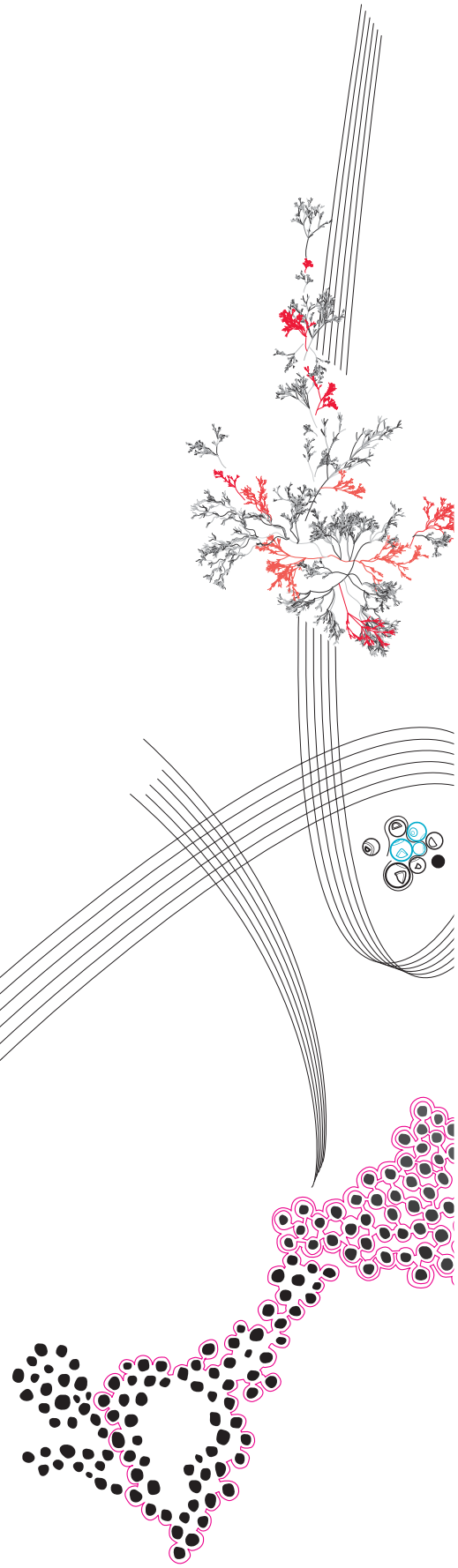

MSc Computer Science
Final Project

Explanation Guided Learning for Sports Video Data

Youri Tomassen

Supervisor: Alexia Briassouli, Tom van Dijk

June, 2025

Department of Computer Science
Faculty of Electrical Engineering,
Mathematics and Computer Science,
University of Twente

UNIVERSITY OF TWENTE.

Contents

Abstract

Deep Neural Networks achieve high accuracy in human activity recognition but often lack transparency, hindering trust in applications like automated sports officiating. Explainable AI (XAI) aims to elucidate model decisions, while Explanation Guided Learning (EGL) seeks to improve model intuition by incorporating explanations into training. This thesis investigates the efficacy of EGL for activity diving recogntion. We propose and evaluate novel EGL methodologies that leverage optical flow to automatically generate ground truth attention maps, addressing the common EGL challenge of reliance on manual annotations. Our methods include: 1) an Optical Flow Guided Learning (OGL) approach using a Dice loss to align model-generated GradCAM attention with optical flow-derived diver silhouettes; 2) an OGL approach that directly transforms input frames using these diver masks (Temporal Mask Transform); and 3) a "Right for the Right Reason" (RRR) loss guided by either GradCAM (RRR + GradCAM) or optical flow-derived attention maps to penalise misleading input gradients (RRR + OGL). These are implemented on a SlowFast network architecture. Results indicate varying effectiveness among EGL approaches. The Dice-based method underperformed, likely due to a fundamental mismatch between GradCAM's attention and binary segmentation masks. However, the Temporal Mask Transform method demonstrates 6.67% improvement at the lowest temporal resolution, and the RRR approach guided by optical flow (RRR + OGL) showed significant improvements, outperforming other augmentation methods at higher temporal resolutions (4.70% improvement). Experiments with denser temporal sampling in the SlowFast model's slow pathway challenged original architectural assumptions of the SlowFast architecture. Our combined approach performed better than prior work while using approximately 56.0% fewer computations. This research demonstrates EGL's potential for enhancing diving action recognition, underscores the viability of optical flow as a source for ground truth attention maps.

Keywords: Explanation-Guided-Learning, XAI, Action Recognition, Diving Video Data

Chapter 1

Introduction

Deep Neural Networks have achieved state-of-the-art accuracy in various machine learning tasks [34, 35], particularly in complex domains such as human activity recognition [7, 45, 15, 10]. These capabilities have enabled AI systems to tackle real-world applications, demonstrating performance that often matches or exceeds human expertise in specialized domains.

Building on these successes, AI is being integrated into sports officiating and judging systems. The evolution started with early statistical applications like the "Moneyball" approach [20] to real-time judging systems deployed at major competitions. Pioneering systems like Hawk-Eye [27] established the foundation for computer vision in sports officiating through precise ball-tracking technology for tennis line calls. Contemporary AI sports judging encompasses automated ball-strike systems in baseball [29], semi-automated offside technology in football [11], and judging support systems in gymnastics [22, 13]. Most recently, experimental AI judging has been introduced for snowboarding competitions at the X Games [4].

The deployment of AI in sports brings challenges that extend beyond traditional machine learning applications. Low-latency requirements are important in competitive sports, where decisions must be rendered in milliseconds to preserve game flow and competitive integrity. Current systems often "take too long to make accurate decisions or trade speed for accuracy" [50], with traditional VAR decisions averaging 70 seconds and causing significant disruptions to competition flow [11]. Perhaps most critically, the decision-making processes of these models often remain opaque [23, 34]. This lack of transparency presents a barrier to trust and adoption in sports officiating.

Explainable AI (XAI) has emerged to address this opacity, offering methods to generate insights into model predictions [39, 36, 41], often in the form of attention maps that highlight important regions in the input. These explanations allow humans to assess whether a model's reasoning aligns with domain-specific intuition for instance, in diving, focusing on the athlete's form and technique rather than irrelevant background elements, or in gymnastics, correctly identifying the critical body positions that determine scoring.

Building on XAI, Explanation Guided Learning (EGL) aims to proactively instill better "intuition" into models by incorporating explanations directly into the training process [34, 35, 40, 12, 14]. EGL frameworks typically augment traditional supervised learning by optimizing not only for correct predictions but also for appropriate attention or saliency maps [34, 35, 40, 12, 43, 14], often derived from human annotations or other guiding signals. This approach is particularly relevant for sports judging applications, where models must demonstrate accurate predictions but also reasoning that aligns with established judging criteria and human expertise.

And even though, state-of-the-art deep learning models excel on large-scale action recognition benchmarks such as Kinetics [7, 45, 24, 18]. They often struggle with tasks that involve complex, fast movements and subtle distinctions—precisely the type of analysis required in competitive diving, gymnastics, or assessing athletic form [52, 38, 21, 28]. XAI can help reveal the limitations of current models on these tasks, while EGL presents a potential pathway to enhance both their performance and interpretability for real-time sports applications. Nevertheless, EGL applications are constrained by the need for additional expert annotations, which are labor-intensive. This annotation burden may explain why there have been no works that apply EGL to video action recognition tasks, representing a significant research gap.

To bridge this gap, this study investigates the application and efficacy of EGL methodologies for sports activity recognition, with a specific focus on diving videos from the Diving48 dataset [21]. We explore novel approaches to EGL, including the use of optical flow to automatically generate ground truth attention maps, thereby addressing a common EGL challenge: the reliance on labor-intensive human annotations. This work introduces and evaluates several EGL strategies: one leveraging a Dice loss to align model-generated attention (via GradCAM [36]) with optical flow-derived diver masks, another utilizing these masks to directly transform input frames, and a third applying a "Right for the Right Reason" (RRR) loss [34] to penalize misleading input gradients, guided by either GradCAM or optical flow-based attention. The research aims to determine how these EGL techniques impact model performance on diving action recognition, the viability of optical flow as an attention guide, and the utility of GradCAM within such EGL frameworks. By introducing EGL to video action recognition and developing automated attention map generation methods, this approach addresses the scarcity of EGL applications in the video domain, and the annotation burden that limits the adoption of explanation-guided techniques.

