Novel Motion-Aware Strategies for Efficient and Accurate Video Analytics

by

Brennan Gebotys

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2022

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

I was co-author with major contributions to the design, implementation, analysis, and writing of the following two papers which are used in this thesis:

> **Brennan Gebotys**, Alexander Wong, David Clausi, "POOF: Efficient Goalie Pose Annotation using Optical Flow". In: *Proceedings of the 9th International Conference on Sport Sciences Research and Technology Support (icSPORTS)*. 2021.

This paper is incorporated in Chapter 3.

> **Brennan Gebotys**, Alexander Wong, David Clausi, "M2A: Motion Aware Attention for Accurate Video Action Recognition". In: *19th Conference on Robots and Vision (CRV)*. 2022.

This paper is incorporated in Chapter 4.

Abstract

Recent advances in machine learning strategies have led to improved results across a variety of fields. A field that would benefit greatly from improved machine learning strategies is video analytics: the analysis of video data. Two applications of importance include pose estimation, which aims to identify the pose of a person in a video and action recognition, which aims to identify the action that is performed in a video. However, key problems such as how to train a pose estimation model with a small number of annotations and how to design an action recognition model to achieve the highest possible accuracy still remain. This thesis explores how effectively leveraging motion information can enable strategies that can solve both of these problems.

The first problem is that for pose estimation models to achieve a high accuracy, they require a large number of pose annotations, which can be expensive to collect. While a naive approach is to annotate a single frame at a time, researchers have investigated how modifying the model training and generating more annotations can reduce the number of annotations required. However, all these approaches either still include requirements that make annotation collection difficult. This thesis introduces a motion-aware pose annotation strategy called POse annotation using Optical Flow (POOF), which explores how motion information can reduce the number of annotations required without any additional constraints. We show that with only a small number of annotations, utilizing POOF's annotations can achieve a +52% improvement in accuracy compared to training on the small number of annotations. By reducing the number of annotations required, POOF should enable pose estimation models to be more easily applied to many more real-world problems.

The second problem is that because there is such a large number of possible design choices, it is difficult to design an action recognition model's architecture to achieve the highest possible accuracy. While state-of-the-art attention mechanisms are a popular choice and have achieved accurate results, a key shortcoming is that they do not leverage any motion information. Motivated by this, this thesis explores how motion can be leveraged with these attention-based mechanisms by introducing a Motion-Aware Attention mechanism called M2A which explicitly leverages both attention and motion information. We show that incorporating motion mechanisms with attention mechanisms using the proposed M2A mechanism can lead from a +15% to a +26% improvement in top-1 accuracy across different backbone architectures, with only a small increase in computational complexity. By better understanding how motion mechanisms can be both accurate and efficient, M2A should enable action recognition solutions to be applied to real-world problems sooner.

Acknowledgements

First, I want to thank my supervisors, Prof Alexander Wong and Prof David Clausi. Thank you both for all of your support throughout my masters, it was a pleasure to complete my masters under two powerhouse researchers.

Secondly, I want to thank Prof Andrea Scott and Prof Yue Hu. Thank you both for carving out time from your exceedingly busy schedules to read and revise my thesis, it is greatly appreciated.

Lastly, I want to thank my parents and all my VIP friends for all their support. May we all ping pong on.

Dedication

I would like to dedicate this thesis to my parents, thanks for all your support, mom and dad.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recent advances in machine learning have unlocked the ability to solve important real-world problems for millions of people worldwide. This includes a variety of problems related to computer vision, natural language processing, and other fields. A subfield of computer vision that would benefit greatly from improved machine learning strategies is video analytics: the understanding and analysis of video data. A key component of video analytics involves understanding human motion that can lead to a better understanding of human behaviour and solve many problems across many fields including robotics, machine learning (ML) powered personal trainers, augmented reality, and more. For example, in fields such as athletics or construction, it is important that the workers are performing their tasks with the correct posture so that they do not become injured. Using video analytics, a person's posture can be automatically analyzed throughout a video and recommendations of how to improve their posture to reduce their risk of injury can be generated.

To understand human motion, there are multiple sources of information that can be utilized. One source of information is the pose of a person across time. To extract this information, pose estimation models are leveraged, which given a video, extract the location of a person's joints at each frame. Another source of information is understanding what a person is doing in a video. To extract this information action recognition models can be utilized, which given a video, classify what action is being performed in it.

While pose estimation and action recognition models are powerful tools for understanding human motion, there are still important problems that need to be solved first before they can be efficiently applied to a large number of diverse domains.

Figure 1.1: An example of pose estimation for sports analytics. From a broadcast video, individual players are extracted and their poses are estimated. This information can then be analyzed to generate insights and recommendations for the players/team including correcting a player's form.

1.1 Pose Estimation for Video Analytics

The standard definition of pose estimation is the problem such that given an image that includes a person, estimate the locations of a specific set of keypoints. These keypoints can be defined as anything but are typically defined to be the person's joints including the left wrist, the right shoulder, etc. In this thesis, we consider a more video-based definition: **pose estimation is the problem such that given a video that includes a person, estimate the locations of a specific set of keypoints in each frame of the video.**

One example application of pose estimation is sports analytics. Figure 1.1 shows a video analytics pipeline using broadcast videos from the National Hockey League (NHL). In this example, players are extracted from a broadcast video frame and then a pose estimation model is leveraged to identify the pose of each player in the frame. This is then repeated for each frame of the video. This pose information can then be analyzed to generate recommendations for the players. For example, the pose information can be analyzed to extract the angles between specific joints (e.g., the right hip and the right knee) to better understand if a player's skating/passing/shooting form is correct or not, which can then be used to correct the form and reduce the probability of injury.

To train a state-of-the-art pose estimation model, a large amount of video data must first be collected. Then for each frame of the video data, the location of all the keypoints must be defined. This set of keypoint locations produces a single pose

annotation corresponding to the respective frame. Then using this annotated data, a pose estimation model is trained such that given an input frame, it should output the keypoint locations of the corresponding pose annotation. An accurate pose estimation model will predict locations that are below an arbitrarily small amount of error away from the annotated keypoints.

While pose estimation models can extract valuable human motion information, a key problem that limits their applicability to a diverse amount of domains is that they require a large number of pose annotations to achieve a high accuracy [29]. Collecting this large number of annotations (typically in the tens of thousands) can be very expensive in terms of cost and time because it can require assigning multiple people to annotate examples full time and waiting months for the annotation process to complete. Due to this cost, training a pose estimation model for a new domain is not always viable for new labs/companies.

While a naive approach is to annotate the locations of every keypoint in a single frame, one frame at a time, researchers have investigated how modifying the model training and generating more annotations can reduce the number of annotations required [2, 36, 10]. However, all these approaches either lead to insufficient accuracy or include annotation requirements which makes annotation collection difficult. This thesis explores solutions of how to reduce the cost of annotation collection to train pose estimation models.

1.2 Action Recognition for Video Analytics

The definition of action recognition is the problem such that given a video that contains a specific action, classify what action was performed. Note that this thesis considers the case where the video only contains one action and does not consider videos which contain multiple actions. There are a large number of applications that rely on accurate action recognition models including classifying actions throughout an NHL game such as shooting, passing, etc., identifying when a person falls to improve response time for healthcare monitoring, and many more. A visual example of an action recognition application is shown in Figure 1.2.

To apply action recognition models to these applications, they must be as accurate as possible. A key component that greatly affects the model's final accuracy is the design of the model. For example, choosing different mechanisms to include inside the model can greatly affect how easy it is to optimize the model which, affects the model's final accuracy.

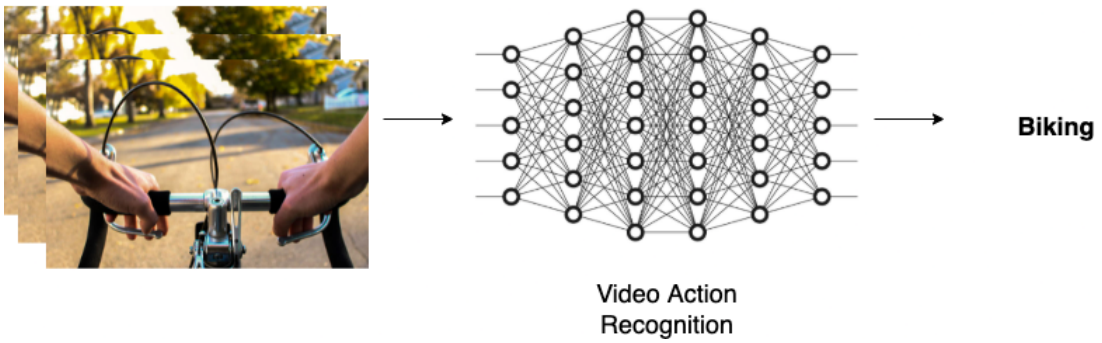

Figure 1.2: An example of video action recognition. A video of a person biking is given to the action recognition model which correctly predicts that the action is biking. This information be utilized across a large number of diverse domains to gain a deeper understanding of human motion and behaviour.

A popular choice when designing an action recognition model is to incorporate attention mechanisms. Attention mechanisms mimic cognitive attention and achieve improved performance by either enhancing or diminishing parts of the input data. While there has been a large amount of research focused on how best to utilize attention mechanisms for video action recognition [48, 17, 3], a shortcoming of these mechanisms is that they only utilize spatial information and do not leverage motion information. This thesis seeks to gain a better understanding of how to effectively leverage attention mechanisms to achieve higher accuracy scores.

1.3 Thesis Overview

While pose estimation and action recognition models are powerful tools for video analytics and understanding human motion, key problems such as the expensive pose annotation processes and achieving accurate action recognition models both need to be solved first before pose estimation and action recognition can be applied to a large number of diverse domains.

