Systems (NeurIPS)*. 2018, pp. 9401–9411.

[100] Hu, Jie, Shen, Li, and Sun, Gang. "Squeeze-and-excitation networks". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2018, pp. 7132–7141.

[101] Hussein, Noureldien, Gavves, Efstratios, and Smeulders, Arnold WM. "Timeception for complex action recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 254–263.

[102] Hutchinson, Matthew and Gadepally, Vijay. "Video Action Understanding: A Tutorial". In: *arXiv preprint arXiv:2010.06647* (2020).

[103] Hwang, Hochul, Jang, Cheongjae, Park, Geonwoo, Cho, Junghyun, and Kim, Ig-Jae. "ElderSim: A Synthetic Data Generation Platform for Human Action Recognition in Eldercare Applications". In: *arXiv preprint arXiv:2010.14742* (2020).

[104] Idrees, Haroon, Zamir, Amir R, Jiang, Yu-Gang, Gorban, Alex, Laptev, Ivan, Sukthankar, Rahul, and Shah, Mubarak. "The THUMOS challenge on action recognition for videos "in the wild"". In: *Computer Vision and Image Understanding* vol. 155 (2017), pp. 1–23.

[105] Ioffe, Sergey and Szegedy, Christian. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International Conference on Machine Learning (ICML)*. 2015, pp. 448–456.

[106] Jaderberg, Max, Simonyan, Karen, Zisserman, Andrew, and Kavukcuoglu, Koray. "Spatial transformer networks". In: *International Conference on Neural Information Processing Systems (NIPS)*. 2015, pp. 2017–2025.

[107] Jarrett, Kevin, Kavukcuoglu, Koray, Ranzato, Marc'Aurelio, and LeCun, Yann. "What is the best multi-stage architecture for object recognition?" In: *International Conference on Computer Vision (ICCV)*. IEEE. 2009, pp. 2146–2153.

[108] Ji, Shuiwang, Xu, Wei, Yang, Ming, and Yu, Kai. "3D convolutional neural networks for human action recognition". In: *Transactions on Pattern Analysis and Machine Intelligence* vol. 35, no. 1 (2013), pp. 221–231.

[109] Ji, Xiaofei, Wang, Changhui, and Ju, Zhaojie. "A new framework of human interaction recognition based on multiple stage probability fusion". In: *Applied Sciences* vol. 7, no. 6 (2017), p. 567.

[110] Ji, Xiaofei, Wang, Changhui, Zuo, Xinmeng, and Wang, Yangyang. "Multiple Feature Voting based Human Interaction Recognition". In: *International Journal of Signal Processing, Image Processing and Pattern Recognition* vol. 9, no. 1 (2016), pp. 323–334.

[111] Jiang, Boyuan, Wang, MengMeng, Gan, Weihao, Wu, Wei, and Yan, Junjie. "Stm: Spatiotemporal and motion encoding for action recognition". In: *International Conference on Computer Vision (ICCV)*. IEEE. 2019, pp. 2000–2009.

[112] Jiang, Yu-Gang, Liu, Jingen, Roshan Zamir, Amir, Laptev, Ivan, Piccardi, Massimo, Shah, Mumbarak, and Sukthankar, Rahul. *THUMOS challenge: Action recognition with a large number of classes, 2013.* 2013. URL: https://www.crcv.ucf.edu/ICCV13-Action-Workshop/.

[113] Jiang, Yu-Gang, Liu, Jingen, Roshan Zamir, Amir, Toderici, George, Laptev, Ivan, Shah, Mumbarak, and Sukthankar, Rahul. *THUMOS challenge: Action recognition with a large number of classes, 2014.* 2014. URL: http://crcv.ucf.edu/THUMOS14/.

[114] Jiang, Yu-Gang, Ye, Guangnan, Chang, Shih-Fu, Ellis, Daniel, and Loui, Alexander C. "Consumer video understanding: A benchmark database and an evaluation of human and machine performance". In: *International Conference on Multimedia Retrieval (ICMR)*. ACM. 2011, pp. 1–8.

[115] Jimenez-Rodriguez, P, Maghsoudi, S, and Munoz-Fernandez, Gustavo A. "Convolution functions that are nowhere differentiable". In: *Journal of Mathematical Analysis and Applications* vol. 413, no. 2 (2014), pp. 609–615.

[116] Kahatapitiya, Kumara and Ryoo, Michael S. "Coarse-Fine Networks for Temporal Activity Detection in Videos". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2021.

[117] Karpathy, Andrej, Johnson, Justin, and Fei-Fei, Li. "Visualizing and understanding recurrent networks". In: *Internation Conference of Learning Representations Workshops (ICLRW)* (2016).

[118] Karpathy, Andrej, Toderici, George, Shetty, Sanketh, Leung, Thomas, Sukthankar, Rahul, and Fei-Fei, Li. "Large-scale video classification with convolutional neural networks". In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2014, pp. 1725–1732.

[119] Kataoka, Hirokatsu, Wakamiya, Tenga, Hara, Kensho, and Satoh, Yutaka. "Would Mega-scale Datasets Further Enhance Spatiotemporal 3D CNNs?" In: *arXiv preprint arXiv:2004.04968* (2020).

[120] Kay, Will, Carreira, Joao, Simonyan, Karen, Zhang, Brian, Hillier, Chloe, Vijayanarasimhan, Sudheendra, Viola, Fabio, Green, Tim, Back, Trevor, and Natsev, Paul. "The Kinetics human action video dataset". In: *arXiv preprint arXiv:1705.06950* (2017).

[121] Khodabandeh, Mehran, Reza Vaezi Joze, Hamid, Zharkov, Ilya, and Pradeep, Vivek. "Diy human action dataset generation". In: *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE. 2018, pp. 1448–1458.

[122] Khodabandeh, Mehran, Vahdat, Arash, Zhou, Guang-Tong, Hajimirsadeghi, Hossein, Javan Roshtkhari, Mehrsan, Mori, Greg, and Se, Stephen. "Discovering human interactions in videos with limited data labeling". In: *Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE. 2015, pp. 9–18.

[123] Kingma, Diederik P and Welling, Max. "Auto-encoding variational bayes". In: (2014).

[124] Kliper-Gross, Orit, Hassner, Tal, and Wolf, Lior. "The action similarity labeling challenge". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol. 34, no. 3 (2011), pp. 615–621.