The remainder of this thesis is structured as follows: Chapter 2 surveys fundamental concepts in action recognition, 3D CNNs, Explainable AI, and Explanation Guided Learning. The last two sections of Chapter 2 contain brief introductions to Optical Flow Moving Object segmentation and the Dice loss which are relevant to the methodology. Chapter 3 outlines the core research questions driving this investigation. Chapter 4 details the proposed methodologies, including dataset characteristics, model architecture, the novel EGL implementations using optical flow and GradCAM, and the evaluation strategy. Chapter 5 presents the experimental results, comparing the different EGL approaches against baseline models, analysing the impact of various hyperparameters, comparing with state-of-the-art, and discussing computational costs. Finally, Chapter 6 discusses the implications and limitations of these findings, and Chapter 7 concludes the thesis with a summary of contributions and potential directions and recommendations for future research.

Chapter 2

Related Works

Having established the importance of explanation-guided learning in sports video analysis, we now examine the foundational work and key developments across multiple relevant domains. This review begins with action recognition approaches before exploring more techniques in explainable AI, and the explanation-guided learning framework. Finally, we examine the fundamental concepts that underpin our proposed methodology.

2.1 Action Recognition

Understanding action recognition is fundamental to this work, as our explanation guided learning approach builds upon the foundations of video understanding to improve both performance in the diving activity recognition.

Action Recognition, also known as Human Action Recognition, is a computer vision task focused on identifying and classifying human activities in video data. The field encompasses varying levels of annotation granularity, with some benchmarks providing clip-level annotations [17, 21, 5, 6], while others offer more detailed segment-level annotations with multiple labels per clip [38, 52, 28].

The scope of these datasets ranges significantly. Large scale benchmarks like Kinetics contain a broad spectrum of activities, from everyday tasks such as cooking and cycling to sports activities like tennis [18]. In contrast, domain specific datasets such as FineGym, FineDiving, Diving48, and WorkoutForm QA focus on specific activities such as: gymnastics, diving, and weight lifting form, respectively. These specialized datasets often go beyond simple action classification by incorporating quality assessment metrics, including jury scores and detailed form deviation measurements [38, 52, 28], enabling more nuanced analysis of human performance and technique.

Contemporary deep learning models attain high accuracy on general action recognition benchmarks such as Kinetics [7, 45, 15, 10, 24]. However, these architectures struggle with fine-grained sports datasets that require understanding of complex movements and scoring criteria. Achieving comparable performance on specialized benchmarks such as FineDiving and FineGym required more sophisticated architectures with dedicated components for temporal modeling and motion parsing [38, 52, 53].

This work will focus on the Diving48 dataset [21], which contains 48 unique dive sequences. For the model to classify a sequence, it must sample the entire clip to capture the nuanced acrobatics of diving. This is in contrast to datasets like Kinetics, where activities such as biking can arguably be classified from a single frame. Diving48 only offers clip-level annotations, while FineDiving or FineGym also provide segment-level annotations. However, these fine-grained datasets are considerably smaller. In the interest of achieving

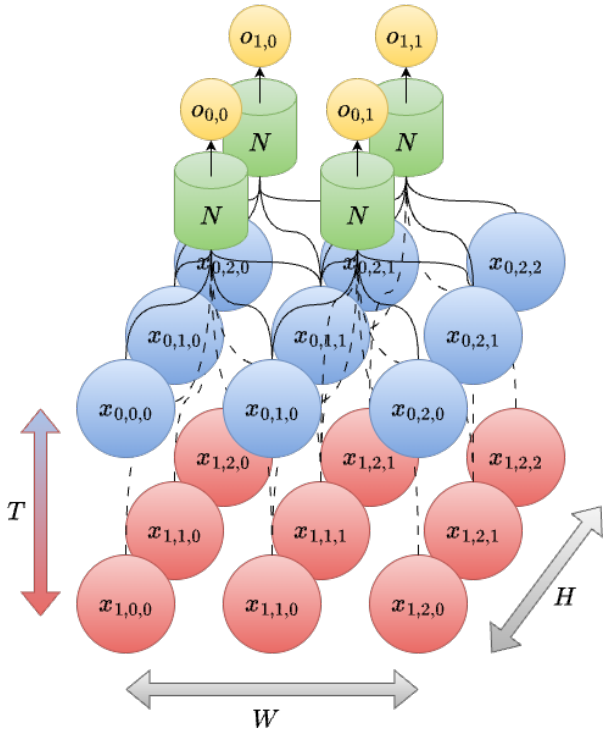

FIGURE 2.1: Conceptual representation of 3D CNN architecture showing hierarchical spatio-temporal feature extraction. This diagram illustrates a simplified cross-sectional view of a 3D convolutional neural network processing video data with spatial dimensions (Width × Height) and temporal dimension T. The blue (first frame) and red (second frame) nodes represent the input video volume with individual spatiotemporal locations $x_{t,h,w}$, green nodes show neurons N, and yellow nodes represent the intermediate feature representations O. The network demonstrates the fundamental principle of hierarchical feature learning, where low-level spatiotemporal patterns are progressively combined into higher-level semantic representations.

better generalizability, a larger dataset is preferred for this study.

Furthermore, one reason why EGL remains under-explored, especially in video action recognition, is that it is labour-intensive to create appropriate datasets containing ground truth attention masks. This work aims to bridge this gap by offering general methods for EGL that do not require human annotators.

Among the various architectural approaches for video understanding, 3D Convolutional Neural Networks have proven effective for temporal modeling in action recognition tasks.

2.2 3D CNNs

While action recognition encompasses various approaches, 3D Convolutional Neural Networks have emerged as an effective architectures for processing video data. These networks extend traditional 2D convolutions into the temporal dimension, enabling them to capture both spatial and temporal features simultaneously.

The ImageNet-1k benchmark, comprising over 1.2 million training images across 1,000 categories [9], posed a significant challenge to early computer vision models. Despite nu-

merous attempts, meaningful classification accuracy remained elusive until the invention of deep Convolutional Neural Networks (CNNs). This architectural breakthrough enabled 2D CNNs to revolutionize image recognition through their ability to learn hierarchical spatial features. These networks leverage pooling layers and strided convolutions to systematically expand their receptive field—the spatial extent of input pixels that influence a neural unit's activation [31]. This architectural principle of increasing receptive fields proved valuable beyond spatial analysis, finding applications in temporal modeling where long-range dependencies between data points needed to be captured [47]. This natural progression led to the development of 3D CNNs for action recognition, where convolutions operate not only across spatial dimensions but also along the temporal axis of sequential RGB frames. 3D CNNs have been widely adopted for action recognition tasks [7, 45, 15, 10] and have dominated state-of-the-art performance for several years.

Figure 2.1 shows a conceptual representation of neurons N processing a 3D video volume $x_{t,w,h}$[1], with the first frame in blue and the second frame in red. Each neuron extracts features within its own local 2×2×2 volume, where the outputs O become inputs to higher-order neurons. In general, lower-order neurons learn to recognise edges, and other primitive shapes of different orientations. In higher-order neurons, these lower-order features are combined into higher-level features representing more complex shapes.

Furthermore, at each convolutional layer, the effective receptive field of the neuron increases. If we imagine another neuron that connects to all the outputs O, this neuron's receptive field would span the entire 3×3×3 volume (if we also extend our input temporally). These concepts are what make CNNs so effective for image recognition, and why this logic can also be extended to video action recognition. By stacking layers of neurons that hierarchically process inputs, we eventually reach a sufficient receptive field that spans the image size or video sequence. However, there are limitations to the size of the receptive field, since most video recognition models are trained on 16-128×224×224 ($T \times H \times W$) inputs [7, 10].

Even though the theoretical receptive field can grow to any size by adding more layers, this too has limitations. In [26], the authors show that convolutions suffer from locality bias by demonstrating that the true receptive field was much smaller than the theoretical receptive field for 2D CNNs. This is probably why many papers use lower temporal resolutions of 16, 64, and 128 frames [7, 45, 10]. Adding more layers also has limitations in its own right, deeper networks suffer from vanishing gradients [31], therefore, its not possible to infinitely stack layers. Furthermore, there are computational constraints as 3D CNNs are 8 times more expensive to compute than 2D CNNs.

For this work, we must acknowledge the existence of vision transformer architectures; a superior architecture [3, 24, 49]. However, we opted to use 3D CNNs since they provide smaller models that are computationally cheaper. This choice aligns with the experimental nature of this work, as smaller models allow for more experimentation within our restricted compute budget. Finally, this work emphasizes exploring better methods for training action recognition models rather than breaking benchmark scores, since these methods are general and can be applied to stronger and larger architectures if they prove beneficial.