A fundamental element of videos which may be able to solve both of these problems is the motion information found across the frames of the video. While there are many ways to leverage motion information, this thesis explores how neural-network-based motion estimations can be leveraged. The two main contributions of this thesis are as follows,

1. Develop and introduce a novel motion-aware pose annotation strategy: POse annotation using Optical Flow (POOF).

2. Develop and introduce a novel motion-aware mechanism for video action recognition: Motion-Aware Attention (M2A).

The rest of the thesis is organized as follows: Chapter 2 reviews relevant background concepts; Chapter 3 introduces and examines POOF; Chapter 4 introduces and examines M2A; and lastly, Chapter 5 discusses conclusions, the impact of the thesis, and future work.

Chapter 2

Background

In this chapter, background information for each proposed contribution is discussed: Section 2.1 briefly describes how to extract motion information from videos; Section 2.2 describes previous research focused on reducing the number of annotations required to train a pose estimation model; Section 2.3 describes previous research focused on designing action recognition mechanisms.

2.1 Motion Information

Motion is fundamental to the visual experience of our world. However, due to noise, reliably extracting motion information from a given video is a difficult task. A standard approach to extract motion information is to estimate the optical flow between two consecutive frames in a video [13]. Optical flow is defined to extract the velocity in image coordinates of each pixel starting from the first frame and moving to the second frame [13]. While there are many ways to estimate the optical flow between two images, this thesis is focused on neural-network-based estimations.

Neural networks have achieved great results across many fields including computer vision, natural language processing, and more. Naturally, neural networks are also being researched to achieve improved optical flow estimation. For example, Recurrent All-Pairs Field Transforms for Optical Flow (RAFT) [41] uses neural networks to estimate optical flow and achieves improved results over other standard approaches which use algorithms to estimate optical flow [49]. Another interesting research direction is exploring how to incorporate optical-flow information into the neural networks directly [45] which could achieve improved accuracy. We further discuss the application

of both of these research directions and how they relate to this thesis's contributions in the next two sections.

2.2 POOF Background

In this section, we briefly describe research that investigates how to achieve accurate pose estimation with a small number of annotations and how this research relates to POOF. Overall, the research solutions can be generally described as either improving the annotation generation (similar to POOF) or modifying the model directly.

A standard approach to applying pose estimation to a new application is to pretrain the model on a large public dataset and then using that learned model to estimate the pose of the new application [1]. This technique is known as pretraining [1]. Pretraining can be extended to finetuning by training the pretrained model on a small number of annotations from the new application [1]. While pretraining and finetuning reduces the number of annotations required, in the case when the pretrained data and the new application data are visually different and/or contains different poses (which is the case for most applications), usually a large number of annotations are still required to learn an accurate model [14].

Research by Neverova et al. is closely related to POOF, which used motion information to extend a small number of annotations to neighbouring frames, generating a large number of annotations easily for dense keypoint estimation [33]. We extend this work by applying it to pose estimation and further investigate its strengths and weaknesses.

Instead of generating more annotations, another approach is to apply a large number of random augmentations to the small number of annotations. For example, common data augmentations include rotating, flipping, and cropping the image and then adjusting the corresponding ground-truth keypoints to account for the augmentation. While generating new annotations aim to automatically generate annotations for unannotated frames, applying augmentations aim to augment the existing annotated frames to look different. Doersch et al. found that pasting generated humans in augmented poses across a variety of background images can lead to improved generalization performance for 3D pose estimation [10]. Hinterstoisser et al. used a similar approach for object detection and found improved performance [20]. However, these techniques usually require additional data to get working (e.g., segmentation information of the poses to be able to paste on different backgrounds) which can be costly.

Another approach is to modify how the pose estimation model is trained. For example, semi-supervised learning is a strategy that creates additional tasks for the model to learn that do not require additional annotations and can lead to improved accuracy [8]. Bertasius et al. used a semi-supervised learning approach to learn a more robust pose estimation model using a small number of sparse video annotations [2]. However, their approach requires additional annotation constraints, which makes collecting annotations difficult.

2.3 M2A Background

Convolutional neural networks (CNN) are a popular neural network architecture choice for processing image data because they have achieved great results in image classification [18]. However, these models cannot be directly applied to video action recognition because of the added temporal dimension of videos. While a standard approach is to apply a CNN to each frame and then average the results [46, 22], recent research has found that adding a temporal mechanism into the model that processes temporal information and then incorporates this information back into the CNN works well without a large computational increase [28]. Simple mechanisms like the Temporal Shift Module (TSM) [28], which shifts network activations across consecutive frames, can achieve state-of-the-art accuracy while remaining computationally efficient. An active area of research is how to design these temporal mechanisms to achieve the highest possible accuracy [45, 48].

2.3.1 Attention Mechanisms

A popular type of temporal mechanisms for action recognition are attention mechanisms. Attention mechanisms mimic cognitive attention by either enhancing or diminishing parts of the input data to achieve improved performance [44]. In the context of video action recognition, attention mechanisms extract temporal information across all the frames of a video and then use this information to either excite or inhibit values of the current frame.

Using attention mechanisms for improved video action recognition has become a popular idea ever since attention showed impressive results in natural language processing tasks [44]. One of the first works to use attention for video action recognition was non-local networks [47] which proposed an attention mechanism across the frames

of a video to improve temporal modeling. The Temporal Adaptive Module mechanism (TAM) [48] further extended this idea by incorporating dynamic convolutions and attention. Recently, TimeSformer [3] processed the frames with a transformer network using a patch-based approach.

Although these attention-based mechanisms have shown impressive results, they do not incorporate any motion information which is a key feature of videos. This thesis explores how motion information can be incorporated to further improve the accuracy.

2.3.2 Motion Mechanisms

Another popular type of temporal mechanisms are motion mechanisms. Typically, motion mechanisms compute the difference of activations between consecutive frames and incorporate this information into the current frame.

Motion mechanisms are motivated by previous research that has shown that motion information is a key feature for achieving high accuracy [40] and that the use of motion information and visual features as input has been shown to outperform visual features alone [28, 5].

The Temporal Enhancement-and-Interaction Network (TEIN) [30] was one of the first motion mechanisms which investigated scaling the activations by the motion information. The Temporal Excitation and Aggregation mechanism (TEA) [27] investigated many more different architectures and mechanisms which computed the difference between consecutive frames. The Temporal Difference Network (TDN) [45] extended this work and investigated multiple motion mechanisms that utilized pooling operations to leverage motion information at multiple spatial scales. However, these methods have not investigated how to utilize state-of-the-art attention mechanisms for further performance improvements. This thesis explores how attention and motion can complement each other.

2.4 Chapter Summary

In this chapter, background information on motion information, pose estimation, and action recognition mechanisms were covered. Optical flow estimation was introduced as a method that extracts motion information from videos. Different methods for

training a pose estimation model with a small number of annotations were introduced including pretraining, annotation generation, and self-supervised learning, however, these methods include additional requirements which make the training process more difficult. Lastly, the background on temporal mechanisms for video action recognition was introduced, including attention mechanisms which have achieved great results, however, they do not leverage motion information.

With the necessary background information covered, in the next chapter, we further explore how motion can be leveraged to reduce the number of annotations required to train a pose estimation model.

Chapter 3

Data-efficient Pose Annotation using Optical Flow: POOF

In this chapter, we describe our pose annotation strategy, POse annotation using Optical Flow: POOF, which leverages motion to reduce the number of annotations required to train a pose estimation model. To generate a large number of annotations, POOF uses the temporal motion between frames of a video to propagate a small number of ground truth keypoints across neighbouring frames. This process generates a multiplicative increase in annotations with no extra cost. The contents of this chapter are largely based on its corresponding paper [14]

The chapter is organized as follows: pose estimation is mathematically defined in Section 3.1; the methodology behind POOF is described in Section 3.2; the results of our experiments are reported in Section 3.3; and lastly, the chapter is concluded in Section 3.4.

3.1 A Mathematical Definition of Pose Estimation

This section further refines the definition of pose estimation used in the introduction: "pose estimation is the problem such that given a video that includes a person, estimate the locations of a specific set of keypoints in each frame of the video.".

First, a few terms are defined. The t-th frame of the input video is defined as a three-dimensional matrix $X_t \in \mathbb{R}^{C \times H \times W}$ where C the number of channels of the frame (for RGB images $C = 3$), H is the height of the frame, and W is the width of

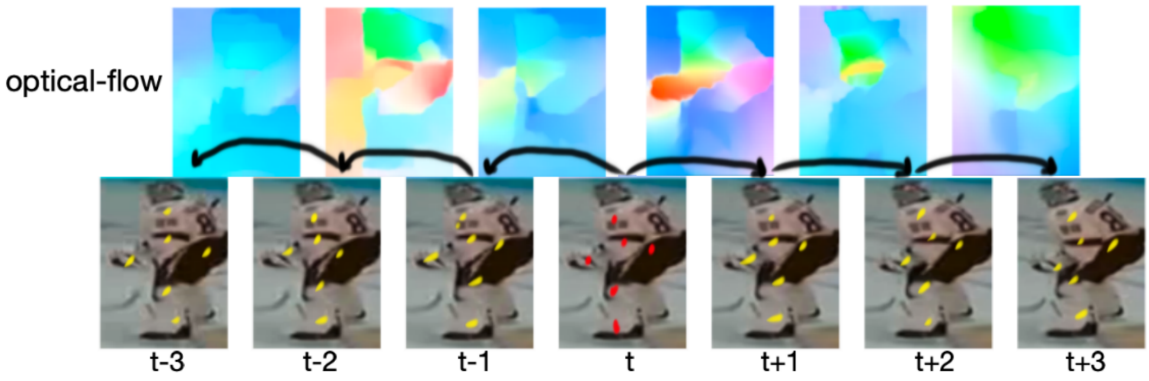

Figure 3.1: Visual description of POOF with $R = 3$. The ground-truth annotated keypoints (shown in red at time t) are propagated to annotate the neighbouring frame's keypoints (shown in gold) using the optical flow estimation between consecutive frames (estimation shown in the top row). This results in a multiplicative increase in annotations with no additional annotation cost.

the frame. The corresponding annotation of k keypoints of the t-th frame is defined as a matrix $K_t \in \mathbb{R}^{k\times 2}$ which contains the x-y coordinates of each keypoint. Then our pose estimation model is defined as f_θ with n parameters $\theta \in \mathbb{R}^n$ which takes X_t as input and outputs a matrix of estimated keypoints $f_\theta(X_t) \in \mathbb{R}^{k\times 2}$. The pose estimation problem can then be defined as the following optimization problem,

$$\underset{\theta}{\operatorname{argmin}} \, ||f_\theta(X_t) - K_t|| \tag{3.1}$$

A typical solution to optimize Equation 3.1 is to define f_θ as a neural network and optimize the equation using a large number of X_t-K_t examples and gradient descent. State-of-the-art pose estimation models typically use CNNs to process an input image and output the predicted x-y coordinates of the keypoints.