[125] Kondratyuk, Dan, Yuan, Liangzhe, Li, Yandong, Zhang, Li, Tan, Mingxing, Brown, Matthew, and Gong, Boqing. "MoViNets: Mobile Video Networks for Efficient Video Recognition". In: *arXiv preprint arXiv:2103.11511* (2021).

[126] Kong, Yu, Jia, Yunde, and Fu, Yun. "Learning human interaction by interactive phrases". In: *European Conference on Computer Vision (ECCV)*. Springer. 2012, pp. 300–313.

[127] Kong, Yu, Kit, Dmitry, and Fu, Yun. "A discriminative model with multiple temporal scales for action prediction". In: *European Conference on Computer Vision (ECCV)*. Springer. 2014, pp. 596–611.

[128] Koohzadi, Maryam and Charkari, Nasrollah Moghadam. "Survey on deep learning methods in human action recognition". In: *IET Computer Vision* vol. 11, no. 8 (2017), pp. 623–632.

[129] Koppula, Hema Swetha, Gupta, Rudhir, and Saxena, Ashutosh. "Learning human activities and object affordances from rgb-d videos". In: *The International Journal of Robotics Research* vol. 32, no. 8 (2013), pp. 951–970.

[130] Kramer, Mark A. "Nonlinear principal component analysis using autoassociative neural networks". In: *AIChE journal* vol. 37, no. 2 (1991), pp. 233–243.

[131] Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. "Imagenet classification with deep convolutional neural networks". In: *International Conference on Neural Information Processing Systems (NIPS)* vol. 25 (2012), pp. 1097–1105.

[132] Krogh, Anders and Hertz, John A. "Generalization in a linear perceptron in the presence of noise". In: *Journal of Physics A: Mathematical and General* vol. 25, no. 5 (1992), p. 1135.

[133] Kuehne, Hildegard, Jhuang, Hueihan, Garrote, Estıébaliz, Poggio, Tomaso, and Serre, Thomas. "HMDB: A large video database for human motion recognition". In: *International Conference on Computer Vision (ICCV)*. IEEE. 2011, pp. 2556–2563.

[134] Lan, Tian, Wang, Yang, Yang, Weilong, Robinovitch, Stephen N., and Mori, Greg. "Discriminative latent models for recognizing contextual group activities". In: *Transactions on Pattern Analysis and Machine Intelligence* vol. 34, no. 8 (2012), pp. 1549–1562.

[135] Laptev, Ivan. "On space-time interest points". In: *International journal of computer vision* vol. 64, no. 2-3 (2005), pp. 107–123.

[136] Laptev, Ivan, Marszalek, Marcin, Schmid, Cordelia, and Rozenfeld, Benjamin. "Learning realistic human actions from movies". In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2008, pp. 1–8.

[137] Laptev, Ivan and Pérez, Patrick. "Retrieving actions in movies". In: *International Conference on Computer Vision (ICCV)*. IEEE. 2007, pp. 1–8.

[138] Lathuilière, Stéphane, Evangelidis, Georgios, and Horaud, Radu. "Recognition of Group Activities in Videos Based on Single-and Two-Person Descriptors". In: *Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2017, pp. 217–225.

[139] LeCun, Yann, Bottou, Léon, Bengio, Yoshua, and Haffner, Patrick. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* vol. 86, no. 11 (1998), pp. 2278–2324.

[140] Li, Chao, Zhong, Qiaoyong, Xie, Di, and Pu, Shiliang. "Collaborative Spatiotemporal Feature Learning for Video Action Recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2019, pp. 7872–7881.

[141] Li, Wanqing, Zhang, Zhengyou, and Liu, Zicheng. "Action recognition based on a bag of 3D points". In: *Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE. 2010, pp. 9–14.

[142] Li, Wenbin and Fritz, Mario. "Recognition of ongoing complex activities by sequence prediction over a hierarchical label space". In: *Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2016, pp. 1–9.

[143] Li, Wenbo, Wen, Longyin, Chang, Ming-Ching, Nam Lim, Ser, and Lyu, Siwei. "Adaptive RNN Tree for Large-Scale Human Action Recognition". In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2017, pp. 1444–1452.

[144] Li, Yan, Ji, Bin, Shi, Xintian, Zhang, Jianguo, Kang, Bin, and Wang, Limin. "Tea: Temporal excitation and aggregation for action recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2020, pp. 909–918.

[145] Li, Yingwei, Li, Yi, and Vasconcelos, Nuno. "Resound: Towards action recognition without representation bias". In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 513–528.

[146] Li, Zhenqiang, Wang, Weimin, Li, Zuoyue, Huang, Yifei, and Sato, Yoichi. "Towards Visually Explaining Video Understanding Networks With Perturbation". In: *Winter Conference on Applications of Computer Vision (WACV)*. 2021, pp. 1120–1129.

[147] Li, Zhenyang, Gavrilyuk, Kirill, Gavves, Efstratios, Jain, Mihir, and Snoek, Cees GM. "VideoLSTM convolves, attends and flows for action recognition". In: *Computer Vision and Image Understanding* vol. 166 (2018), pp. 41–50.

[148] Lin, Ji, Gan, Chuang, and Han, Song. "TSM: Temporal shift module for efficient video understanding". In: *International Conference on Computer Vision (ICCV)*. IEEE. 2019, pp. 7083–7093.

[149] Lin, Tsung-Yi, Maire, Michael, Belongie, Serge, Hays, James, Perona, Pietro, Ramanan, Deva, Dollár, Piotr, and Zitnick, C Lawrence. "Microsoft COCO: Common objects in context". In: *European Conference on Computer Vision (ECCV)*. Springer. 2014, pp. 740–755.

[150] Liu, Jingen, Luo, Jiebo, and Shah, Mubarak. "Recognizing realistic actions from videos "in the wild"". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2009, pp. 1996–2003.

[151] Liu, Zhaoyang, Luo, Donghao, Wang, Yabiao, Wang, Limin, Tai, Ying, Wang, Chengjie, Li, Jilin, Huang, Feiyue, and Lu, Tong. "Teinet: Towards an efficient architecture for video recognition". In: *Conference on Artificial Intelligence (AAAI)*. Vol. 34. 07. 2020, pp. 11669–11676.

[152] Liu, Zhaoyang, Wang, Limin, Wu, Wayne, Qian, Chen, and Lu, Tong. "TAM: Temporal Adaptive Module for Video Recognition". In: *arXiv preprint arXiv:2005.06803* (2020).