Therefore, this work focuses on 3D CNNs due to their more favourable computational requirements. In particular, we use the SlowFast architecture, a 3D CNN designed to mimic the human retinal system to achieve state-of-the-art performance that competes with newer vision transformers [10, 3, 49].

[1] In reality, the input for RGB video contains an additional channel dimension c. However, for clarity, the channel dimension was omitted for clarity.

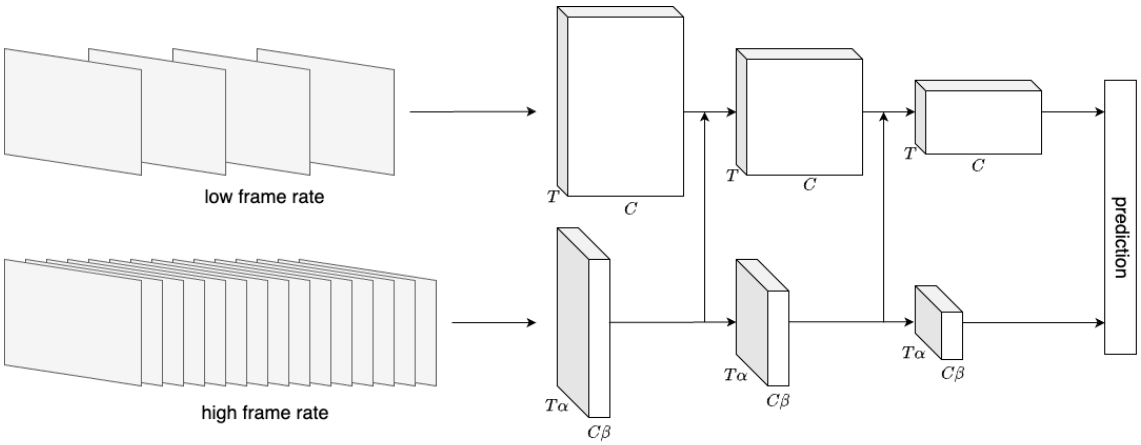

FIGURE 2.2: Diagram of the SlowFast network's principle behind the slow (top) and fast (bottom) pathways. The slow path uses a lower frame rate T, but has higher channel depth C, suitable for processing stable scene elements such as textures, colours, and lighting conditions. Motion changes often occur faster than changes in the identity of the subject performing those motions. Therefore, perceiving motion effectively requires higher frame rates $T\alpha$, and lower channel depth $C\beta$ ($\beta = \frac{1}{8}$). Cross-connections after each block fuse the temporal features with the spatial features.

2.2.1 SlowFast

The SlowFast architecture is inspired by the human retinal system. The authors observed that the retina uses different cell types (M- and P-cells) operating at varying spatial and temporal resolutions, with P-cells emphasising spatial information and M-cells capturing higher-fidelity temporal data. Motion changes often occur faster than changes in the identity of the subject performing those motions. Therefore, perceiving motion effectively requires higher frame rates $T\alpha$. For example, consider a diver performing a somersault from a diving board: the diver's skin colour and swimsuit that determine their identity in the scene remain unchanged, while the rapid motion changes from the somersault require high temporal resolution to capture accurately.

Accordingly, the SlowFast model (Figure 2.2) has separate spatial and temporal convolution pathways. The spatial convolutions run at high spatial resolution but apply strong temporal stride, while the temporal convolutions process frames at a lower stride. Similarly, the spatial/slow pathway's convolutions have higher channel depth, allowing this path to capture more complex features. Cross-connections after each block fuse the temporal features with the spatial features. This design allows the temporal pathway to operate at only 20% of the total compute, as fewer channels are required compared to jointly processing the spatial and temporal dimensions. The SlowFast architecture has shown improved performance on video action recognition tasks [10].

The relative temporal stride and channel depth are controlled by the parameters α and β respectively. Therefore, if the base temporal resolution is T, then the slow path processes T frames, while the fast pathway processes $T\alpha$ frames. In similar fashion, β controls the channel depth: for any layer C_i in the slow pathway, the fast pathway will have a channel depth of $C_i\beta$, where β is a fraction (for example $\frac{1}{8}$). In [10], the authors show that on the Kinetics benchmark they achieve the best performance with an α of 8. The present work challenges this claim as we found models to perform better at $\alpha = 4$. Lowering α increases

the temporal resolution, which we theorise allows the model to process temporally complex motion patterns within the slow pathway's higher channel depth convolutions.

While 3D CNNs provide an effective foundation for video understanding, their complex architectures make it difficult to determine whether they learn appropriate features or rely on spurious correlations a critical concern that explainable AI methods can address and that forms the foundation for explanation guided learning approaches that actively steer models toward learning the right features during training.

2.3 Explainable AI

For a simple model, such as a linear model, the best explanation is the model itself [25]. However, for complex models, the original model is often too complex to serve as an explanation. Therefore, we need to resort to using simpler explanation models as proxies. XAI methods help create simpler, more interpretable models from complex ones.

Furthermore, while supervised deep learning exposes metrics like loss and accuracy, these are insufficient, as models may memorize the training dataset, leading to poor generalizability [23]. Moreover, models have been shown to take shortcuts, relying on confounding information in the data rather than genuine features for classification [40, 34, 12]. Confounders in a dataset are regions of the image the model uses to classify some input that are often unrelated to the class itself [37, 12]. This highlights the need for tools that help humans understand how deep learning models make decisions. The research field dedicated to addressing these issues is referred to as Explainable AI (XAI).

XAI can help clarify model decisions through simplified explanation models. This is crucial in domains where trust and transparency are vital, such as healthcare, finance, and autonomous systems. These explanations also promote broader user adoption by making AI models more interpretable [25]. By uncovering how decisions are made, XAI can also help identify and mitigate bias, ensuring models do not perpetuate societal inequities.

Furthermore, XAI assists in debugging models by revealing weaknesses, such as overreliance on irrelevant features or 'shortcuts'. Correcting these shortcuts can significantly improve model performance by reducing overfitting and increasing generalizability [34, 40, 43, 12]. For example, in activity recognition models, identifying misleading features that contribute little to classification can help improve the model's robustness to unseen data.

Though XAI can provide insight into model behaviour, these explanations are only simplified representations of what occurs inside the model. Models can still rely on confounding features that are not identifiable through XAI alone [40]. Therefore, it is important to exercise caution when utilizing XAI explanations. Additionally, XAI methods can be computationally expensive. For example, methods like SHAP require multiple passes through the model to explain a single example [25, 47, 43].

2.3.1 Explainable AI Methods

A common way to interpret image classifiers, or action recognition models, is through saliency maps, or attention maps, which are used interchangeably. There are different ways to obtain attention maps. A common method is Gradient Class Activation Map, (GradCAM) [36]. GradCAM weights model's activations in the final convolutional layer by the gradients with respect to the final layer [36]. This can be loosely interpreted as where - and in case of action recognition, also *when* - the model was looks. And how much it influenced the classification of a particular class. Figure 2.3 shows a series of frames from the GradCAM output from a person performing diving acrobatics. In this figure you see

that the heatmap is changing in colour and intensity. With blue representing very little attention, and red very high attention. As the dive progresses, the colour shifts from green, to yellow to red. This visualisation shows the model paid most attention to the middle 4 frames, and at the position of the diver. The fifth frame has the highest attention values as the divers are in mid somersault. This GradCAM visualisation closely aligns with 'our own' intuition on when and where models should look.

FIGURE 2.3: This figure shows a series of frames from the GradCAM output for two people performing diving acrobatics. The attention map is displayed using a colour map that ranges from blue-red; red meaning more *attention*. The attention-maps changes over time, and the last frame shows a curious artifact in the bottom right corner. These images where generated using an SlowFast network network [10], trained on the Diving48 dataset [21]. The GradCAM implementation from [36]

Local Interpretable Model-agnostic Explanations (LIME) on the other hand, is an model agnostic model explanation method [39, 34]. LIME focuses on explaining individual predictions. It approximates the behaviour of the complex model by learning a simpler model. The simpler model is fitted with slightly perturbed inputs. LIME runs these perturbed samples through the model. Then a simple model (linear regression) is fitted with the inputs of the model and the outputs of the model. This simple model can then show which features had the most influence on the original prediction[39, 34]. Generating new examples, and fitting linear model to each training example can be computationally expensive, especially if you want explain the entire training set[34].

SHapley Additive exPlanations (SHAP) is another model agnostic method [43, 47]. It is a unified framework for interpreting model predictions by attributing a score to each feature, representing its contribution to the final outcome. SHAP measures a feature's importance by evaluating how the prediction changes when the feature is included or excluded [43]. Shapley values are calculated by averaging the marginal contributions of a feature across all possible subsets of features, ensuring a fair and consistent allocation of importance. If excluding a feature leads to a large change in the prediction, the feature has a relatively large impact; conversely, if there is little change, the feature has a minimal effect on the model's decision.