3.2 Pose Annotation using Optical Flow: POOF

This section introduces the methodology of POOF. First, a few terms are defined. The term "ground-truth annotation" is defined as a set of keypoints that has been labeled by a person which we assume to be correctly labeled. The term "pseudo-annotation" is defined as a set of keypoints that have been generated by an algorithm

that may or may not be correctly labeled. And the hyperparameter, R, is defined as the number of frames before and after the ground-truth annotation to which the keypoints will be propagated to.

We define $M_{i,j}$ as the optical flow estimation between the i-th and j-th frame represented as a three-dimensional matrix of size $H \times W \times 2$. The coordinates (k, l) are referenced in $M_{i,j}$ using $M_{i,j,(k,l)}$, which represents how the pixels of the i-th frame at coordinates (k, l) moved to the j-th frame in terms of a change in the x and y coordinates.

The first step of our motion-aware annotation strategy requires collecting a small number of ground-truth annotations across a video. We aim to have diverse annotations that cover a variety of poses that are temporally far apart from each other. Ideally, we want to select annotations that are at least $2 \times R$ frames apart. This is because when we propagate the ground-truth keypoints to the nearest R frames, if the ground-truth keypoints frames are $2 \times R$ apart, there will be no overlap in the pseudo-annotations and we will maximize the amount of annotated data created.

For each ground-truth annotation at time t, K_t, we use an optical flow estimation model to predict the motion between consecutive frames, $M_{t,t+1}$ $\forall t \in [t-R, t+R-1]$.

We then create psuedo-annotations for the frames which surround the ground-truth annotation frame, K_{t-1} and K_{t+1}, by propagating the the ground-truth annotation K_t to its neighbouring frames by explicitly leveraging the motion between the frames, $M_{t-1,t}$ and $M_{t,t+1}$, as follows:

$$K_{t-1} = K_t - M_{t-1,t,K_t} \tag{3.2}$$

$$K_{t+1} = K_t + M_{t,t+1,K_t} \tag{3.3}$$

where $M_{t,t+1,K_t}$ is $M_{t,t+1}$ indexed at the coordinates of K_t. We repeat equation 3.2 $\forall t \in [t-R, t)$ and equation 3.3 $\forall t \in (t, t+R]$ to obtain keypoints $\forall t \in [t-R, t+R]$. Figure 3.1 shows a visual description of POOF for $R = 3$.

3.3 Experiments

In the following section, the performance of models trained on annotations generated by POOF was investigated. However, first, the models and datasets used throughout the experiments are described.

3.3.1 Setup

Throughout the experiments, the publicly-available code for Multi-Stage Pose Network (MSPN) [26] was used as the pose estimation model and similarily, the publicly-available code for Recurrent All-Pairs Field Transforms For Optical Flow (RAFT) [41] was used as the optical flow estimation model. The pose estimation model was trained for 10 epochs using the Adam optimizer [24] with a learning rate of 0.01 and a batch size of 32. Through manual inspection, these values produced the best results and so were chosen to be used throughout the experiments. The optical flow estimation model used the publicly-available pretrained weights from the Sintel dataset [4].

3.3.2 Metrics

For metrics, we record the final validation accuracy of the model after training and refer to it as 'Accuracy' in the experiment tables. We define a keypoint to be accurate if the mean absolute error (MAE) between the predicted keypoint coordinate and the ground-truth keypoint coordinate is less than twenty units. We chose a threshold of twenty through visual inspection of different MAE distances across different examples [1]. We also perform further experiments on different choices of threshold values in Section 3.3.7.

3.3.3 Datasets

To investigate the performance of models trained on annotations generated by POOF, we run extensive experimental studies on a National Hockey League (NHL) goalie pose dataset, which was derived from broadcast videos and contains many similar features to other real-world datasets including but not limited to: dataset-specific poses, which are not common in public datasets, joint occlusions caused by skaters skating in front of other skaters or making contact with other skaters, and image blurriness caused from camera movement. Furthermore, the visual appearance of an NHL game is very different compared to the scenes in large public pose datasets such as the Common Objects in Context (COCO) dataset [29]. For example, this includes visual diffences

[1] The value of twenty is quite arbitrary and does not correspond to any physically consistent meaning except that the predicted keypoint is visually close to the ground-truth keypoint in the corresponding image.

Figure 3.2: An example of a predicted goalie pose using a model with COCO [29] (a large public dataset) pretrained weights. We see that the goalie is on both of his pads, which is a dataset-specific pose that is not common across large public datasets, leading to an incorrect prediction. To achieve acceptable results, more pose annotations of the goalie are required.

such as white or black goalies pads which occlude the player's leg below the knee, skates which occlude the player's feet, unique jersey colors, and more.

Throughout the data, the hockey goalie has been cropped out of the broadcast video and resized to a 256 $\times$ 192 image, the same size as images from the COCO dataset. This was done so that COCO pretrained weights could be used in the experiments. The ground-truth annotations were selected to be temporally sparse and contain a variety of poses across 6 different broadcast videos. The examples were selected to include both examples of when the goalie is not occluded at all and when the goalie is semi-occluded. The same approach was used for the validation examples, but across 2 broadcast videos (which were not included in the training set) that resulted in 16 total annotations. Throughout the experiments, we used a radius of 10 (R=10) unless stated otherwise.

Figure 3.2 shows an example from the dataset as well as the predicted pose from a model which has been pretrained on the large pose estimation dataset, COCO [29]. We can see that the model incorrectly classified the goalie's pose. This is likely because goalie images are visually different from examples in the COCO dataset including goalie pads that cover the knees and the goalie is in a pose which is not common in the COCO dataset (i.e., the goalie is on his knees).

Table 3.1: Accuracy of a pose estimation model, initialized with COCO pretrained weights, trained on different data: None which uses only pretrained weights; GT Annotations which uses 69 ground-truth annotations; and POOF which uses 69 ground-truth annotations and 1314 pseudo-annotations generated by POOF. Training on annotations generated by POOF results in the highest accuracy of 75.66%.

Init Weights	Training Data	# Annotations	Accuracy
COCO	None	0	60.53
	GT Annotations	69	23.03
	POOF	69 + 1314	**75.66**

3.3.4 Training on annotations generated by POOF

Table 3.1 shows the accuracy of a pose estimation model, initialized with COCO pretrained weights, trained on different data. We compared three types of training data: 'None' which evaluates the pretrained model directly on the dataset, 'GT Annotations' which finetunes the model on 69 ground-truth annotations (a small amount), and our proposed method, 'POOF' which trains the model on the 69 ground-truth annotations as well as 1314 pseudo-annotations derived from POOF.

Table 3.1 shows that using only pretrained weights achieves an accuracy of 60.53% and training on the small number of ground-truth annotations leads to a 37% decrease in accuracy and achieves a score of 23%. This shows that training on a small number of annotations can sometimes be worse than no annotations. One reason why training on a small number of annotations leads to worse accuracy compared to only using pretrained weights may be because when optimizing the model on a small number of annotations the model overfits to the data which leads to worse accuracy on the validation set.

We also see that training on POOF annotations achieves an accuracy of 75.66% which is a 15% increase in accuracy (from 60% to 75%) compared to using only pretrained weights (None) while using the same number of ground-truth annotations as GT Annotations. This shows that utilizing POOF annotations, which leverage the inherent motion found in videos, can significantly improve the accuracy of models compared to using only pretrained weights with only a small number of annotations.

3.3.5 The Effects of Pretrained Weights

Table 3.2: Accuracy of a pose estimation model, initialized with two different pretrained weights: None which uses randomly initialized weights; and Hockey Players which uses pretrained weights from a hockey player dataset, trained on different data: None which uses only pretrained weights; GT Annotations which uses ground-truth annotations; and POOF which uses ground-truth annotations and pseudo-annotations generated by POOF.

Init Weights	Training Data	Accuracy
None	None	0.00
	GT Annotations	0.06
	POOF	**38.82**
Hockey Players	None	69.08
	GT Annotations	**80.92**
	POOF	80.26

Using the same data (including POOF annotations) from the previous section, we also perform experiments to understand the effect of using different pretrained weights. Specifically, we compared the change in performance when using no pretrained weights/random initialization and using pretrained weights from a similar domain. We investigate using no pretrained weights/random initialization because it is likely that there will not be any pretrained weights to use when attempting to predict keypoints that are not in public datasets (e.g., hockey stick keypoints, corner of goalie pads, etc.). We also investigated using pretrained weights from a similar domain because it is likely that some companies/labs work on multiple projects and have annotations from a similar domain. As a similar domain to NHL goalie pose estimation, we choose to investigate initializing weights that were trained on a larger NHL *hockey player* pose dataset which was derived from NHL broadcast videos. This is a private dataset that was provided by the University of Waterloo, Sports Analytics group in the Vision and Image Processing Lab.

Table 3.2 shows the results of training on different types of data using three different weight initializations: randomly initialized weights (None) and pretrained weights from the hockey player dataset (Hockey Players).

Table 3.3: Accuracy of a pose estimation model using COCO pretrained weights, trained on annotations generated by POOF across different propagation radius sizes. A larger radius creates more pseudo-annotations but can lead to more incorrect pseudo-annotations, while a smaller radius creates more accurate pseudo-annotations but produces a smaller amount of pseudo-annotations.

R	# Examples	Accuracy
5	255	51.64
10	420	**61.50**
20	670	35.21

We see that when not using any pretrained weights (None), POOF significantly outperforms training on ground-truth annotations and increases the accuracy by 38% (from 0.06 to 38.82). This shows POOF is an important resource to consider when annotating new keypoints which are not included in large benchmarks. This result was not discovered in previous research.