[153] Liu, Zhaoyang, Wang, Limin, Wu, Wayne, Qian, Chen, and Lu, Tong. "TAM: Temporal Adaptive Module for Video Recognition". In: (2021).

[154] Long, Xiang, Gan, Chuang, De Melo, Gerard, Wu, Jiajun, Liu, Xiao, and Wen, Shilei. "Attention clusters: Purely attention based local feature integration for video classification". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2018, pp. 7834–7843.

[155] Loshchilov, Ilya and Hutter, Frank. "SGDR: Stochastic gradient descent with warm restarts". In: *International Conference on Learning Representations (ICLR)* (2017).

[156] Lowe, David G. "Object recognition from local scale-invariant features". In: *International Conference on Computer Vision (ICCV)*. Vol. 2. IEEE. 1999, pp. 1150–1157.

[157] Lu, Jiasen, Xu, Ran, and Corso, Jason J. "Human action segmentation with hierarchical supervoxel consistency". In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2015, pp. 3762–3771.

[158] Luo, Chenxu and Yuille, Alan L. "Grouped spatial-temporal aggregation for efficient action recognition". In: *International Conference on Computer Vision (ICCV)*. IEEE. 2019, pp. 5512–5521.

[159] Lyu, Siwei and Simoncelli, Eero P. "Nonlinear image representation using divisive normalization". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2008, pp. 1–8.

[160] Marıén-Jiménez, Manuel J, Yeguas, Enrique, and De La Blanca, Nicolás Pérez. "Exploring STIP-based models for recognizing human interactions in TV videos". In: *Pattern Recognition Letters* vol. 34, no. 15 (2013), pp. 1819–1828.

[161] Marszalek, Marcin, Laptev, Ivan, and Schmid, Cordelia. "Actions in context". In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2009, pp. 2929–2936.

[162] Materzynska, Joanna, Berger, Guillaume, Bax, Ingo, and Memisevic, Roland. "The jester dataset: A large-scale video dataset of human gestures". In: *International Conference on Computer Vision Workshops (ICCVW)*. 2019.

[163] McNemar, Quinn. "Note on the sampling error of the difference between correlated proportions or percentages". In: *Psychometrika* vol. 12, no. 2 (1947), pp. 153–157.

[164] Meng, Yue, Panda, Rameswar, Lin, Chung-Ching, Sattigeri, Prasanna, Karlinsky, Leonid, Saenko, Kate, Oliva, Aude, and Feris, Rogerio. "AdaFuse: Adaptive Temporal Fusion Network for Efficient Action Recognition". In: *International Conference of Learning Representations (ICLR)*. 2021.

[165] Messing, Ross, Pal, Chris, and Kautz, Henry. "Activity recognition using the velocity histories of tracked keypoints". In: *International Conference on Computer Vision (ICCV)*. IEEE. 2009, pp. 104–111.

[166] Mettes, Pascal and Snoek, Cees GM. "Spatial-Aware Object Embeddings for Zero-Shot Localization and Classification of Actions". In: *International Conference on Computer Vision (ICCV)*. IEEE. 2017.

[167] Miao, Qiguang, Li, Yunan, Ouyang, Wanli, Ma, Zhenxin, Xu, Xin, Shi, Weikang, Cao, Xiaochun, Liu, Zhipeng, Chai, Xiujuan, Liu, Zhuang, et al. "Multimodal Gesture Recognition Based on the ResC3D Network". In: *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE. 2017.

[168] Mohammadi, Sadegh, Kiani, Hamed, Perina, Alessandro, and Murino, Vittorio. "Violence detection in crowded scenes using substantial derivative". In: *Advanced Video and Signal Based Surveillance (AVSS)*. IEEE. 2015, pp. 1–6.

[169] Monfort, Mathew, Andonian, Alex, Zhou, Bolei, Ramakrishnan, Kandan, Bargal, Sarah Adel, Yan, Tom, Brown, Lisa, Fan, Quanfu, Gutfreund, Dan, Vondrick, Carl, et al. "Moments in time dataset: One million videos for event understanding". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol. 42, no. 2 (2019), pp. 502–508.

[170] Montavon, Grégoire, Lapuschkin, Sebastian, Binder, Alexander, Samek, Wojciech, and Müller, Klaus-Robert. "Explaining nonlinear classification decisions with deep taylor decomposition". In: *Pattern Recognition* vol. 65 (2017), pp. 211–222.

[171] Mordvintsev, Alexander, Olah, Christopher, and Tyka, Mike. *Inceptionism: Going Deeper into Neural Networks*. 2015. URL: https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html.

[172] Mousavi, Hossein, Mohammadi, Sadegh, Perina, Alessandro, Chellali, Ryad, and Murino, Vittorio. "Analyzing tracklets for the detection of abnormal crowd behavior". In: *Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2015, pp. 148–155.

[173] Naidu, Rakshit, Ghosh, Ankita, Maurya, Yash, Kundu, Soumya Snigdha, et al. "IS-CAM: Integrated Score-CAM for axiomatic-based explanations". In: *arXiv preprint arXiv:2010.03023* (2020).

[174] Naidu, Rakshit and Michael, Joy. "SS-CAM: Smoothed Score-CAM for sharper visual feature localization". In: *arXiv preprint arXiv:2006.14255* (2020).

[175] Nguyen, Anh, Dosovitskiy, Alexey, Yosinski, Jason, Brox, Thomas, and Clune, Jeff. "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks". In: *International Conference on Neural Information Processing Systems (NeurIPS)*. 2016, pp. 3395–3403.

[176] Nguyen, Anh, Yosinski, Jason, and Clune, Jeff. "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images". In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2015, pp. 427–436.

[177] Nguyen, Anh, Yosinski, Jason, and Clune, Jeff. "Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks". In: *International Conference in Machine Learning Workshops (ICMLW)* (2016).

[178] Niebles, Juan Carlos, Chen, Chih-Wei, and Fei-Fei, Li. "Modeling temporal structure of decomposable motion segments for activity classification". In: *European conference on computer vision (ECCV)*. Springer. 2010, pp. 392–405.

[179] Niebles, Juan Carlos and Fei-Fei, Li. "A hierarchical model of shape and appearance for human action classification". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2007, pp. 1–8.

[180] Niebles, Juan Carlos, Wang, Hongcheng, and Fei-Fei, Li. "Unsupervised learning of human action categories using spatial-temporal words". In: *International journal of computer vision* vol. 79, no. 3 (2008), pp. 299–318.