SHAP can be applied to any input domain, including images, text, and numeric data

[43]. In the image domain, SHAP perturbs regions of pixels (e.g., masking certain areas) and analyses the model's output changes to determine the importance of different pixel regions. This results in an importance map, similar to methods like GradCAM and LIME. Unlike GradCAM, SHAP can identify which features reduce accuracy on the final output [47]. This makes SHAP quite versatile, as it can pinpoint features that boost a prediction but also those that reduce it.

However, SHAP is computationally expensive, as it requires multiple forward passes through the network to assess each feature's contribution, which grows exponentially with the number of features.

2.3.2 Explainable AI for 3D CNNs

Explainable AI for 3D CNNs is a relatively under-explored area of research. Earlier methods adopted saliency maps and GradCAM on streams of images, but they often overlooked the contribution of the temporal kernel in 3D convolutions, although it has been shown that motion plays a small yet significant role in classifying activities [7]. Recent works have incorporated the temporal axis in 3D convolutions to better explain the contribution of motion to activity classification.

These are some common methods to give an explanation for why a model classified a certain samples. All of the examples fall under the umbrella of attention maps. These methods are sometimes also referred to as saliency maps, or activation.

One notable approach is Saliency Tubes, created by [41], a gradient-based activation method that maps the activations from the final convolutional layer back to the input, highlighting the regions that most influenced the model's decision. This method is similar to GradCAM but adds temporal filters to the activation map. Due to the high dimensionality of the convolutional layer ($T \times H \times W \times D$), where T is the temporal axis, H and W are the spatial axes, and D are the channels, the authors applied a threshold (τ) to filter out low activations that contribute little to the final classification. By adopting gradient-based filtering, they were able to quantify the contribution of motion to classification.

In another work, [16] proposed a selective relevance map, which uses derivative-based filtering to remove low or near-constant activations in saliency maps. This technique results in a map that only exposes activations corresponding to motion, thus helping to isolate regions of interest based on movement.

In addition to saliency-based approaches, occlusion-based methods provide another way to identify which parts of the model contribute to its decisions. A common technique is to occlude the object of interest and observe whether this changes the output. [46] developed such a method for RGB video streams. One challenge with video inputs is that objects are often in motion, so occlusion creates a "tube" that tracks the object over time. To address this, they tracked the object of interest and adjusted the position of the occluded region using optical flow. Optical flow quantifies the motion between two frames through motion vectors, and these vectors were used to adjust the occlusion in each frame.

Price et al. (2021) [30] employed SHAP (SHapley Additive explanations) to quantify the temporal contribution of individual frames to the model's decision-making process. Their methodology generated frame-wise SHAPley values, revealing the temporal dynamics of positive and negative influences throughout the video sequence. This temporal attribution analysis provided insights into which segments of the activity most significantly influenced the model's final classification.

While XAI methods help us understand model decisions post-hoc, a more proactive approach involves incorporating explanations directly into the learning process. This leads us to Explanation-Guided Learning (EGL), which leverages explanations during model train-

ing to improve both performance and interpretability. Furthermore, explanation guided learning remains under explored for action recognition tasks, making it essential to understand the motivation behind this approach and its general framework before examining specific applications to video understanding.

2.4 Explanation-Guided Learning

When a person is asked to classify an animal, the person will likely look at the colour or texture of the fur, the shape of its ears, or any other distinguishing features. Where the surroundings play a minimal role; is there grass or snow? In other words our decision should be invariant to the environment. We want our deep learning models to have this same intuition. When training data is limited in variety, models tend use 'clever tricks' in order to beat the accuracy high score [40]. For example, it could deduce that snow correlates with polar bears. Explanation-Guided Learning (EGL) addresses this limitation by incorporating additional objectives into the model's loss function, ensuring that the model jointly optimises its predictions and explanations, which in turn improves generalisability [14]. And prevent models from relying on confounding factors [12].

The algorithm operates as follows: A model is trained on a dataset consisting of inputs x with corresponding labels y and explanations A. These explanations A are included as part of the training set, and are often created by a human annotator. During the training process, the model generates its own explanations $\hat{A}$, which can be produced using methods such as LIME or GradCAM [12].

There are typically two approaches for incorporating explanations into the training pipeline: 1) augmenting the loss function by adding a term that measures the distance between the ground truth explanation A and the generated explanation $\hat{A}$, and 2) augmenting the data by masking undesired regions of the image [12].

In [12] the authors showed that EGL prevents models from relying on confounding factors. Yet, they also showed that some of the models they tested where insensitive to the quality of the annotation. Meaning the model improved regardless of the annotation quality. This lack of sensitivity to feedback quality raises questions about the reliability of EGL across different model architectures. Intuitively, we would expect a model's performance to degrade with incomplete feedback and improve with comprehensive feedback.

2.4.1 The Explanation-Guided Learning Framework

This section examines the Explanation Guided Learning Framework introduced by [14], which provides a theoretical foundation for incorporating explanation-based learning into contemporary deep learning architectures. Understanding this framework is crucial as it establishes the mathematical and conceptual groundwork for augmenting traditional supervised learning with explanation guidance.

Gao et al. (2024) [14] establish a foundation for Explanation Guided Learning through a framework. Their formulation, expressed in Eq 2.1, captures the essential components and interactions within EGL systems. The architectural implementation of this framework is illustrated in Figure 2.4, which provides a visualization of the mathematical relationships and data flow described by Eq 2.1.

In the EGL framework, X, Y, and A are the respective inputs, labels, and annotations (often a saliency map). $f(X, Y)$ represents the deep learning model, and $\mathcal{L}_{\text{pred}}$ denotes the loss with respect to the model's predictions. This first term represents a standard machine learning objective. The second term, starting with *alpha*, introduces $g(f(X, Y)$, where g

is an *explainer* that generates an explanation $\hat{A}$, often a saliency map, from the model $f(X, Y)$. The ground truth explanation, A.

$\mathcal{L}_{\text{exp}}$ is the explanation loss, and Ω is the regularisation term that tracks an inherent property of the explanation. α and β control the influence of these respective terms. The explanation loss $\mathcal{L}_{\text{exp}}$ is an additional signal to help model identify important cues. An example of this could be a human-created attention map $\hat{A}$, where $\mathcal{L}_{\text{exp}}(g(f(X, Y), \hat{A}))$ has to maximise the overlapping area between the generated explanation $\hat{A}$ and the A. Thereby, steering the model towards a set of parameters that look at patterns in the area of the attention map.

$$\min \mathcal{L}_{\text{pred}}(f(X, Y)) + \alpha \mathcal{L}_{\text{exp}}(g(f(X, Y), A)) + \beta \Omega(g(f(X, Y))) \tag{2.1}$$

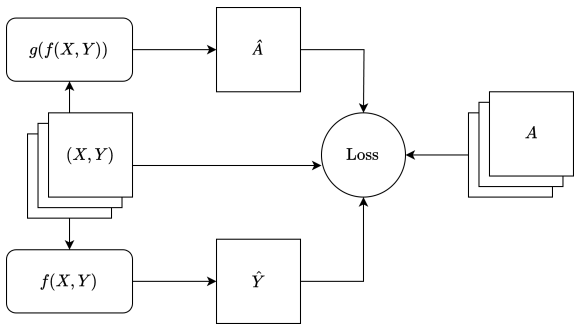

FIGURE 2.4: X, Y are the labeled dataset and serve as an input to the model $f(X, Y)$. The model produces $\hat{Y}$ as an output. For each X, Y the explainer $g(f(X, Y))$ generates an explanation $\hat{A}$ for the model $f(X, Y)$. Finally, $\hat{Y}, Y, \hat{A}$, and A serve as input to the **Loss**. Here A is a human generated explanation, often an attribution map.

In summary, EGL incorporates human-provided explanations into the model's objective function with the goal of achieving better test performance and improved generalisability. These explanations, often derived from human annotations, ensure that the model's decisions align with interpretable features. However, EGL faces a major challenge due to the reliance on handcrafted annotations, which can be difficult to scale and may introduce subjectivity [34, 35, 14]. Thus implementers should be wary when building EGL models as it may yet introduce additional biases.