Table 3.2 also shows that when using pretrained weights from the hockey player dataset, training on POOF annotations achieves the same accuracy as training on the small number of ground-truth annotations (80.26% vs 80.92%). This shows that when the domains of the pretrained weights and the new dataset are similar it may be best to train on a small number of annotations. This agrees with the results found by [33]. However, this also shows that POOF is most effective when the domains between the pretrained dataset and the new dataset are different, which was not discovered in previous research.

3.3.6 The Effect of the Propagation Radius

We also investigated the effect of using different propagation radius (R) values. Table 3.3 shows the accuracy of a trained model on POOF annotations which were generated using a R value of 5, 10, and 20. Note that in this experiment, we used a different set of goalie pose annotations so the accuracy results are different from previous experiments.

Table 3.3 shows that the best accuracy is achieved when we train a model using annotations generated by POOF derived with a propagate radius of ten ($R = 10$).

We hypothesize that when $R = 5$, POOF did not generate enough annotations and that when $R = 20$ POOF generated too many incorrect pseudo-annotations because of small errors in close frames which result in larger errors in frames further away including occlusions (e.g, hockey players skating between the camera and goalie), blurriness (e.g, from camera movement), and more. This result is likely very dataset specific. For example, since hockey games include a large amount of occlusions due to hockey players slaming into eachother, fast movements, and the image resolution is not particularly high, there is a high sensitivity to the chosen R value. However, in datasets with less occlusions and better motion characteristics, the accuracy may not be as sensitive to a different R values.

While [33] only investigated using a radius of 3 frames, POOF shows that performance can be improved using a radius of up to 10 frames.

In practice, R should be selected based on the dataset. If occlusions and blurriness are minimized throughout the dataset, then keypoint propagation should work better for a longer distance, and so a larger R value should be chosen. However, if occlusions and blurriness occur often in the dataset, then a lower R value should be chosen to reduce the number of incorrect pseudo-annotations generated. Future research related to R is further discussed in Section 5.3.1.

3.3.7 Various Accuracy Thresholds

To further understand the performance improvement when training a model on annotations generated by POOF, we also investigated the accuracy score across different accuracy threshold values (which represents the maximum distance a predicted keypoint can be from the ground-truth keypoint and still be classified as correct).

Figure 3.3 shows the accuracy divided by 100 on the y-axis and different accuracy thresholds on the x-axis. The black line is a straight line that represents 100% accuracy. A steeper slope corresponds with a better model.

We can see that the lines which use POOF (orange and red) are much steeper than the lines which do not (green and blue). If we look at the orange vs green and red vs blue lines, we see that there is a large improvement using POOF across many accuracy thresholds. This further confirms that training on POOF annotations can improve the accuracy of models compared to using only pretrained weights

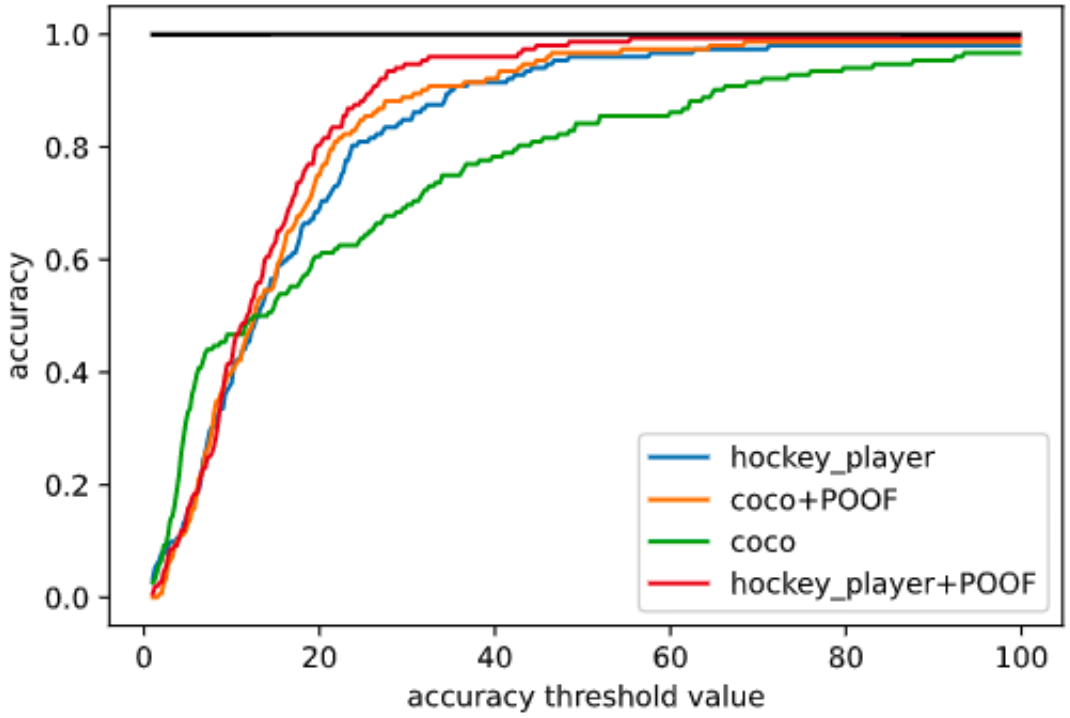

Figure 3.3: Figure with the accuracy divided by 100 on the y-axis and the accuracy threshold value on the x-axis. We compared results using weights that were pretrained to predict hockey player poses (hockey_player), COCO pretrained weights (COCO) both with (+POOF) and without POOF. We see that utilizing POOF annotations can increase the accuracy score across most threshold values for both Hockey Player and COCO pretrained weights.

Table 3.4: Accuracy on specific joints with (+POOF) and without POOF using different pretrained weights (e.g., COCO and HockeyPlayer). Change in accuracy using POOF in brackets. The largest joint accuracy improvements are bolded. We see that POOF consistently improves the accuracy of most joints by a significant amount.

Joint	COCO	+POOF	HockeyPlayer	+POOF
L shoulder	86	93 (+7%)	80	93 (+13%)
R shoulder	100	100 (+0%)	100	100 (+0%)
L elbow	50	64 (+14%)	71	78 (+7%)
R elbow	80	80 (+0%)	90	90 (+0%)
L wrist	26	**66 (+40%)**	40	53 (+13%)
R wrist	58	50 (-8%)	33	58 (+25%)
L hip	58	83 (+25%)	66	91 (+25%)
R hip	81	72 (-9%)	63	72 (+9%)
L knee	57	85 (+27%)	57	**85 (+28%)**
R knee	33	50 (+17%)	66	75 (+9%)
L ankle	66	80 (+14%)	80	86 (+6%)
R ankle	41	75 (+34%)	91	83 (-8%)
Mean	61	74 (+13%)	69	80 (+11%)

3.3.8 Change in Per-Joint Accuracy

Lastly, we investigated the accuracy improvement across each joint to further understand where POOF's performance improvement comes from.

Table 3.4 shows the accuracy across all the joints. The joint names are formatted to have the side of the body, followed by the body part (e.g., the left shoulder keypoint is formatted as L shoulder). The second column shows the results of the initial weights used (without any training) (e.g., COCO) and the third column (e.g., +POOF) shows the results after applying POOF. The same format is used in the fourth and fifth columns which are used to compare using pretrained weights from the hockey player dataset without POOF (e.g., HockeyPlayer) and with POOF (e.g., +POOF). We show the percentage improvement achieved when using POOF in brackets.

We see that POOF improves the accuracy by more than 10% in 7/12 joints (e.g., +40% L wrist in the COCO columns). However, POOF also sometimes results in worse accuracy (e.g., -9% R ankle in the HockeyPlayer column). We hypothesize this could be because the model overfit to the noise in the propagated keypoints. In practice, this could be solved by using an ensemble of models where for each keypoint the best performing model is used to predict it.

3.4 Chapter Summary

In this chapter, we introduced POOF, a data-efficient motion-aware pose annotation strategy that utilizes optical flow to propagate ground-truth annotations to neighbouring frames. POOF improves on the previous work of pose estimation solutions by removing data annotation constraints such as requiring a ground-truth keypoint every n-frames. Using an NHL goalie dataset derived from broadcast video, we show that POOF can improve performance with a very small amount of annotations and that it performs best when transferring models between different domains (in Table 3.2). We also show POOF can achieve significantly improved results over using pretrained weights across various accuracy thresholds. Furthermore, we showed this performance improvement is achieved across most individual joints and also suggested multiple directions for future research. Overall, by explicitly leveraging the inherent motion found in datasets that are derived from videos, this research should significantly reduce the time required for annotating pose data across different domains without compromising model accuracy and allow pose estimation to be more easily applied to a wide variety of domains.

While its clear motion can improve the data efficiency of pose estimation, in the next chapter, we explore how motion can also be leveraged to improve video action recognition models.

Chapter 4

Motion-Aware Attention for Video Action Recognition

In this chapter, we explore how motion can be leveraged to improve action recognition models and introduce our motion-aware attention mechanism (M2A). We make our code publicly-available at https://github.com/gebob19/M2A. The contents of this chapter are largely based on its corresponding paper [15].

The chapter is organized as follows: Section 4.1 introduces a mathematical definition of attention; Section 4.2 explains the methodology of M2A; Section 4.3 reports experimental results on the M2A mechanism; and Section 4.4 concludes the chapter.

4.1 A Mathematical Definition of Attention

Attention was initially popularized by achieving great results in natural language processing [44] and has since been applied to a wide range of fields, one of which is video action recognition [47, 27, 3]. In this section, we provide a mathematical definition of attention, which we then utilize in our motion-aware mechanism.