[181] Oikonomopoulos, Antonios, Patras, Ioannis, and Pantic, Maja. "Spatiotemporal salient points for visual recognition of human actions". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* vol. 36, no. 3 (2006), pp. 710–719.

[182] Olah, Chris, Satyanarayan, Arvind, Johnson, Ian, Carter, Shan, Schubert, Ludwig, Ye, Katherine, and Mordvintsev, Alexander. "The building blocks of interpretability". In: *Distill* vol. 3, no. 3 (2018), e10.

[183] Oneata, Dan, Verbeek, Jakob, and Schmid, Cordelia. "Action and event recognition with Fisher vectors on a compact feature set". In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2013, pp. 1817–1824.

[184] Oord, Aaron van den, Dieleman, Sander, Zen, Heiga, Simonyan, Karen, Vinyals, Oriol, Graves, Alex, Kalchbrenner, Nal, Senior, Andrew, and Kavukcuoglu, Koray. "Wavenet: A generative model for raw audio". In: *arXiv preprint arXiv:1609.03499* (2016).

[185] Pan, Sinno Jialin and Yang, Qiang. "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* vol. 22, no. 10 (2010), pp. 1345–1359.

[186] Park, Dong Huk, Hendricks, Lisa Anne, Akata, Zeynep, Rohrbach, Anna, Schiele, Bernt, Darrell, Trevor, and Rohrbach, Marcus. "Multimodal explanations: Justifying decisions and pointing to the evidence". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 8779–8788.

[187] Park, Eunbyung, Han, Xufeng, Berg, Tamara L, and Berg, Alexander C. "Combining multiple sources of knowledge in deep CNNs for action recognition". In: *Applications of Computer Vision (WACV)*. IEEE. 2016, pp. 1–8.

[188] Park, Jongchan, Woo, Sanghyun, Lee, Joon-Young, and Kweon, In-So. "BAM: Bottleneck Attention Module". In: *British Machine Vision Conference (BMVC)*. British Machine Vision Association (BMVA). 2018.

[189] Paszke, Adam, Gross, Sam, Massa, Francisco, Lerer, Adam, Bradbury, James, Chanan, Gregory, Killeen, Trevor, Lin, Zeming, Gimelshein, Natalia, Antiga, Luca, et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems (NeurIPS)* vol. 32 (2019), pp. 8026–8037.

[190] Patron-Perez, Alonso, Marszalek, Marcin, Reid, Ian, and Zisserman, Andrew. "Structured learning of human interactions in TV shows". In: *Transactions on Pattern Analysis and Machine Intelligence* vol. 34, no. 12 (2012), pp. 2441–2453.

[191] Patron-Perez, Alonso, Marszalek, Marcin, Zisserman, Andrew, and Reid, Ian D. "High Five: Recognising human interactions in TV shows". In: *British Machine Vision Conference (BMVC)*. Vol. 1. (BMVA). 2010, p. 2.

[192] Peng, Wei, Hong, Xiaopeng, and Zhao, Guoying. “Video action recognition via neural architecture searching”. In: *International Conference on Image Processing (ICIP)*. IEEE. 2019, pp. 11–15.

[193] Petsiuk, Vitali, Das, Abir, and Saenko, Kate. “RISE: Randomized Input Sampling for Explanation of Black-box Models”. In: (2018).

[194] Pham, Hieu, Guan, Melody Y, Zoph, Barret, Le, Quoc V, and Dean, Jeff. “Efficient Neural Architecture Search via Parameter Sharing”. In: *arXiv preprint arXiv:1802.03268* (2018).

[195] Piergiovanni, AJ, Angelova, Anelia, Toshev, Alexander, and Ryoo, Michael S. “Evolving space-time neural architectures for videos”. In: *International Conference on Computer Vision (ICCV)*. 2019, pp. 1793–1802.

[196] Piergiovanni, AJ and Ryoo, Michael. “AViD Dataset: Anonymized Videos from Diverse Countries”. In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 16711–16721.

[197] Poppe, Ronald. “A survey on vision-based human action recognition”. In: *Image and vision computing* vol. 28, no. 6 (2010), pp. 976–990.

[198] Prabhakar, Karthir and Rehg, James M. “Categorizing turn-taking interactions”. In: *European Conference on Computer Vision (ECCV)*. Springer. 2012, pp. 383–396.

[199] Price, Will and Damen, Dima. “Play Fair: Frame Contributions in Video Models”. In: *Asian Conference on Computer Vision (ACCV)*. 2020.

[200] Qian, Ning. “On the momentum term in gradient descent learning algorithms”. In: *Neural networks* vol. 12, no. 1 (1999), pp. 145–151.

[201] Qiu, Zhaofan, Yao, Ting, and Mei, Tao. “Learning spatio-temporal representation with pseudo-3D residual networks”. In: *International Conference on Computer Vision (ICCV)*. IEEE. 2017, pp. 5534–5542.

[202] Qiu, Zhaofan, Yao, Ting, Ngo, Chong-Wah, Tian, Xinmei, and Mei, Tao. “Learning spatio-temporal representation with local and global diffusion”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2019, pp. 12056–12065.

[203] Radosavovic, Ilija, Kosaraju, Raj Prateek, Girshick, Ross, He, Kaiming, and Dollár, Piotr. “Designing network design spaces”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2020, pp. 10428–10436.

[204] Ramaswamy, Harish Guruprasad et al. “Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization”. In: *Winter Conference on Applications of Computer Vision (WACV)*. 2020, pp. 983–991.

[205] Raptis, Michalis and Sigal, Leonid. “Poselet key-framing: A model for human activity recognition”. In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2013, pp. 2650–2657.

[206] Reddy, Kishore K and Shah, Mubarak. “Recognizing 50 human action categories of web videos”. In: *Machine Vision and Applications* vol. 24, no. 5 (2013), pp. 971–981.

[207] Ribeiro, Marco Tulio, Singh, Sameer, and Guestrin, Carlos. “" Why should i trust you?" Explaining the predictions of any classifier”. In: *ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1135–1144.

[208] Rifai, Salah, Glorot, Xavier, Bengio, Yoshua, and Vincent, Pascal. "Adding noise to the input of a model trained with a regularized objective". In: *arXiv preprint arXiv:1104.3250* (2011).

[209] Rodriguez, Mikel D, Ahmed, Javed, and Shah, Mubarak. "Action MATCH a Spatio-temporal maximum average correlation height filter for action recognition". In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2008, pp. 1–8.