2.4.2 Examples of Explanation-Guided Learning

In [43], the authors describe two methods for using model explanations to aid learning. Both methods utilize SHAP to identify the most significant features for classifying the current sample. The first method employs SHAP to mask areas that contribute to misclassification. The assumption here is that a model's incorrect prediction is due to an incorrect explanation. Therefore, by using the explanation to mask those areas and retraining the model, the model can learn the correct features. This approach is based on two assumptions: first, that the label y is correct for the sample x, and second, that the model's explanation accurately reflects the features it used to classify the sample x. This method can be viewed as a form of feature selection, as masking part of the input is effectively selecting which parts of the image to use. The second method uses SHAP to reweigh specific examples in the loss function, creating a weighted loss function $w_i \cdot \mathcal{L}$. Here, $\mathcal{L}$

represents the loss, and w_i is the weight assigned to a specific example. The reasoning is that when a sample x_i is misclassified, penalising the model with w_i will cause the model's loss to react more strongly during back propagation. This gives the model a chance to find a new parameter set that better the maps x_i to y_i.

In Ross et al. (2017) [34], the authors propose a method for explaining and regularising deep neural networks by selectively penalising input gradients through an additional regularisation term in the loss function. This regularisation term is based on annotations $A \in 0, 1$. This regularisation term imposes a soft inductive bias on regions marked by A. Discouraging large gradients in annotated regions, while allowing the model flexibility to use these regions as needed. The authors did not rely on human annotations but instead used an approach that iteratively modifies the annotations to generate a spectrum of models. These models are evaluated to determine which best aligns with the intended behaviour. In a follow-up paper, the authors improved their approach by using human annotations, and GradCAM instead of LIME [35].

Selvaraju et al. (2019) [37] addresses a model's reliance on language priors for visual question and answering tasks. Their research highlighted a critical issue where models exhibited bias towards common linguistic associations rather than actual visual evidence for instance, automatically classifying bananas as yellow despite an image clearly depicting an *green*, unripe banana. This demonstrated how models *often* default to statistical priors rather than processing visual information accurately. Their approach enhanced the loss function's sensitivity to specific image regions through human-guided attention mapping, leading to improvements in visual question answering performance [37]. However, while their method reduced the reliance on language priors, they were not able to completely eliminate these biases [37].

All previous methods used input gradients or model estimates to propagate back to the input space as explanations [40]. However, these methods do not reveal the internal concepts the model uses to make its final decision. In "Right for the Right Concept" [40], the authors developed a neural symbolic concept learner that jointly optimises for both visual and symbolic input. To this end, they introduced a new dataset containing information such as colour and shapes. They found that the model indeed relied confounding information on concepts that are not identifiable through visual explanations alone. By providing feedback at the semantic level, they were able to improve the model's performance and generalisability.

This concludes our introduction to explanation-guided learning. As previously stated, explanation-guided learning faces major challenges as it relies on additional hand-crafted annotations. Other works address this limitation by omitting ground truth annotations altogether. However, none have attempted to address this limitation directly by automating the creation of ground truth attention maps, which is something this work aims to explore in addition to methods that do not use ground truth annotations. Lastly, to our knowledge none of previous works have applied EGL to the video action recognition task.

Having established the theoretical foundations of explanation-guided learning and its limitations, it becomes crucial to examine the datasets available for sports action recognition tasks. The choice of dataset significantly impacts both the feasibility of applying EGL methods and the potential for developing automated annotation approaches, particularly in the context of action understanding.

2.5 Datasets

Kinetics stands as one of the most comprehensive human action recognition datasets in the field [18]. The original version, Kinetics-400, contains approximately 300,000 videos spanning 400 distinct action labels, with each clip lasting approximately 10 seconds [18]. The dataset encompasses a diverse range of human activities and interactions, from everyday actions like bike riding and handshaking to sports activities such as tennis. Following the success and widespread adoption of the original dataset, the authors released two significant expansions. Kinetics-600 introduced an additional 200 action categories [5], followed by Kinetics-700 [6], which further expanded the label space. These updates not only increased the number of action categories but also implemented improved data collection processes, enhancing the overall quality and reliability of the dataset. The largest critique on this dataset, that for many classes only few frames are needed from the sequence to classify the activity.

Diving48 is a competitive diving dataset containing approximately 18,000 videos with 48 unique dive sequences. Figure 2.5 illustrates all the unique dive sequences present in the Diving48 dataset. Each sequence is defined by three components: 1) Takeoff: The initial moment when the athlete launches from the diving board. 2) Flight: The aerial phase where the athlete performs complex maneuvers including somersaults and twists. 3) Entry: The final phase where the diver enters the water in a specific position

While these components are fundamental to each dive sequence, it's important to note that Diving48 only provides coarse-grained labels, meaning individual components within the dive sequence are not separately annotated. The Diving48 dataset was designed to train action recognition models in the absence of confounding information. This design choice was motivated by the dataset's uniform background across all videos. Unlike other action recognition datasets where background elements might inadvertently influence model predictions, Diving48's consistent setting forces models to focus exclusively on the diver's movements and form for sequence classification, eliminating potential noise from varying backgrounds.

WorkoutForm QA is another AQA datase, but focused on the form of different weight lifting exercises [28].

All datasets discussed thus far have coarse-grained annotations, where each clip has a single label. In contrast, fine-grained action recognition datasets can have multiple labels per clip, with each action demarcated by specific start and end frames. A notable example is FineGym, a fine-grained action recognition dataset in the domain of gymnastics that provides temporal annotations at both the action and sub-action level. FineGym implements a three-level semantic hierarchy [38]. At the coarsest level, it describes *events* within gymnastics, where each event consists of one or more *sets*, and each set is described by multiple *elements*. This hierarchical structure was specifically designed to investigate action recognition models' ability to learn and understand complex sequential patterns. Notably, the authors demonstrated that state-of-the-art models which performed well on coarse-grained datasets like Kinetics struggled to achieve comparable performance on FineGym, highlighting the increased complexity of fine-grained action recognition [38]. Building upon their work with FineGym, the same authors developed the FineDiving dataset [52]. While FineDiving maintains a similar approach to fine-grained temporal annotations, it introduces an important additional feature: jury scores. This inclusion enables models to not only recognize actions but also optimize for quality assessment [52], making it particularly valuable for developing comprehensive sports analysis systems.

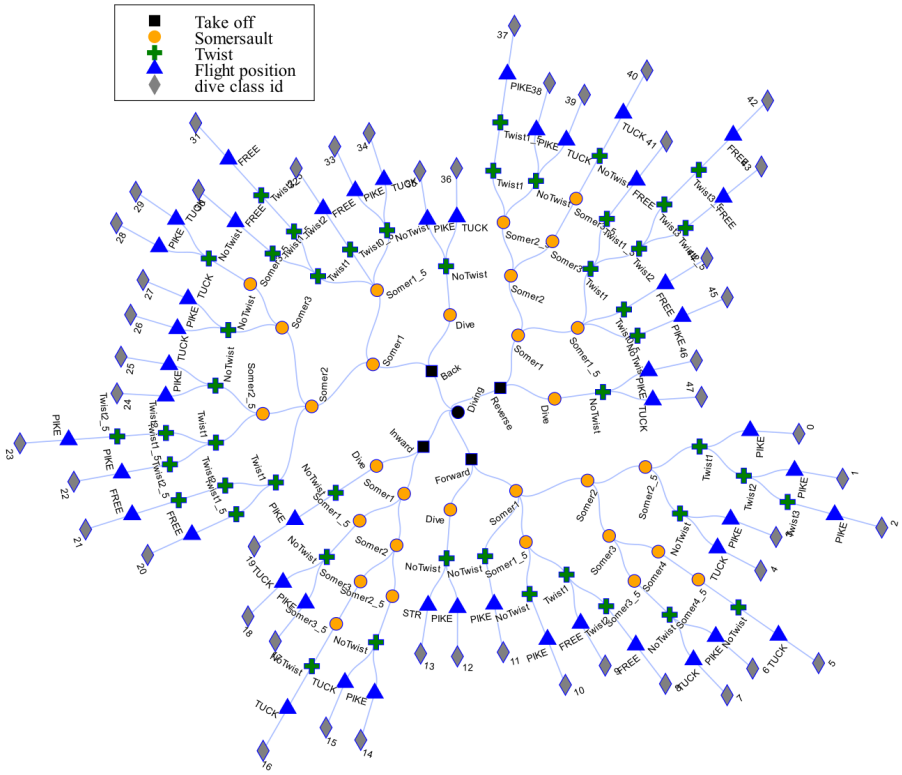

FIGURE 2.5: This diagram (taken from Li et al. [21]) shows all the possible dive sequences in the Diving48 dataset. Each dive is identified by a starting position, followed by a somersault(s) and twist(s), and finishing with a flight position. The final node (diamond) shows the assigned class identifier.