For a sequence of length of n where each item in the sequence is a vector with size d, attention is defined to operate a query matrix $Q \in \mathbb{R}^{n \times d}$, a key matrix $K \in \mathbb{R}^{n \times d}$, and a value matrix $V \in \mathbb{R}^{n \times d}$ [44]:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4.1}$$

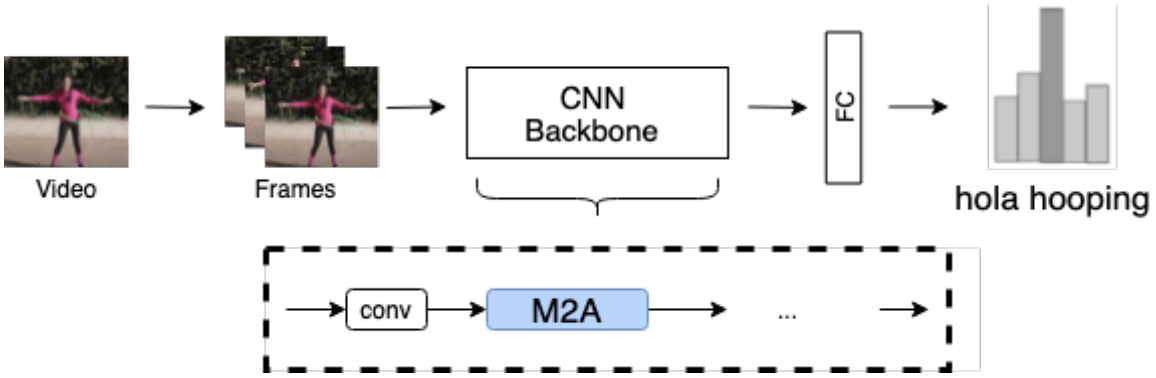

Figure 4.1: Overview of the proposed M2A mechanism within the context of deep-learning-driven video action recognition. The M2A mechanism can be added to any deep neural network backbone architecture to explicitly leverage motion characteristics. Frames are first sampled from the video, are processed by the network, and are classified using a fully-connected layer (FC).

where softmax is the softmax function and $\sqrt{d_k}$ scales the product to reduce the impact of large dot-product values in the numerator. Intuitively, the dot-product operation between Q and K produces a similarity score between the query and the key matrices which then scales the value matrix V, exciting and inhibiting certain values.

With a video being defined as a four-dimensional matrix $U \in \mathbb{R}^{T \times C \times H \times W}$ where T is the number of frames, to apply attention mechanisms across a video, U is reshaped to $\hat{U} \in \mathbb{R}^{T \times [C * H * W]}$. The Q, K, and V vectors are then defined each as different linear projections of $\hat{U}$ with learned weight matrices W_Q, W_K, and W_V:

$$W_Q \hat{U} = Q \tag{4.2}$$

$$W_K \hat{U} = K \tag{4.3}$$

$$W_V \hat{U} = V \tag{4.4}$$

while this is the case for standard attention, in this thesis we do not apply a linear projection and instead set Q, K, and V equal to $\hat{U}$ to reduce the computational cost of the mechanism. This type of attention is also refereed to as self-attention [9]. All together, this thesis's attention is defined as:

$$\text{Attention}(U) = \text{softmax}\left(\frac{\hat{U}\hat{U}^T}{\sqrt{d_k}}\right)\hat{U} \tag{4.5}$$

4.2 Motion-Aware Attention (M2A)

In this section, we introduce the methodology of our motion-aware attention (M2A) mechanism. We define the input activation as a four-dimensional matrix $X \in \mathbb{R}^{T\times C\times H\times W}$ where T is the temporal dimension, C is the number of channels of each frame[1], H is the height of each frame, and W is the width of each frame. Figure 4.2 shows a visual description of our mechanism. M2A consists of four key stages:

1. **Channel Reduction**: following the standard practice of action recognition mechanisms, we apply a convolution operation to our input X to reduce the number of channels by a factor of R. This allows us to compute future operations efficiently (we use $R = 8$ throughout the paper). This produces a new vector, $X_t \in \mathbb{R}^{T\times[C/R]\times H\times W}$. Following standard attention-based practice, we then apply a layer normalization operation.

2. **Motion Mechanism**: we then compute a shifted representation of X_t, which we denote as X_{t+1}, by shifting the temporal axis of X_t to the left and filling the last index with the values of the first frame. To extract motion information, we compute the difference between X_{t+1} and X_t: $X_{t+1} - X_t$. This represents the difference in activation values between consecutive frames, emulating the motion between frames.

3. **Attention Mechanism**: to help focus on motion patterns found across frames, we flatten the frames to create a matrix with shape $T \times [H * W * C/R]$ and apply self-attention across the time axis. This is followed by a skip connection of the first convolved input.

4. **Incorporation**: based on previous research [45, 27, 30], we apply a convolution operation to increase the number of channels from C/R back to C, followed by

[1]Since M2A is inserted into the neural network, the input to the mechanism is not the original image but a network activation. Network activations typically have C values larger than 3 (e.g., 256 is a common C value for network activations). We use C to describe both the channels for the input image and the channels for a network activation to be succinct.

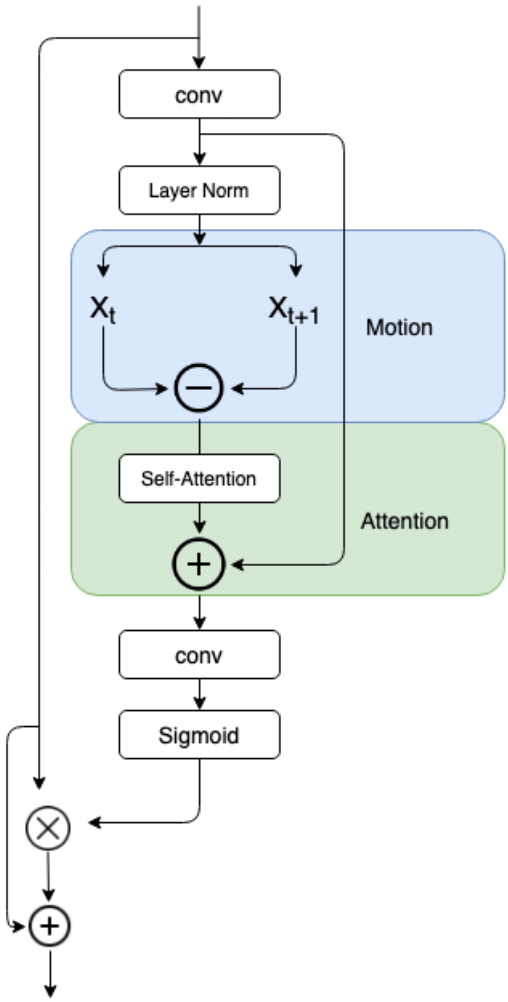

Figure 4.2: The proposed M2A mechanism consists of a motion block (shown in blue) which extracts motion information across consecutive frames and an attention block (shown in green) which focuses on relevant motion patterns found across frames.

the *sigmoid* activation function which scales value to within the $[0, 1]$ range. We then apply an element-wise multiplication on the original input, followed by a skip connection to incorporate the temporal information back into the current frame.

4.3 Experiments

In this section, we report the results of our experiments. Following standard practice [45], the temporal mechanism is inserted after the first convolution of each backbone's block. Furthermore, all backbones are initialized with ImageNet [37] pre-trained weights. We performed all our experiments on the Something-Something V1 (SSv1) [16] dataset, a standard video action recognition benchmark. Figure 4.3 shows three example video sequences from the benchmark dataset with their corresponding labels. Throughout the experiments, we uniformly sampled 8 frames from the video and use them as input to the model; we also use a 2D-ResNet18 backbone unless stated otherwise. To evaluate the performance of our models, we report the number of Giga multiply-accumulate operations per video (GMACs/video) as a measure of model efficiency, and the Top-1 and Top-5 accuracy (Top-1 Acc and Top-5 Acc) as a measure of the model's performance. Throughout the experiments, when comparing different mechanisms, the respective paper's publicly-available code is always used. We train all the models on a desktop computer that has four GPUs and 24 CPUs.

4.3.1 Ablation Experiments

To understand how motion and attention mechanisms contribute to the model's performance and if the results generalize across different backbones we run ablation studies across three backbones: 2D-ResNet18 [19], 2D-MobileNetV2 [38], and I3D-inflated-Resnet18 [5, 7]. Specifically, we compared each backbone with no temporal mechanism (None), with M2A but with the motion block removed (M2A-Attention), with M2A but with the attention block removed (M2A-Motion), and with the full M2A mechanism (i.e., with both motion and attention blocks) (M2A). In the tables, the percentage improvement from None is noted in brackets and the best accuracy is bolded.

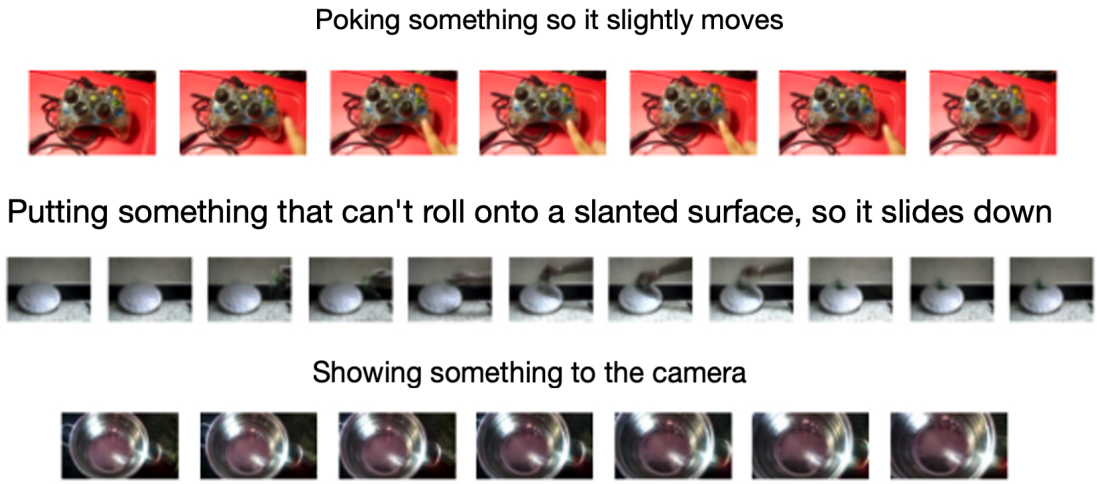

Figure 4.3: Example video sequences from the Something-Something V1 dataset and their corresponding labels.