[210] Roweis, Sam T and Saul, Lawrence K. "Nonlinear dimensionality reduction by locally linear embedding". In: *science* vol. 290, no. 5500 (2000), pp. 2323–2326.

[211] Rumelhart, David E, Hinton, Geoffrey E, and Williams, Ronald J. *Learning internal representations by error propagation*. Tech. rep. California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[212] Ryoo, Michael S and Aggarwal, Jake K. "Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities". In: *International Conference on Computer Vision (ICCV)*. IEEE. 2009, pp. 1593–1600.

[213] Ryoo, Michael S, Piergiovanni, AJ, Tan, Mingxing, and Angelova, Anelia. "Assemblenet: Searching for multi-stream neural connectivity in video architectures". In: *arXiv preprint arXiv:1905.13209* (2019).

[214] Ryoo, MS and Aggarwal, JK. "Stochastic representation and recognition of high-level group activities". In: *International journal of computer Vision* vol. 93, no. 2 (2011), pp. 183–200.

[215] Sadanand, Sreemananth and Corso, Jason J. "Action bank: A high-level representation of activity in video". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2012, pp. 1234–1241.

[216] Salimans, Tim and Kingma, Diederik P. "Weight normalization: a simple reparameterization to accelerate training of deep neural networks". In: *International Conference on Neural Information Processing Systems (NIPS)*. 2016, pp. 901–909.

[217] Santurkar, Shibani, Tsipras, Dimitris, Ilyas, Andrew, and Madry, Aleksander. "How Does Batch Normalization Help Optimization?" In: *International Conference on Neural Information Processing Systems (NeurIPS)*, no. 31 (2018).

[218] Schuldt, Christian, Laptev, Ivan, and Caputo, Barbara. "Recognizing human actions: A local SVM approach". In: *International Conference on Pattern Recognition (ICPR)*. Vol. 3. IEEE. 2004, pp. 32–36.

[219] Sefidgar, Yasaman S, Vahdat, Arash, Se, Stephen, and Mori, Greg. "Discriminative key-component models for interaction detection and recognition". In: *Computer Vision and Image Understanding* vol. 135 (2015), pp. 16–30.

[220] Selvaraju, Ramprasaath R, Cogswell, Michael, Das, Abhishek, Vedantam, Ramakrishna, Parikh, Devi, Batra, Dhruv, et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization." In: *Internation Conference on Computer Vision (ICCV)*. IEEE. 2017, pp. 618–626.

[221] Sener, Fadime and Ikizler-Cinbis, Nazli. "Two-person interaction recognition via spatial multiple instance embedding". In: *Journal of Visual Communication and Image Representation* vol. 32 (2015), pp. 63–73.

[222] Sermanet, Pierre, Eigen, David, Zhang, Xiang, Mathieu, Michael, Fergus, Rob, and LeCun, Yann. "Overfeat: Integrated recognition, localization and detection using convolutional networks". In: *International Conference on Learning Representations (ICLR)*. 2014.

[223] Shao, Dian, Zhao, Yue, Dai, Bo, and Lin, Dahua. "Finegym: A hierarchical video dataset for fine-grained action understanding". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 2616–2625.

[224] Shapley, Lloyd S. "A value for n-person games". In: *Contributions to the Theory of Games* vol. 2, no. 28 (1953), pp. 307–317.

[225] Shariat, Shahriar and Pavlovic, Vladimir. "A new adaptive segmental matching measure for human activity recognition". In: *International Conference on Computer Vision (ICCV)*. IEEE. 2013, pp. 3583–3590.

[226] Sharma, Shikhar, Kiros, Ryan, and Salakhutdinov, Ruslan. "Action Recognition using Visual Attention". In: *International Conference on Learning Representations Workshops (ICLRw)*. 2016.

[227] Shrikumar, Avanti, Greenside, Peyton, and Kundaje, Anshul. "Learning important features through propagating activation differences". In: *International Conference on Machine Learning (ICML)*. PMLR. 2017, pp. 3145–3153.

[228] Sigurdsson, Gunnar A, Varol, Gül, Wang, Xiaolong, Farhadi, Ali, Laptev, Ivan, and Gupta, Abhinav. "Hollywood in homes: Crowdsourcing data collection for activity understanding". In: *European Conference on Computer Vision*. Springer. 2016, pp. 510–526.

[229] Simonyan, Karen and Zisserman, Andrew. "Two-stream convolutional networks for action recognition in videos". In: *International Conference on Neural Information Processing Systems (NIPS)*. 2014, pp. 568–576.

[230] Simonyan, Karen and Zisserman, Andrew. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[231] Sindagi, Vishwanath A and Patel, Vishal M. "Ha-ccn: Hierarchical attention-based crowd counting network". In: *Transactions on Image Processing* vol. 29 (2019), pp. 323–335.

[232] Singh, Bharat, Marks, Tim K, Jones, Michael, Tuzel, Oncel, and Shao, Ming. "A multi-stream bi-directional recurrent neural network for fine-grained action detection". In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2016, pp. 1961–1970.

[233] Singh, Tej and Vishwakarma, Dinesh Kumar. "Video benchmarks of human action datasets: a review". In: *Artificial Intelligence Review* vol. 52, no. 2 (2019), pp. 1107–1154.

[234] Smaira, Lucas, Carreira, João, Noland, Eric, Clancy, Ellen, Wu, Amy, and Zisserman, Andrew. "A Short Note on the Kinetics-700-2020 Human Action Dataset". In: *arXiv preprint arXiv:2010.10864* (2020).

[235] Soomro, Khurram, Zamir, Amir Roshan, and Shah, Mubarak. "UCF101: A dataset of 101 human actions classes from videos in the wild". In: *arXiv preprint arXiv:1212.0402* (2012).

[236] Springenberg, J, Dosovitskiy, Alexey, Brox, Thomas, and Riedmiller, M. "Striving for Simplicity: The All Convolutional Net". In: *International Conference on Learning Representations Workshops (ICLRW)*. 2015.

[237] Srinivasan, Vignesh, Lapuschkin, Sebastian, Hellge, Cornelius, Müller, Klaus-Robert, and Samek, Wojciech. "Interpretable human action recognition in compressed domain". In: *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2017, pp. 1692–1696.

[238] Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* vol. 15, no. 1 (2014), pp. 1929–1958.

[239] Stergiou, Alexandros. "The Mind's Eye: Visualizing Class-Agnostic Features of CNNs". In: *International Conference on Image Processing (ICIP)*. IEEE. 2021.