2.6 Optical Flow Motion Segmentation

Previous works such as I3D [7] leverage optical flow frames in a dual-path architecture, in which two I3D models are trained. One is trained on RGB video and the second is trained on optical flow frames. In [7] the authors should that including optical frames enhanced performance, demonstrating the inherent value of optical flow. In contrast, this work aims to leverage optical flow frames to perform moving object segmentation—optical flow motion segmentation. The premise of optical flow motion segmentation is to automatically separate video frames into different groups, where each group contains pixels or regions that belong to objects moving in the same way [1]. Such segments appear to be good truth attention masks, which is relevant to the present work.

This work leverages FlowSAM for optical flow motion segmentation, which builds upon the work of Segment Anything (SAM) by prompting through optical flow frames for moving object segmentation [19, 51]. SAM segments a target image's scene through a prompt encoder. In the original work, the SAM prompt encoder was trained on points and bounding boxes. Thereby, you could point to an object in the scene and that object would then be segmented from the background.

FlowSAM extended the SAM model by training an additional prompt encoder for optical flow frames [51], thereby allowing moving object segmentation using optical flow frames. FlowSAM operates in three modes: 1) FlowI-SAM uses optical flow frames for moving object segmentation; 2) FlowP-SAM combines RGB and optical flow; and 3) FlowT-SAM uses optical flow prompts to temporally match segments in a sequence. This solves temporal

continuity errors in sequences of segments.

By leveraging FlowSAM's ability to automatically segment moving objects from optical flow frames, this work addresses a critical limitation in explanation-guided learning: the need for manually annotated ground truth attention masks. The segmented diver masks generated through optical flow motion segmentation provide an automated alternative to human annotation, enabling the application of EGL methods to sports action recognition without the labour-intensive process of creating hand-crafted explanations.

2.7 Dice Loss

This loss function is typically applied in the context of medical image segmentation. The Dice loss excels at segmenting anatomical structures, tumors, organs, and lesions in medical images like MRI, CT scans, and X-rays [42, 54]. It's especially valuable because medical segmentation often involves highly imbalanced datasets where the target region is much smaller than the background [42, 54]. Unlike cross-entropy loss, Dice loss naturally handles imbalanced datasets without requiring manual weight adjustments [42, 54]. This makes it ideal for scenarios where the positive class represents a small fraction of pixels. For example, a dataset where positive pixels represent only 1% of the image, cross-entropy loss is dominated by the 99% background pixels. Even if the model predicts all pixels as background, it achieves 99% accuracy. Dice loss, however, would be 0 because there's no intersection between predicted and true positive regions. Furthermore, Dice loss is sometimes used alongside other losses to improve mask quality and boundary precision [54, 2].

The Dice Loss, as defined by Eq 2.2, takes the intersection between the predicted area P and ground truth area T and scales it by the sum of these respective areas. The Dice coefficient $\text{Dice}(P,T) = \frac{2|P \cap T|}{|P|+|T|}$ produces a value in the range $[0, 1]$, with 1.0 being a perfect match and 0 indicating no overlap. To use this as a loss function for training deep neural networks, the Dice loss is computed as $\text{DiceLoss} = 1 - \text{Dice}(P,T)$, ensuring that lower loss equals better performance.

$$\mathcal{L}_{\text{Dice}}(P,T) = 1 - \frac{2|P \cap T|}{|P| + |T|} \tag{2.2}$$

As discussed, Dice loss is widely used in medical image segmentation due to its effectiveness with imbalanced datasets. This work adapts Dice loss for attention alignment within the EGL framework, which is well-suited to our application. The positive masks representing our divers occupy significantly fewer pixels than the background, creating the class imbalance scenario where Dice loss excels. Additionally, Dice loss has been combined with other loss functions in previous work, making it an ideal component for our multi-objective optimization approach.

Based on the review of relevant literature and available datasets, we list several gaps in the present research. Namely, the application of EGL on video action recognition tasks, and lack of datasets that contain ground truth saliency maps. The following research questions address gaps in current approaches to sports action recognition, with particular focus on integrating EGL with diving video analysis.

Chapter 3

Research Questions

Studies have shown that EGL possesses the ability to revise model behaviour when it uses confounding features [12, 35, 34, 40, 43]. To the best of our knowledge, none have investigated action recognition datasets such as Diving48 in the context of EGL. Furthermore, there exists a significant gap in addressing the central limitation in EGL, which is the lack of ground truth saliency maps. The present study aims to address these gaps with following research questions:

1. **Q1:** *How effectively does Explanation Guided Learning improve performance on fine-grained action recognition tasks?* Most existing EGL studies focus on image classification. This question directly evaluates EGL's transferability to video action recognition.

2. **Q2:** *Can optical flow-derived segmentation masks serve as effective ground truth explanations for video-based EGL without human annotation?* This addresses the scalability challenge of EGL by investigating whether automatically generated masks from optical flow can replace costly human-annotated explanations while maintaining learning benefits.

3. **Q3:** *To what extent does constraining model attention to align with ground truth diver segmentation masks improve both classification accuracy and explanation quality?* This examines whether attention alignment approaches can simultaneously optimise for correct predictions and meaningful attention patterns that correspond to the actual subject of interest in the domain of action recognition.

4. **Q4:** *How do different explanation guidance approaches (attention alignment vs. input masking vs. gradient penalisation) compare in their effectiveness for fine-grained action recognition?* This evaluates the relative merits of three distinct EGL strategies: Dice loss for attention alignment, direct input transformation through masking, and Right for the Right Reason gradient penalisation, to determine which approach best leverages optical flow-derived explanations.

To address these research questions systematically, we propose a methodology that combines multiple approaches. This methodology builds upon existing work while introducing novel techniques specifically designed for diving video classification.

Chapter 4

Methodology

This section outlines the proposed methodology for improving action recognition models through Explanation-Guided Learning (EGL) on the Diving48 benchmark.

This approach consists of three main phases: first, establishing a baseline action recognition model; second, implementing and evaluating two distinct EGL methods; and third, conducting performance analysis.

The methodology is structured as follows:

- Section 4.1 presents a detailed analysis of the Diving48 dataset [21], our chosen benchmark for competitive diving action recognition
- Section 4.2 describes the architecture and implementation details of our deep learning model
- Sections 4.3 and 4.4 introduce four novel EGL methods designed to enhance model performance and interpretability
- Section 4.5 details our evaluation strategy, which assesses both model performance, and attention map quality
- Section 4.6 describes the data augmentation and pre-processing pipeline.

4.1 Dataset

The model will be trained and evaluated on the Diving48 dataset [21], which presents unique challenges in action recognition due to its complex sequential diving manoeuvrers. Each dive classification requires the model to comprehend and analyse an intricate sequence of movements, making it an ideal benchmark for testing sophisticated action recognition capabilities. This work makes use of the second version of the Diving48 dataset which exclude some samples that have been miss labelled, this is also the reason class 30 is missing in 4.1.

Diving48 offers several distinct advantages over other prominent action recognition datasets such as Kinetics-400:

- **Short Duration**: Diving48 clips are short, around 4 seconds, in contrast to the significantly longer sequences found in Kinetics-400 and FineGym [18, 38]. This brevity facilitates more efficient training cycles, particularly valuable given computational resource constraints.

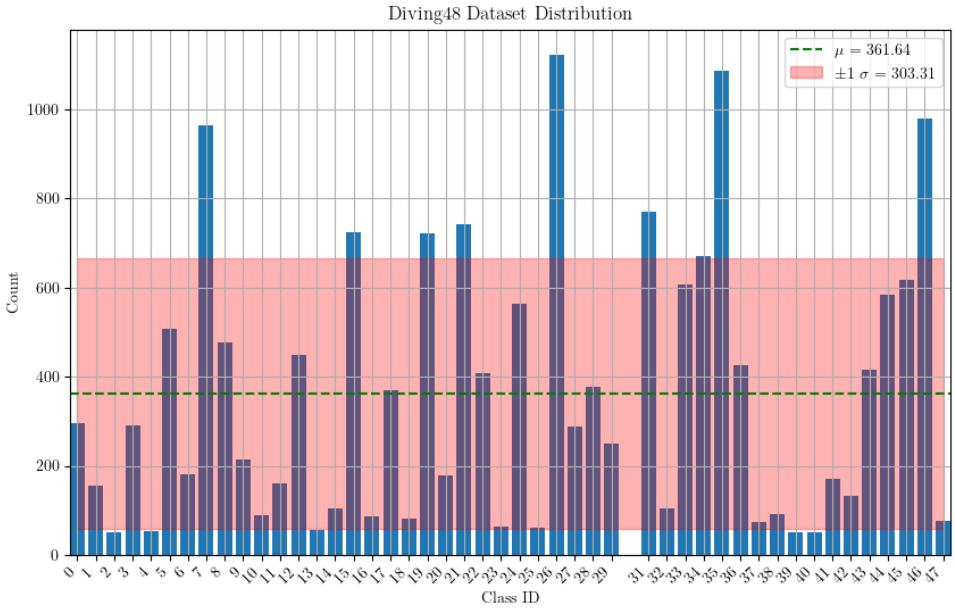

FIGURE 4.1: This figure shows the distribution for the class labels of the Diving48 dataset [52]. The plot includes the mean class label count ($\mu = 361.64$) and $\pm$ 1 standard deviation ($\sigma = 303.31$). The distribution for the Diving 48 dataset is fairly unbalanced, as indicated by the variance σ. This means we should investigate whether the model can perform better with a balanced dataset compared to the default - the unbalanced dataset.