2D-ResNet18 Backbone

We first perform our ablation experiments using the 2D-ResNet18 backbone since it is a popular choice for computer vision tasks and has shown to achieve good results without being very computationally expensive [19]. Table 4.1 shows the results of the ablation experiments. We see that M2A-Attention improves upon the Top-1 accuracy compared to None by +3%, but it is unable to achieve a large improvement. This could be because it is only focusing on similar visual features across frames and is unable to extract motion information. We also see that M2A-Motion outperforms None by +15%, which shows the importance of extracting motion information for improved video action recognition performance. Lastly, M2A achieves the highest accuracy improvement of +20%, meaning that using attention to focus on motion patterns across frames is the best compared to using only motion or only attention. More specifically, comparing M2A to M2A-Attention, we see that incorporating motion with attention mechanisms outperforms attention-only mechanisms by +17% in Top-1 accuracy which further supports the idea that incorporating motion mechanisms with attention mechanisms can lead to improved results for video action recognition. We see a similar trend for the Top-5 accuracy which shows that these improvements are achieved across multiple metrics. Lastly, we see that including the

M2A mechanism only increases the GMACs/video by only +1.6% showing that M2A is very computationally efficient.

Table 4.1: Ablation study of different temporal mechanisms using a 2D-ResNet18 backbone. M2A achieves the highest Top-1 accuracy improvement of +20% compared to None.

Mechanism	GMACs/video	Top-1 Acc	Top-5 Acc
None	14.57	13.9	38.9
M2A-Attention	14.79	16.9 (+3%)	42.0 (+3%)
M2A-Motion	14.79	28.9 (+15%)	56.7 (+17%)
M2A	14.81	**34.7 (+20%)**	**63.4 (+24%)**

Table 4.2: Ablation study of different temporal mechanisms using a 2D-MobileNetV2 backbone. M2A achieves the highest accuracy improvement of +21% compared to None with only 2.58 GMACs/video.

Mechanism	GMACs/video	Top-1 Acc	Top-5 Acc
None	2.55	13.8	37.8
M2A-Attention	2.58	13.3 (-0.5%)	37.2 (-0.6%)
M2A-Motion	2.58	32.0 (+18%)	60.6 (+22%)
M2A	2.58	**35.6 (+21%)**	**64.4 (+26%)**

Table 4.3: Ablation study of different temporal mechanisms using a I3D-ResNet18 backbone. We see only a small improvement when using M2A compared to None. This means that M2A is unlikely to further improve 3D CNNs.

Mechanism	GMACs/video	Top-1 Acc	Top-5 Acc
None	22.52	27.0	53.5
M2A-Attention	22.65	26.9 (-0.1%)	**54.1 (+0.6%)**
M2A-Motion	22.65	26.3 (-0.7%)	53.6 (+0.1%)
M2A	22.67	**27.1 (+0.1%)**	53.5 (0%)

2D-MobileNetV2 Backbone

Relative to 2D-ResNet18 backbones, 2D-MobileNetV2 backbones are typically used in more resource-constrained settings where computational power and latency must be kept to a minimum (e.g., mobile and edge devices). Since this is a common setting for video action recognition, we perform the same ablation studies using the 2D-MobileNetV2 backbone. Table 4.2 shows the results of the ablation experiments. We see that the ablation study follows a similar trend as in the 2D-ResNet18 ablation study which further supports the idea that further accuracy improvements can be achieved by incorporating motion mechanisms with attention mechanisms for video action recognition. Furthermore, we see that M2A achieves a higher Top-1 accuracy with the 2D-MobileNetV2 backbone (i.e., 35.6%) compared to the 2D-ResNet18 backbone (i.e., 34.7%) while having more than five times lower GMACs/video (i.e., 14.81 GMACs/video with a 2D-ResNet18 backbone and 2.58 GMACs/video with a 2D-MobileNetV2). This shows that M2A generalizes across different backbone architectures and is a viable option in resource-constrained settings.

I3D-ResNet18 Backbone

I3D-inflated-ResNet18 backbones use 3D convolutions instead of 2D convolutions. While 2D CNNs model the frames individually, these 3D CNNs explicitly model the temporal aspect of videos. This has been shown to achieve accurate results without requiring any additional temporal mechanisms. To understand if M2A can improve 3D convolution networks, Table 4.3 shows the results of the same ablation study but uses an I3D-inflated-ResNet18 backbone. We see only a small improvement using M2A compared to None. This means that M2A is unlikely to further improve 3D CNNs since temporal information is already modelled accurately. However, we also see that using 3D CNNs is much more computationally expensive requiring 22 GMACs/video. Furthermore, if we compare these results to our 2D-MobileNetV2 experiments in Table 4.2 we see that M2A achieved a Top-1 accuracy of 35.6% with only 2.58 GMACs/video while using 3D CNNs achieved only 27.0% Top-1 accuracy with approximately 10 times the computational cost. This shows that our M2A mechanism can outperform 3D CNNs while being significantly more computationally efficient.

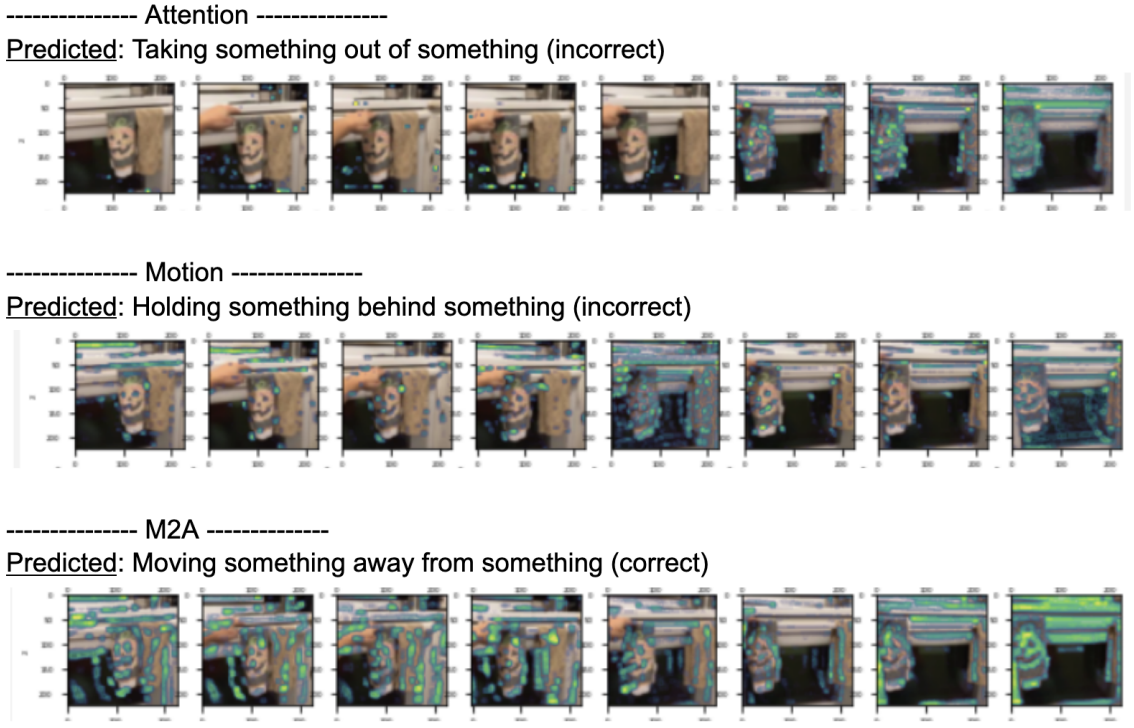

Figure 4.4: Grad-CAM heatmaps of a ResNet18 backbone on an example video sequence from the Something-Something V1 validation dataset with just the attention mechanism (i.e., M2A-Attention), just the motion mechanism (i.e., M2A-Motion), and the proposed M2A mechanism which incorporates both motion and attention. Images are of the last five frames of an eight-frame video sequence. We see that when we combine both motion and attention within the proposed M2A mechanism the model pays attention to a large number of the frames and correctly classifies the video as moving something away from something, whereas it is unable to classify correctly when we use just motion or just attention.

4.3.2 Visualizing Model Focus with Grad-CAM

To further understand the difference between motion, attention, and both motion and attention, we visualized where the model is focusing using Grad-CAM [39] heatmaps.

Grad-CAM works by computing how a specific layer's activations A_l contribute to classifying a specific class y_c: $G = \frac{\partial y_c}{\partial A_l}$. Activation values that have a large G value contribute greatly to the classification of the specific class and thus it is said that the model focuses on those values. The opposite is also true for activation values that have a small G value. Using this information Grad-CAM can visualize what areas of the images are important to classifying a specific class. In our experiments, we apply Grad-CAM with y_c equal to the ground-truth class and visualize an arbitrary A_l since it is not feasible to visualize all possible layers.

Figure 4.4 shows the resulting heatmaps extracted from the second layer of a 2D-ResNet18 backbone. The blue/green values show where the model is focusing. We also include the predicted and ground truth class.

The first section of Figure 4.4 shows the results when only using the attention component of the proposed M2A mechanism (i.e., M2A-Attention). It can be observed that when we only use an attention mechanism to model temporal information, the model only seems to focus on the last few frames. Furthermore, it incorrectly classifies the action as "Taking something out of something".

The second section shows the results when only using the motion component of the proposed M2A mechanism (i.e., M2A-Motion). Here the model does not seem to focus on anything specific and incorrectly classifies it as "Holding something behind something".

In the last section, we see the results from the proposed M2A mechanism. We see that when we combine both motion and attention within the proposed M2A mechanism the model pays attention to a large amount of the frames and correctly classifies the video as "Moving something away from something", whereas it is unable to when we use just motion or just attention. This shows that combining motion with attention can lead to the model focusing on more important details in the video to correctly classify the action.