[240] Stergiou, Alexandros, Kapidis, Georgios, Kalliatakis, Grigorios, Chrysoulas, Christos, Poppe, Ronald, and Veltkamp, Remco. "Class Feature Pyramids for Video Explanation". In: *International Conference on Computer Vision Workshop (ICCVW)*. IEEE. 2019, pp. 4255–4264.

[241] Stergiou, Alexandros, Kapidis, Georgios, Kalliatakis, Grigorios, Chrysoulas, Christos, Veltkamp, Remco, and Poppe, Ronald. "Saliency tubes: Visual explanations for spatio-temporal convolutions". In: *International Conference on Image Processing (ICIP)*. IEEE. 2019, pp. 1830–1834.

[242] Stergiou, Alexandros and Poppe, Ronald. "Analyzing human-human interactions: A survey". In: *Computer Vision and Image Understanding* vol. 188 (2019), p. 102799.

[243] Stergiou, Alexandros and Poppe, Ronald. "Learn to cycle: Time-consistent feature discovery for action recognition". In: *Pattern Recognition Letters* vol. 141 (2021), pp. 1–7.

[244] Stergiou, Alexandros and Poppe, Ronald. "Multi-Temporal Convolutions for Human Action Recognition in Videos". In: *International Joint Conference of Neural Networks (IJCNN)*. IEEE. 2021.

[245] Stergiou, Alexandros and Poppe, Ronald. "Spatio-Temporal FAST 3D Convolutions for Human Action Recognition". In: *International Conference on Machine Learning Applications (ICMLA)*. IEEE. 2019, pp. 1830–1834.

[246] Stergiou, Alexandros, Poppe, Ronald, and Grigorios, Kalliatakis. "Refining activation downsampling with SoftPool". In: *International Conference on Computer Vision (ICCV)*. IEEE. 2021.

[247] Stergiou, Alexandros, Poppe, Ronald, and Veltkamp, Remco C. "Learning Class-Specific Features with Class Regularization for Videos". In: *Applied Sciences* vol. 10, no. 18 (2020), p. 6241.

[248] Sudhakaran, Swathikiran, Escalera, Sergio, and Lanz, Oswald. "Gate-Shift Networks for Video Action Recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2020, pp. 1102–1111.

[249] Sun, Fengdong and Li, Wenhui. "Saliency guided deep network for weakly-supervised image segmentation". In: *Pattern Recognition Letters* vol. 120 (2019), pp. 62–68.

[250] Sundararajan, Mukund, Taly, Ankur, and Yan, Qiqi. "Axiomatic attribution for deep networks". In: *International Conference on Machine Learning (ICML)*. 2017, pp. 3319–3328.

[251] Sung, Jaeyong, Ponce, Colin, Selman, Bart, and Saxena, Ashutosh. "Unstructured human activity detection from rgbd images". In: *International Conference on Robotics and Automation (ICRA)*. IEEE. 2012, pp. 842–849.

[252] Sutskever, Ilya, Martens, James, Dahl, George, and Hinton, Geoffrey. "On the importance of initialization and momentum in deep learning". In: *International Conference on Machine Learning*. PMLR. 2013, pp. 1139–1147.

[253] Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, and Rabinovich, Andrew. "Going deeper with convolutions". In: *Computer Vision and Pattern Recognition, (CVPR)*. IEEE. 2015.

[254] Szegedy, Christian, Vanhoucke, Vincent, Ioffe, Sergey, Shlens, Jon, and Wojna, Zbigniew. "Rethinking the inception architecture for computer vision". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2016, pp. 2818–2826.

[255] Tenenbaum, Joshua B, De Silva, Vin, and Langford, John C. "A global geometric framework for nonlinear dimensionality reduction". In: *science* vol. 290, no. 5500 (2000), pp. 2319–2323.

[256] Tian, Yicong, Sukthankar, Rahul, and Shah, Mubarak. "Spatiotemporal deformable part models for action detection". In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2013, pp. 2642–2649.

[257] Tian, Yonglong, Luo, Ping, Wang, Xiaogang, and Tang, Xiaoou. "Deep learning strong parts for pedestrian detection". In: *International Conference on Computer Vision (ICCV)*. IEEE. 2015, pp. 1904–1912.

[258] Tran, Du, Bourdev, Lubomir, Fergus, Rob, Torresani, Lorenzo, and Paluri, Manohar. "Learning spatiotemporal features with 3D convolutional networks". In: *International Conference on Computer Vision (ICCV)*. IEEE. 2015, pp. 4489–4497.

[259] Tran, Du and Sorokin, Alexander. "Human activity recognition with metric learning". In: *European conference on computer vision*. Springer. 2008, pp. 548–561.

[260] Tran, Du, Wang, Heng, Torresani, Lorenzo, and Feiszli, Matt. "Video Classification With Channel-Separated Convolutional Networks". In: *International Conference on Computer Vision (ICCV)*. IEEE. 2019, pp. 5552–5561.

[261] Tran, Du, Wang, Heng, Torresani, Lorenzo, Ray, Jamie, LeCun, Yann, and Paluri, Manohar. "A Closer Look at Spatiotemporal Convolutions for Action Recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2018, pp. 6450–6459.

[262] Tran, Du and Yuan, Junsong. "Max-margin structured output regression for spatio-temporal action localization". In: *International Conference on Neural Information Processing Systems (NIPS)*. 2012, pp. 350–358.

[263] Tran, Khai N, Bedagkar-Gala, Apurva, Kakadiaris, Ioannis A, and Shah, Shishir K. "Social Cues in Group Formation and Local Interactions for Collective Activity Analysis". In: *International Conference on Computer Vision Theory and Applications (VISAPP)*. 2013, pp. 539–548.

[264] Tran, Khai N, Gala, Apurva, Kakadiaris, Ioannis A, and Shah, Shishir K. "Activity analysis in crowded environments using social cues for group discovery and human interaction modeling". In: *Pattern Recognition Letters* vol. 44 (2014), pp. 49–57.

[265] Tu, Zhigang, Xie, Wei, Qin, Qianqing, Poppe, Ronald, Veltkamp, Remco C, Li, Baoxin, and Yuan, Junsong. "Multi-stream CNN: Learning representations based on human-related regions for action recognition". In: *Pattern Recognition* vol. 79 (2018), pp. 32–43.