- **Sequential Complexity**: The sport of diving inherently requires the model to analyze complete action sequences for accurate classification. Unlike simpler actions in other datasets (such as waving, bike riding, or swimming) that can often be identified from a single frame, diving classifications demand comprehensive temporal understanding.

The Diving48 dataset suffers from a class in balance. Figure 4.1 shows this class in balance clearly. Considering that the entire dataset contains approximately 16K samples, then the top 4 classes make up almost half the dataset. Furthermore, we showed that there is a strong correlation between the number of samples and the individual class accuracy as shown in A.1.

4.2 Model

The SlowFast-50D network [10] serves as our primary model for action recognition. This architecture processes input videos represented as tensors $X \in \mathbb{R}^{C \times T \times H \times W}$, where $C = 3$ represents RGB channels, T denotes the temporal resolution (number of frames), and $H \times W$ specifies spatial dimensions (typically 224×224 for training and 256×256 for testing). The model function $f(X, \theta)$ maps these input tensors to class predictions through learned parameters θ.

This architecture achieves competitive performance within its computational constraints through efficient temporal filter design [10]. Notably, the network demonstrates superior

computational efficiency, operating at one-fifth the cost of comparable R(2+1)D architectures [45, 10]. This computational advantage is particularly relevant given our project's limited computing resources and the need for rapid experimental iterations.

For our implementation, we utilize the SlowFast-50 variant pre-trained according to the specifications outlined in the original work [10]. This configuration represents the most compact model in the SlowFast family while maintaining favorable computational characteristics compared to other 3D CNN architectures [10], establishing our baseline for further experimentation.

4.2.1 GradCAM Implementation

Two of our methods—described in Sections 4.3.2 and 4.4—rely on attention maps generated using GradCAM [36]. We adapt this technique to the SlowFast architecture by generating attention maps along both spatial and temporal dimensions.

Traditionally, GradCAM produces attention maps by computing gradients of class outputs with respect to activations in the final convolutional layer. In our SlowFast implementation, this corresponds to the 5th residual block, which serves as the last convolutional layer in our network architecture. Therefore, this work uses the 5th layer's activations. We generate separate GradCAM attention maps for both the slow and fast pathways of the network, producing a predicted attention maps $\hat{A} \in \mathbb{R}^{1\times T_s\times H\times W}$, and $\hat{A} \in \mathbb{R}^{1\times T\times H\times W}$. Where T is the temporal resolution for the fast pathway and $T_s = T/\alpha$, for the slow pathway. Section 4.3.2 explains why these two attention maps are important.

4.3 Optical Flow Guided Learning

The essence of this work lies in finding *generalisable* explanation guided learning methods. The theoretical framework for EGL is fairly straightforward: additional annotations from which deep learning models can learn better representations. Previous works focused on obtaining human-generated ground truth explanations such as attention maps. However, these authors concluded that the time and energy requirements for generating such annotations present an intractable problem for sufficiently large datasets [34, 35, 14]. The ability to automatically create ground truth explanations is therefore an important challenge to address.

One observation unique to video data is that divers, or more generally objects, appear naturally segmented from the background in optical flow frames. Figure 4.2 reveals striking similarities between optical flow representations and the attribution maps previously discussed in Figure 2.3. In Figure 4.2, we can clearly observe the silhouettes of divers performing their acrobatics. This visual correspondence motivates the core premise of Optical Flow Guided Learning, which proposes utilizing optical flow frames as the ground truth explanation. We believe that, with sufficient pre-processing, these optical flow frames can be transformed into annotations for use in the EGL framework, making them ideal attribution maps. However, it should be noted that compression artifacts present in the source videos get exacerbated in the optical flow frames, as shown in Figure 4.2. As a result, the ground truth attention maps quality can be poor at times.

This work proposes two novel approaches to EGL that use optical flow frames to obtain ground truth labels: 1) the Dice-based approach (4.3.2) and 2) the Transform approach (4.3.3). However, before discussing the specifics of these methods, let us first address how diver ground truth attention maps are obtained from optical flow frames.

FIGURE 4.2: This figure displays two sample frames from the Diving48 dataset. The figure displays for frames from a single dive sequence. For each frame we display a RGB video frames top-left, optical flow frames, top-right, candidate masks bottom left, and diver binary masks bottom right. These samples should give the reader an impression of the quality of the dataset, and outputs from the pre-processing steps. In Appendix A.2 you can find an extended version of this figure, containing more example frames.

4.3.1 Ground Truth Attention Maps

Figure 4.3 presents our pre-processing pipeline for obtaining diver ground truth Attention Maps A. This approach leverages moving object segmentation, a computer vision task that separates moving objects from the background. We employ FlowSAM [51], a moving object segmentation deep learning technique developed as an extension to Segment Anything [19]. FlowSAM enhances SAM's capabilities by allowing prompting with optical flow frames, which guide the model to segment objects in motion.

The pipeline works as follows: an RGB video X is encoded into an optical flow video stream X_{of} using RAFT [44]. Next, both X and X_{of} are processed by FlowSAM to generate candidate masks M_c, which are ordered by depth layers as classified by the model. Finally, a function $f(M_c)$ determines the most likely diver mask by focusing on the central vertical region of the frame (one-third of the total width, centred) and identifying the most prominent mask ID among the first three depth levels. Specifically, it counts the occurrences of each mask ID in this central region and selects the one with the most pixels. If no suitable mask is found, an empty mask is returned. This heuristic works well for diving videos because divers are consistently positioned in the centre of the frame and typically appear in the foreground layers, making them easily distinguishable from background elements.

This approach leaves a lot of uncertainty about whether the correct mask is chosen by the algorithm. For example, background objects could be misclassified as foreground objects. Furthermore, the raw pixel count of the central object can be misleading due to poor mask quality. There is also limited temporal coherence between masks; whilst RAFT attempts to match masks temporally, this is not always effective. Therefore, this selection approach is not particularly robust, though it is sufficient for the purposes of this work. More sophisticated implementations are considered in our discussion section.

Moreover, the quality of these masks is quite poor at times. The poor quality is likely a result of compression artefacts in the source data, which become more noticeable during fast motion. This is evident when there is very little motion on camera—the quality of the diver masks and optical flow frames is very crisp.

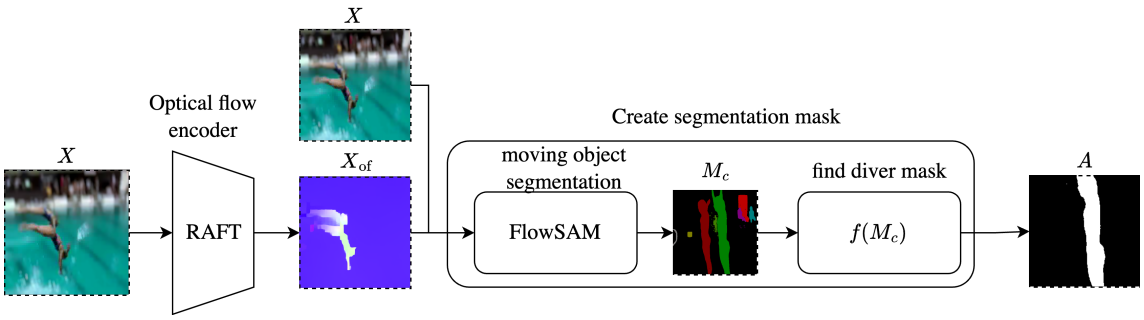

FIGURE 4.3: This figure features our pre-processing pipeline, which takes an input rgb video X, and converts into a binary mask A. The pipeline starts by encoding our input X, into a Optical Flow X_{of} using RAFT [44]. In the next stage, the (X, X_{of}) pair is used as an input to FlowSAM [51] - a moving object segmentation model, that uses optical flow as a prompt. FlowSAM outputs multiple candidate masks M_c, which are filtered by the $f(M)$ to find the most likely binary mask sequence A for the diver.