Table 4.4: Comparison of different state-of-the-art (SOTA) attention mechanisms used in M2A with a 2D-ResNet18 backbone. We found that M2A's attention performed the best and that there was not a significant performance difference across the other SOTA attention mechanisms.

Attention mechanism	Top-1 Acc
M2A-Attention	**34.7**
TAM [48]	31.6
S+T Patch (s=4) [3]	31.9
S+T Patch (s=8) [3]	34.2

4.3.3 Using SOTA Attention

We also investigated if we could improve the performance of M2A by using state-of-the-art (SOTA) attention mechanisms. Table 4.4 shows the results of M2A using M2A's attention block (M2A-Attention), using space and time based attention block (S+T Patch with a patch of size $s \times s$) from [3], and using TAM attention [48]. We found that M2A's attention performed the best and that there was not a significant performance difference across the other SOTA attention mechanisms.

4.3.4 Comparison to SOTA

Next, we compared M2A's performance to other SOTA temporal mechanisms including, TSM [28], TEA [27], TDN [45], and TAM [48]. TSM does not incorporate any attention or motion information and instead shifts values across consecutive frames, while TEA and TDN are motion-only temporal mechanisms, and TAM is an attention-only temporal mechanism. The TDN and TEA papers involve multiple mechanisms, some of which modify the backbone directly, so to conduct a fair comparison, we only use the individual temporal mechanisms of each method. Specifically, we use the long-term mechanism for TDN and the motion excitation mechanism for TEA. Furthermore, to understand if M2A can further improve the performance of other temporal mechanisms, we also compare using both M2A and TSM (M2A + TSM).

Table 4.5 shows that M2A can achieve higher accuracy than complex state-of-the-art motion/attention-only mechanisms (TEA, TDN, and TAM). Furthermore, we see

Table 4.5: Comparison of different state-of-the-art temporal mechanisms. We find M2A can achieve higher accuracy than complex SOTA motion/attention-only mechanisms (TEA, TDN, and TAM). Furthermore, we see M2A + TSM achieves the highest accuracy showing that M2A is a complementary mechanism.

Temporal mechanism	GMACs/Video	Top-1 Acc
M2A + TSM	14.81	**39.3**
TSM [28]	14.57	39.0
M2A	14.81	34.7
TEA [27]	14.83	34.3
TDN [45]	15.13	28.6
TAM [48]	14.79	21.0

that M2A is comparable in terms of GMACs/Video to all the other mechanisms which shows that it is a computationally efficient mechanism. Comparing M2A to TSM, we see that TSM outperforms M2A in Top-1 accuracy by approximately 4%. However, we also see that M2A + TSM outperforms TSM by +0.3% in Top-1 accuracy, showing that M2A is a complementary mechanism that can be combined with other temporal mechanisms to achieve better performance.

Per-Class Comparison to SOTA

To further understand where M2A outperforms the other SOTA mechanisms, Fig 4.5 shows the difference in Top-1 accuracy across all the classes in the SSv1 dataset when M2A is compared to TAM, TDN, and TEA. Specifically, each bar represents a class in the SSv1 dataset (e.g., Showing something to the camera) and the height of the bar represents the difference between M2A and the compared mechanism (e.g., in the M2A VS TAM chart, the largest bar has a height of 60 which means M2A achieves 60% better accuracy in that class compared to TAM). We see M2A achieves large improvements in most classes when compared to the other SOTA mechanisms, specifically, we see up to +60%, +20%, and +10% Top-1 accuracy improvements compared to TAM, TDN, and TEA respectively. Furthermore, we investigated which specific classes had the largest difference. Comparing M2A to TAM, the largest improved classes were:

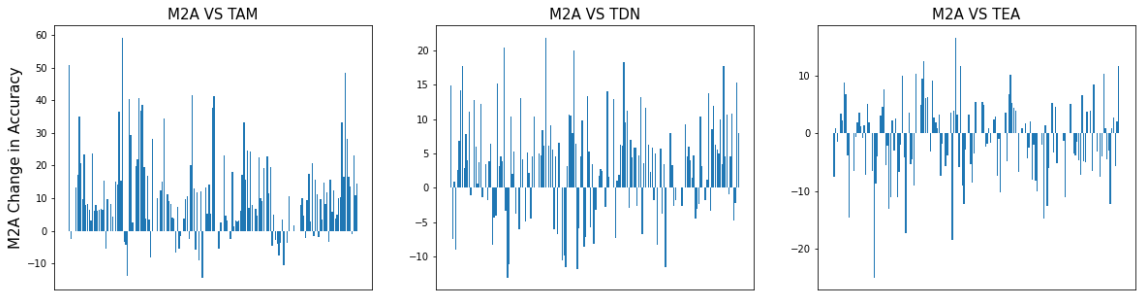

Figure 4.5: The change in accuracy using M2A compared to other SOTA mechanisms for each class in the SSV1 dataset. Each bar represents a class in the SSV1 dataset (e.g., Showing something to the camera) and the height of the bar represents the difference between M2A and the compared mechanism (e.g., in the M2A VS TAM chart, the largest bar has a height of 60 which means M2A achieves 60% better accuracy in that class compared to TAM). We see M2A achieves large improvements in most classes when compared to the other SOTA mechanisms.

- "Moving away from something with your camera" (+59.3%)
- "Approaching something with your camera" (+50.7%)

Comparing M2A to TDN, the largest improved classes were:

- "Poking something so that it falls over" (+21.9%)
- "Moving away from something with your camera" (+20.4%)

And lastly comparing M2A to TEA, the largest improved classes were:

- "Pretending to put something on a surface" (+16.7%)
- "Poking something so it slightly moves" (+12.5%)

We see that M2A improves on motion-oriented classes which include 'moving' and 'approaching' (i.e., "Moving away from something with your camera") and interaction classes such as 'poking something' (i.e., "Poking something so it slightly moves"). This shows that incorporating motion information with attention mechanisms can improve the classification of motion-oriented classes and interaction-based classes in videos compared to using just motion or just attention mechanisms.

Table 4.6: Comparison of different SOTA motion and attention mechanisms inserted into a ResNet18 backbone. We also show the improvement made by incorporating motion with each attention mechanism in brackets and bold the highest accuracy for each attention mechanism. We see that all attention methods are improved when motion is incorporated.

Motion Mechanism	Top-1 Acc		
	None	M2A-Attention	TAM-Attention [48]
None	13.9	16.9	21.0
M2A-Motion	28.9	**34.7 (+17.8%)**	31.6 (+10.6%)
TDN-Motion [45]	28.6	29.0 (+12.1%)	25.3 (+4.3%)
TEA-Motion [27]	**34.3**	33.9 (+17.0%)	**33.3 (+12.3%)**

4.3.5 Extending SOTA motion/attention-only mechanisms

Lastly, we attempted to extend state-of-the-art motion-only and attention-only mechanisms by incorporating attention and motion respectively. The results are shown in Table 4.6. The first column states the motion mechanism used and the first row states the attention mechanism used. For example, the cell which intersects M2A-Motion and M2A-Attention is the full M2A mechanism. We also show the improvement made by incorporating motion with each attention mechanism in brackets and bold the highest accuracy for each attention mechanism.

We see that all attention methods are improved when motion is incorporated (e.g., TAM-Attention achieves 21.0% Top-1 accuracy without motion and achieves 33.3% Top-1 accuracy when combined with TEA-motion). However, we also see that the motion mechanisms are not always improved when attention mechanisms are incorporated. Specifically, TDN and TEA mechanisms see a very small or negative change in accuracy when attention is added to them (e.g., TEA-Motion achieves 34.3% Top-1 accuracy without attention and 33.9% with M2A-Attention). This may be because their mechanisms do something similar to attention mechanisms so incorporating attention mechanisms with them does not lead to any improvements.

4.4 Chapter Summary

In this chapter, we introduce a new temporal mechanism, motion-aware attention (M2A), which utilizes both motion and attention for accurate video recognition. We showed that M2A can accurately recognize actions across multiple CNN backbones including 2D-ResNet18, 2D-MobileNet, and I3D-ResNet18 and that the proposed M2A mechanism can lead to a +15% to +26% improvement in Top-1 accuracy with only a small increase in computational complexity. Furthermore, we showed how other SOTA attention mechanisms can be further improved by explicitly incorporating motion characteristics. Lastly, we also showed that M2A achieves competitive accuracy and efficiency compared to other SOTA temporal mechanisms and can lead to up to +60% in Top-1 accuracy across specific classes in SSV1. We hope this research helps develop more accurate and efficient temporal mechanisms for video action recognition.

In the next chapter, we conclude the thesis by summarizing the contributions, discussing the impact of the thesis, and discussing future research directions.

Chapter 5

Conclusion

In this chapter, Section 5.1 briefly summarize the thesis, Section 5.2 discusses the impact of this thesis's work, and lastly, Section 5.3 discusses possible future work.

5.1 Summary of Thesis and Contributions

In this thesis, we introduced two novel motion-aware strategies that explicitly leverage motion to achieve improved video analytics:

This thesis introduced POOF, a motion-aware pose annotation strategy that leveraged the motion found in pose datasets that were derived from videos to create a multiplicative increase in annotations with no additional cost. Furthermore, unlike previous research, the approach did not have any constraints such as requiring a ground-truth keypoint every n-frames. Using an NHL goalie dataset derived from broadcast video, we showed that POOF can improve performance with a very small amount of annotations and that it performs best when transferring models between different domains. Furthermore, we showed this performance improvement is achieved across most individual joints.

This thesis also introduced a new temporal mechanism, motion-aware attention (M2A), which utilizes both motion and attention to achieve accurate video recognition. We showed that utilizing motion with attention mechanisms is critical to achieving the best performance compared to using only using one or the other. Furthermore, we showed that this result occurs across both the 2D-ResNet18 backbone and the 2D-MobileNet backbone resulting in a +15% to +26% improvement in Top-1

accuracy with only a small increase in computational complexity. Furthermore, we showed how other SOTA attention mechanisms can be further improved by explicitly incorporating motion characteristics. Lastly, we showed that M2A achieves competitive accuracy and efficiency compared to other SOTA temporal mechanisms and can lead to up to +60% in Top-1 accuracy across specific classes in the SSV1 dataset.