[266] Turchini, Francesco, Seidenari, Lorenzo, and Del Bimbo, Alberto. "Understanding and localizing activities from correspondences of clustered trajectories". In: *Computer Vision and Image Understanding* (2016).

[267] Ulyanov, Dmitry, Vedaldi, Andrea, and Lempitsky, Victor. "Instance normalization: The missing ingredient for fast stylization". In: *arXiv preprint arXiv:1607.08022* (2016).

[268] Vallacher, Robin R and Wegner, Daniel M. "Action identification theory". In: *Handbook of theories of social psychology* vol. 1 (2011), pp. 327–349.

[269] Vallacher, Robin R and Wegner, Daniel M. "Levels of personal agency: Individual variation in action identification." In: *Journal of Personality and Social psychology* vol. 57, no. 4 (1989), p. 660.

[270] Van Laarhoven, Twan. "L2 regularization versus batch and weight normalization". In: *arXiv preprint arXiv:1706.05350* (2017).

[271] Varol, Gül, Laptev, Ivan, and Schmid, Cordelia. "Long-term Temporal Convolutions for Action Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).

[272] Vezzani, Roberto and Cucchiara, Rita. "Annotation collection and online performance evaluation for video surveillance: The visor project". In: *International Conference on Advanced Video and Signal Based Surveillance*. IEEE. 2008, pp. 227–234.

[273] Vezzani, Roberto and Cucchiara, Rita. "ViSOR: Video surveillance on-line repository for annotation retrieval". In: *International Conference on Multimedia and Expo*. IEEE. 2008, pp. 1281–1284.

[274] Vrigkas, Michalis, Nikou, Christophoros, and Kakadiaris, Ioannis A. "A review of human activity recognition methods". In: *Frontiers in Robotics and AI* vol. 2 (2015), p. 28.

[275] Wager, Stefan, Wang, Sida, and Liang, Percy. "Dropout training as adaptive regularization". In: *International Conference on Neural Information Processing Systems (NIPS)*. 2013, pp. 351–359.

[276] Wang, Fei, Jiang, Mengqing, Qian, Chen, Yang, Shuo, Li, Cheng, Zhang, Honggang, Wang, Xiaogang, and Tang, Xiaoou. "Residual attention network for image classification". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 3156–3164.

[277] Wang, Feng, Liu, Haijun, and Cheng, Jian. "Visualizing deep neural network by alternately image blurring and deblurring". In: *Neural Networks* vol. 97 (2018), pp. 162–172.

[278] Wang, Haofan, Wang, Zifan, Du, Mengnan, Yang, Fan, Zhang, Zijian, Ding, Sirui, Mardziel, Piotr, and Hu, Xia. "Score-CAM: Score-weighted visual explanations for convolutional neural networks". In: *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2020, pp. 24–25.

[279] Wang, Heng, Kläser, Alexander, Schmid, Cordelia, and Liu, Cheng-Lin. "Dense trajectories and motion boundary descriptors for action recognition". In: *International journal of computer vision* vol. 103, no. 1 (2013), pp. 60–79.

[280] Wang, Heng and Schmid, Cordelia. "Action recognition with improved trajectories". In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2013, pp. 3551–3558.

[281] Wang, Heng, Ullah, Muhammad Muneeb, Klaser, Alexander, Laptev, Ivan, and Schmid, Cordelia. "Evaluation of local spatio-temporal features for action recognition". In: *British Machine Vision Conference (BMVC)*. BMVA Press. 2009.

[282] Wang, Jason, Perez, Luis, et al. "The effectiveness of data augmentation in image classification using deep learning". In: *Convolutional Neural Networks Vis. Recognit* vol. 11 (2017).

[283] Wang, Le, Zang, Jinliang, Zhang, Qilin, Niu, Zhenxing, Hua, Gang, and Zheng, Nanning. "Action Recognition by an Attention-Aware Temporal Weighted Convolutional Neural Network". In: *Sensors* vol. 18, no. 7 (2018), p. 1979.

[284] Wang, Limin, Xiong, Yuanjun, Wang, Zhe, Qiao, Yu, Lin, Dahua, Tang, Xiaoou, and Van Gool, Luc. "Temporal segment networks: Towards good practices for deep action recognition". In: *European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 20–36.

[285] Wang, Xiaolong, Girshick, Ross, Gupta, Abhinav, and He, Kaiming. "Non-local neural networks". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2018, pp. 7794–7803.

[286] Wang, Xiaolong, Jabri, Allan, and Efros, Alexei A. "Learning correspondence from the cycle-consistency of time". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2019, pp. 2566–2576.

[287] Weinland, Daniel, Ronfard, Remi, and Boyer, Edmond. "Free viewpoint action recognition using motion history volumes". In: *Computer vision and image understanding* vol. 104, no. 2-3 (2006), pp. 249–257.

[288] Weinzaepfel, Philippe, Martin, Xavier, and Schmid, Cordelia. "Human action localization with sparse spatial supervision". In: *arXiv preprint arXiv:1605.05197* (2016).

[289] White, John G, Southgate, Eileen, Thomson, J Nichol, and Brenner, Sydney. "The structure of the nervous system of the nematode Caenorhabditis elegans". In: *Philos Trans R Soc Lond B Biol Sci* vol. 314, no. 1165 (1986), pp. 1–340.

[290] Willems, Geert, Tuytelaars, Tinne, and Van Gool, Luc. "An efficient dense and scale-invariant spatio-temporal interest point detector". In: *European Conference on Computer Vision (ECCV)* (2008), pp. 650–663.

[291] Woo, Sanghyun, Park, Jongchan, Lee, Joon-Young, and Kweon, In So. "Cbam: Convolutional block attention module". In: *European conference on computer vision (ECCV)*. 2018, pp. 3–19.

[292] Wu, Chao-Yuan, Girshick, Ross, He, Kaiming, Feichtenhofer, Christoph, and Krähenbühl, Philipp. "A Multigrid Method for Efficiently Training Video Models". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2020, pp. 153–162.

[293] Wu, Chenxia, Zhang, Jiemi, Savarese, Silvio, and Saxena, Ashutosh. "Watch-n-patch: Unsupervised understanding of actions and relations". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 4362–4370.

[294] Wu, Yuxin and He, Kaiming. "Group normalization". In: *European conference on computer vision (ECCV)*. 2018, pp. 3–19.