5.2 Impact of Thesis

Improving our understanding of human motion can solve many important problems across many fields including robotics, ML-powered personal trainers, augmented reality, and more, which affects millions of people worldwide. While machine learning has the potential to solve these problems, key problems such as data-efficient and model accuracy must be solved first.

This thesis's contribution of POOF should significantly reduce the amount of time required for annotating pose data across different domains without compromising accuracy and allow pose estimation to be more easily applied to a large number of diverse domains.

This thesis's contribution of M2A improves our understanding of how motion mechanisms can be both accurate and efficient, and how they can improve state-of-the-art mechanisms such as attention. This understanding should enable action recognition solutions to be applied to real-world problems sooner.

Overall, this thesis showed that a fundamental aspect of video analytics is the motion found between frames and that if this information is explicitly utilized, it can achieve improved accuracy and enable data-efficient strategies for improved video analytics and solve the key problems of data efficiency and accuracy.

5.3 Future Research

In this section, we describe some limitations of POOF and M2A and some potential future research directions.

5.3.1 POOF

More Diverse Datasets

One limitation of our research is that we only tested POOF on a single NHL goalie dataset due to lack of time. It would be interesting to experiment across a wider variety of datasets to assess the performance consistency of POOF. One example that would be interesting to investigate is different sports such as soccer or basketball where the athlete's motion and the video characteristics are very different compared to hockey.

Additional Studies on Hyperparameters

Another avenue for future research is to further investigate the effect of using different R values. In our research, we only investigated three potential values, but it would be interesting to test more values to further understand how different values of R affect the model's accuracy. Ideally, one could also investigate how to select R quantitatively rather than qualitatively for easier hyperparameter selection. For example, one approach might be to annotate two images which are t frames apart from each other, compute the optical flow between them twice (once starting at the first frame going towards the last frame, and another starting from the last frame going backwards to the first frame), and propagate the ground-truth annotations from each frame to the opposite frame. Then for the two ground-truth frames, the error between the ground-truth annotation and the propagated annotation could indicate if a value of $R = t$ would be a good selection. If the error between the propagated-annotation and the ground-truth annotation is small, then $R = t$ is a good choice because the propagated keypoints will be correct. If the error is large, the R value is likely too large to use with the dataset because the keypoint propagation will lead to inaccurate keypoints. A search algorithm can be included on top of this approach, searching for the largest value of R which achieves an error below a specific threshold. While Table 3.3 showed that the accuracy decreases when using a radius value of 20 because it is too large, it would be interesting to investigate what kinds of datasets enable using larger values.

The Problem of Transforming Keypoints

One of the main limitations with POOF is that the optical flow estimation is unable to account for keypoints that start as visible and later transform to become occluded

by either another object occluding the keypoints or because the person rotates in a way that occludes the keypoint of interest. Furthermore, POOF is unable to account for keypoints that were labelled as occluded but transform to become visible later in the video. Different solutions could be experimented with to solve this problem which could allow us to label longer sequences and reduce the amount of noise in the propagated annotations. This would likely further improve the model accuracy. One potential solution could be to incorporate visual information in the keypoint propagation stage similar to [6].

5.3.2 M2A

Improved Motion-Aware Mechanisms

While we designed M2A to be a simple mechanism to show how motion can be leveraged for improved action recognition, future work may investigate more complex motion-aware mechanisms for further improved accuracy. Specifically, it would be interesting to see how neural architecture search methods [12] could be leveraged to discover new mechanisms with even better accuracy.

Other Datasets

While we experimented with M2A on a first-person oriented action recognition dataset, it would be interesting to see how it performs on a dataset with a third-person orientation. For example, the Kinetics dataset [23] is a popular third-person orientation dataset that could be experimented on. This would be interesting for applications that require a third-person orientation. One example of this is healthcare monitoring, which monitors the actions of residents and automatically recognizes important events (e.g., recognizing that a person has fallen). Enabling more accurate action recognition models could enable the automation of these tasks which could lead to sending help to a resident faster and improving the safety of all residents.

References

[1] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 17–36. JMLR Workshop and Conference Proceedings, 2012.

[2] Gedas Bertasius, Christoph Feichtenhofer, Du Tran, Jianbo Shi, and Lorenzo Torresani. Learning temporal pose estimation from sparsely-labeled videos. *arXiv*, (NeurIPS):1–12, 2019.

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is Space-Time Attention All You Need for Video Understanding? arXiv preprint arXiv:2102.05095, 2021.

[4] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision*, pages 611–625. Springer, 2012.

[5] João Carreira and Andrew Zisserman. Quo Vadis, action recognition? A new model and the kinetics dataset. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:4724–4733, 2017.

[6] James Charles, Tomas Pfister, Derek Magee, David Hogg, and Andrew Zisserman. Personalizing human video pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3063–3072, 2016.

[7] Chun-Fu Richard Chen, Rameswar Panda, Kandan Ramakrishnan, Rogerio Feris, John Cohn, Aude Oliva, and Quanfu Fan. Deep analysis of cnn-based spatio-temporal representations for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6165–6175, 2021.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[9] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 551–561, Austin, Texas, November 2016. Association for Computational Linguistics.

[10] Carl Doersch and Andrew Zisserman. Sim2real transfer learning for 3d human pose estimation: motion to the rescue. *Advances in Neural Information Processing Systems*, 32, 2019.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv preprint arXiv:2010.11929, 2020.

[12] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *The Journal of Machine Learning Research*, 20(1):1997–2017, 2019.

[13] David Fleet and Yair Weiss. Optical flow estimation. In *Handbook of mathematical models in computer vision*, pages 237–257. Springer, 2006.

[14] Brennan Gebotys, Alexander Wong, and David Clausi. Poof: Efficient goalie pose annotation using optical flow. In *9th International Conference on Sport Sciences Research and Technology Support, icSPORTS 2021*, pages 116–122, 01 2021.

[15] Brennan Gebotys, Alexander Wong, and David A Clausi. M2a: Motion aware attention for accurate video action recognition. *19th Conference on Robots and Vision (CRV)*, 2022.

[16] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The 'Something Something' Video Database for Learning and Evaluating Visual Common Sense. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-Octob:5843–5851, 2017.

[17] Bo He, Xitong Yang, Zuxuan Wu, Hao Chen, Ser-Nam Lim, and Abhinav Shrivastava. GTA: Global Temporal Attention for Video Action Understanding. arXiv preprint arXiv:2012.08510, 2020.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. ResNet. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.

[20] Stefan Hinterstoisser, Olivier Pauly, Hauke Heibel, Martina Marek, and Martin Bokeloh. An annotation saved is an annotation earned: Using fully synthetic training for object instance detection. *arXiv preprint arXiv:1902.09967*, 2019.

[21] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[22] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Fei Fei Li. Large-scale video classification with convolutional neural networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[23] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 2017.

[26] Wenbo Li, Zhicheng Wang, Binyi Yin, Qixiang Peng, Yuming Du, Tianzi Xiao, Gang Yu, Hongtao Lu, Yichen Wei, and Jian Sun. Rethinking on multi-stage networks for human pose estimation. *arXiv preprint arXiv:1901.00148*, 2019.

[27] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. TEA: Temporal Excitation and Aggregation for Action Recognition. *Proceedings of the*

IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 906–915, 2020.

[28] Ji Lin, Chuang Gan, Kuan Wang, and Song Han. TSM: Temporal Shift Module for Efficient Video Understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 7083–7093, 2020.

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

[30] Zhaoyang Liu, Donghao Luo, Yabiao Wang, Limin Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Tong Lu. Teinet: Towards an efficient architecture for video recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11669–11676, 2020.

[31] Alexander Mathis, Thomas Biasi, Steffen Schneider, Mert Yuksekgonul, Byron Rogers, Matthias Bethge, and Mackenzie W Mathis. Pretraining boosts out-of-domain robustness for pose estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1859–1868, 2021.

[32] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve Restricted Boltzmann machines. In *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*, 2010.

[33] Natalia Neverova, James Thewlis, Riza Alp Guler, Iasonas Kokkinos, and Andrea Vedaldi. Slim densepose: Thrifty learning from sparse annotations and motion cues. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10915–10923, 2019.

[34] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016.

[35] Tomas Pfister, James Charles, and Andrew Zisserman. Flowing convnets for human pose estimation in videos. In *Proceedings of the IEEE international Conference on Computer Vision*, pages 1913–1921, 2015.

[36] Javier Romero, Matthew Loper, and Michael J Black. Flowcap: 2d human pose from optical flow. In *German Conference on Pattern Recognition*, pages 412–423. Springer, 2015.

[37] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.

[38] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018.

[39] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *International Journal of Computer Vision*, 128(2):336–359, 2020.

[40] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in Neural Information Processing Systems*, 1(January):568–576, 2014.

[41] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, pages 402–419. Springer, 2020.

[42] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.

[43] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 Inter:4489–4497, 2015.

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[45] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. TDN: Temporal Difference Networks for Efficient Action Recognition. arXiv preprint arXiv:2012.10071, 2021.

[46] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc van Gool. Temporal segment networks: Towards good practices for deep action recognition. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9912 LNCS:20–36, 2016.

[47] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local Neural Networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

[48] Wayne Wu, Chen Qian, and Tong Lu. TAM: Temporal Adaptive Module For Video Recognition. arXiv preprint arXiv:2005.06803, 2021.

[49] Jia Xu, René Ranftl, and Vladlen Koltun. Accurate Optical Flow via Direct Cost Volume Processing. In *CVPR*, 2017.

[50] Dingwen Zhang, Guangyu Guo, Dong Huang, and Junwei Han. Poseflow: A deep motion representation for understanding human behaviors in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6762–6770, 2018.

[51] Zilong Zhong, Zhong Qiu Lin, Rene Bidart, Xiaodan Hu, Ibrahim Ben Daya, Zhifeng Li, Wei Shi Zheng, Jonathan Li, and Alexander Wong. Squeeze-And-Attention networks for semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 13062–13071, 2020.