[295] Wu, Zuxuan, Jiang, Yu-Gang, Wang, Xi, Ye, Hao, and Xue, Xiangyang. "Multi-stream multi-class fusion of deep networks for video classification". In: *Multimedia Conference (ACMM)*. (ACM). 2016, pp. 791–800.

[296] Xia, Lu and Aggarwal, JK. "Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera". In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2013, pp. 2834–2841.

[297] Xie, Saining, Girshick, Ross, Dollár, Piotr, Tu, Zhuowen, and He, Kaiming. "Aggregated residual transformations for deep neural networks". In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2017, pp. 5987–5995.

[298] Yang, Ceyuan, Xu, Yinghao, Shi, Jianping, Dai, Bo, and Zhou, Bolei. "Temporal pyramid network for action recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2020, pp. 591–600.

[299] Yang, Yi, Baker, Simon, Kannan, Anitha, and Ramanan, Deva. "Recognizing proxemics in personal photos". In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2012, pp. 3522–3529.

[300] Yang, Yi and Ramanan, Deva. "Articulated pose estimation with flexible mixtures-of-parts". In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2011, pp. 1385–1392.

[301] Yao, Benjamin Z, Nie, Bruce X, Liu, Zicheng, and Zhu, Song-Chun. "Animated pose templates for modeling and detecting human actions". In: *IEEE transactions on pattern analysis and machine intelligence* vol. 36, no. 3 (2014), pp. 436–452.

[302] Yeung, Serena, Russakovsky, Olga, Jin, Ning, Andriluka, Mykhaylo, Mori, Greg, and Fei-Fei, Li. "Every moment counts: Dense detailed labeling of actions in complex videos". In: *International Journal of Computer Vision* vol. 126, no. 2 (2018), pp. 375–389.

[303] Yin, Yafeng, Yang, Guang, Xu, Jin, and Man, Hong. "Small group human activity recognition". In: *International Conference on Image Processing (ICIP)*. IEEE. 2012, pp. 2709–2712.

[304] Yosinski, Jason, Clune, Jeff, Bengio, Yoshua, and Lipson, Hod. "How transferable are features in deep neural networks?" In: *International Conference on Neural Information Processing Systems (NIPS)*. 2014, pp. 3320–3328.

[305] Yosinski, Jason, Clune, Jeff, Fuchs, Thomas, and Lipson, Hod. "Understanding neural networks through deep visualization". In: *International Conference on Machine Learning Workshops (ICMLW)*. 2015.

[306] Yu, Gang and Yuan, Junsong. “Fast action proposals for human action detection and search”. In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2015, pp. 1302–1311.

[307] Yu, Gang, Yuan, Junsong, and Liu, Zicheng. “Propagative hough voting for human activity recognition”. In: *European Conference on Computer Vision (ECCV)*. Springer. 2012, pp. 693–706.

[308] Zagoruyko, Sergey and Komodakis, Nikos. “Wide residual networks”. In: (2016).

[309] Zeiler, Matthew D and Fergus, Rob. “Visualizing and understanding convolutional networks”. In: *European conference on computer vision (ECCV)*. Springer. 2014, pp. 818–833.

[310] Zeiler, Matthew D and Fergus, Robert. “Stochastic pooling for regularization of deep convolutional neural networks”. In: *International Conference on Learning Representationsm (ICLR)*. 2013.

[311] Zhang, Bo, De Natale, Francesco GB, and Conci, Nicola. “Recognition of social interactions based on feature selection from visual codebooks”. In: *Conference on Image Processing (ICIP)*. IEEE. 2013, pp. 3557–3561.

[312] Zhang, Chiyuan, Bengio, Samy, Hardt, Moritz, Recht, Benjamin, and Vinyals, Oriol. “Understanding deep learning requires rethinking generalization”. In: *International Conference on Learning Representations (ICLR)*. 2017.

[313] Zhang, Jianming, Bargal, Sarah Adel, Lin, Zhe, Brandt, Jonathan, Shen, Xiaohui, and Sclaroff, Stan. “Top-down neural attention by excitation backprop”. In: *International Journal of Computer Vision* vol. 126, no. 10 (2018), pp. 1084–1102.

[314] Zhang, Xiaoning, Wang, Tiantian, Qi, Jinqing, Lu, Huchuan, and Wang, Gang. “Progressive attention guided recurrent network for salient object detection”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 714–722.

[315] Zhang, Yimeng, Liu, Xiaoming, Chang, Ming-Ching, Ge, Weina, and Chen, Tsuhan. “Spatio-temporal phrases for activity recognition”. In: *European Conference on Computer Vision ECCV*. Springer. 2012, pp. 707–721.

[316] Zhao, Hang, Torralba, Antonio, Torresani, Lorenzo, and Yan, Zhicheng. “HACS: Human action clips and segments dataset for recognition and temporal localization”. In: *International Conference on Computer Vision (ICCV)*. IEEE. 2019, pp. 8668–8678.

[317] Zhao, Hengshuang, Zhang, Yi, Liu, Shu, Shi, Jianping, Change Loy, Chen, Lin, Dahua, and Jia, Jiaya. “PSANet: Point-wise spatial attention network for scene parsing”. In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 267–283.

[318] Zhao, Rui, Ali, Haider, and Smagt, Patrick van der. “Two-stream RNN/CNN for action recognition in 3D videos”. In: *International Conference on Intelligent Robots (IROS)*. IEEE. 2017, pp. 4260–4267.

[319] Zhou, Bolei, Khosla, Aditya, Lapedriza, Agata, Oliva, Aude, and Torralba, Antonio. “Learning deep features for discriminative localization”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2921–2929.

[320] Zhou, Bolei, Sun, Yiyou, Bau, David, and Torralba, Antonio. “Interpretable basis decomposition for visual explanation”. In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 119–134.

[321] Zhou, Yizhou, Sun, Xiaoyan, Zha, Zheng-Jun, and Zeng, Wenjun. "MiCT: Mixed 3D/2D Convolutional Tube for Human Action Recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2018, pp. 449–458.

[322] Zintgraf, Luisa M, Cohen, Taco S, Adel, Tameem, and Welling, Max. "Visualizing deep neural network decisions: Prediction difference analysis". In: (2017).

[323] Zoph, Barret and Le, Quoc V. "Neural architecture search with reinforcement learning". In: *Internation Conference on Learning Representations (ICLR)* (2017).

[324] Zoph, Barret, Vasudevan, Vijay, Shlens, Jonathon, and Le, Quoc V. "Learning Transferable Architectures for Scalable Image Recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.