FlowSAM offers three distinct input methods; this work uses FlowP-SAM, which combines RGB and optical flow to generate segmentation masks. Three key parameters control the trade-off between segmentation quality and computational cost.

The gridside parameter controls the number of prompt points uniformly sampled per frame. We used a gridside of 20, following the default configuration set by the original authors. The max objects parameter determines the number of segments that FlowSAM proposes. We selected 10 objects, though a lower value would be preferable for computational efficiency. However, experimentation with smaller values revealed that the diver would occasionally be misclassified as background and subsequently excluded from segmentation. Therefore, we chose to allow more object proposals to provide greater flexibility in our mask selection algorithm.

The third parameter is gaps, which controls the temporal frame intervals used for optical flow computation. The original authors implemented a preprocessing step that generates optical flow at multiple gap sizes and found that prompting FlowSAM with multiple gaps significantly improves performance [51]. They recommend using gaps of 1, -1, and 2, -2. However, this approach requires generating multiple sets of optical flow frames at different temporal intervals, substantially increasing computational overhead. Therefore, we chose to use only a gap size of 1 to balance performance with computational efficiency. Using the full multi-gap approach represents a promising direction for future work to possibly improve segmentation accuracy.

4.3.2 Dice

Before describing our implementation, we examine whether GradCAM attention maps and segmentation masks are theoretically compatible. GradCAM creates attention maps by looking at which features the model considers most important for classification, while segmentation masks simply outline the entire shape of objects. More specifically, while GradCAM also highlights relative temporal importance, the attention maps treat every frame as equally important. These two approaches highlight different things, which suggests they may not align well together - something our experiments confirm.

Figure 4.3 illustrates our Dice-based approach to Optical Flow Guided Learning. This

method leverages the diver masks obtained through the preprocessing steps described earlier to guide the model's attention mechanism.

The process begins with an input video frame X, which is passed through our SlowFast network augmented with GradCAM to generate an attention map $\hat{A}$. This attention map represents the model's current focus areas when making predictions. Simultaneously, we utilize the ground truth diver mask A derived from optical flow processing as described in Section 4.6.2. This binary mask serves as the ideal attention target that we want our model to learn.

During training, the model receives a tuple $(A, \hat{A}, O, Y)$, where: A is the ground truth diver mask (binary segmentation). $\hat{A}$ is the model-generated attention map from GradCAM. O is the model's output prediction. And Y is the ground truth class label. Our loss function combines two components:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{Dice}}(A, \hat{A}) + \beta \mathcal{L}_{\text{ce}}(O, Y)$$

The Dice loss component is defined as:

$$\mathcal{L}_{\text{Dice}}(A, \hat{A}) = 1 - \frac{2|A \cap \hat{A}|}{|A| + |\hat{A}|}$$

Where $\mathcal{L}_{\text{Dice}}$ is the Dice similarity coefficient loss that measures the overlap between the ground truth mask and the generated attention map, encouraging the model to focus on the correct regions of interest. $\mathcal{L}_{\text{ce}}$ is the standard cross-entropy loss for classification. The hyperparameters α and β control the relative importance of each loss component.

So far we have treated $\hat{A}$ and as a single attention map. However, as described in Section 4.2.1, we actually produce two separate attention maps: one for the slow path way and another for fast pathway. This allows us to calculate the loss for the two pathways separately. By applying our loss to both pathways, we ensure that gradients flow through and influence the entire architecture, rather than affecting only a single pathway. This approach allows the model to make adjustments to both temporal (slow path) and motion (fast path) based on the respective attention maps.

By jointly optimizing for both accurate classification and attention alignment with the diver's actual position, the model learns to focus on relevant features while improving its predictive performance.

4.3.3 Temporal Mask Transform

Our Transform approach offers a more straightforward method for explanation guided learning. As illustrated in Figure 4.5, this technique leverages the pre-processed diver attention masks A to segment RGB video frames X through a simple masking operation $M \cdot X$.

The core concept is simple: by applying the binary mask to the original frame, we create a masked image MX where only the pixels corresponding to the diver region retain their values, while background pixels are set to zero (black). This transformation forces the model to attend exclusively to the relevant subject area, as all contextual information outside the diver silhouette is eliminated.

We implement this segmentation transformation as a configurable pre-processing step with a predefined probability, complementing standard augmentations such as random cropping, random horizontal flipping, and normalization (described in Section 4.6.2). This probabilistic application provides a balance between learning from fully segmented inputs and maintaining contextual information when necessary.

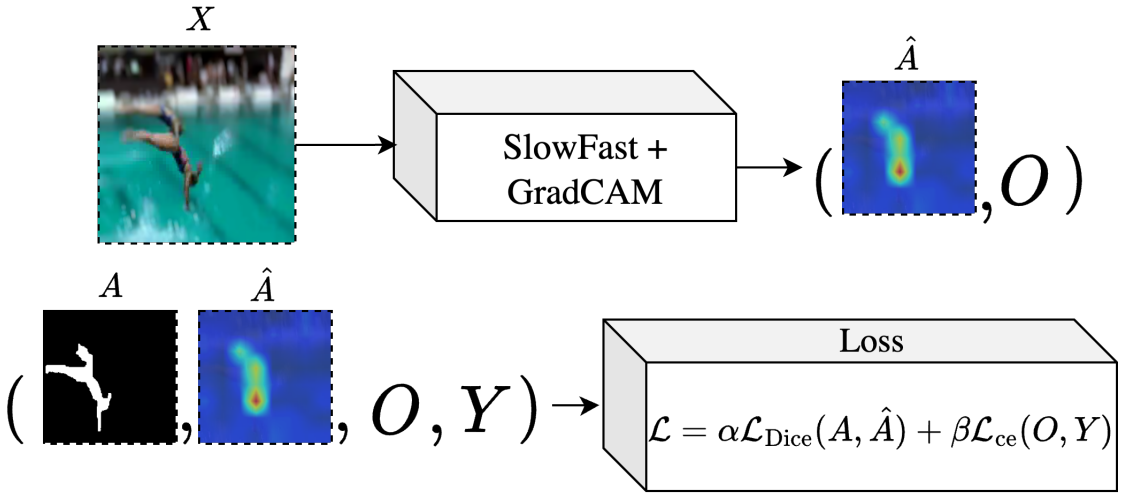

FIGURE 4.4: Diagram of the Dice-based Optical Flow Guided Learning approach. The model processes rgb video X through SlowFast+GradCAM to generate attention map $\hat{A}$, while using ground truth mask A, from optical flow pre-processing step 4.3.1. These are combined with classification outputs (O, Y) in the loss function $\mathcal{L} = \alpha\mathcal{L}_{\text{Dice}}(A, \hat{A}) + \beta\mathcal{L}_{\text{ce}}(O, Y)$ to jointly optimize for accurate classification and attention localization.

By constraining the visual information available to the model, we facilitate learning of important features directly related to the diver's movements, and posture.

4.4 Right for the Right Reason Learning

This section describes our application of the right-for-the-right-reason loss $\mathcal{L}_{\text{rrr}}$, introduced by Ross et al. (2017) [34]. The theoretical foundation for this method was discussed in Section 2.4.

As shown in Equation 4.1, the $\mathcal{L}_{\text{rrr}}$ loss function penalizes gradient magnitudes within specific regions of an input sample X, marked by a binary attention map $A \in \{0,1\}^{T\times H\times W}$, where $\hat{y}_n$ represents the predicted probability for class n. This approach imposes a soft inductive bias on the model by discouraging reliance on features in regions marked by A, while still allowing the model flexibility to use these regions when necessary for correct classification.

$$\mathcal{L}_{\text{rrr}} = \lambda A \frac{\partial}{\partial X} \sum_{n=0}^{N} \log(\hat{y}_n)^2 \tag{4.1}$$

We designed two methods for obtaining the ground truth attention maps needed for this approach:

The first method, illustrated in Figure 4.6, uses GradCAM-derived attention maps. As shown in the diagram, an RGB video frame X is processed through a SlowFast+GradCAM network, producing both class probabilities O and an attention map $\hat{A}$. This attention map undergoes thresholding $(t > \hat{A})$ to create a binary mask A. When applied in the loss

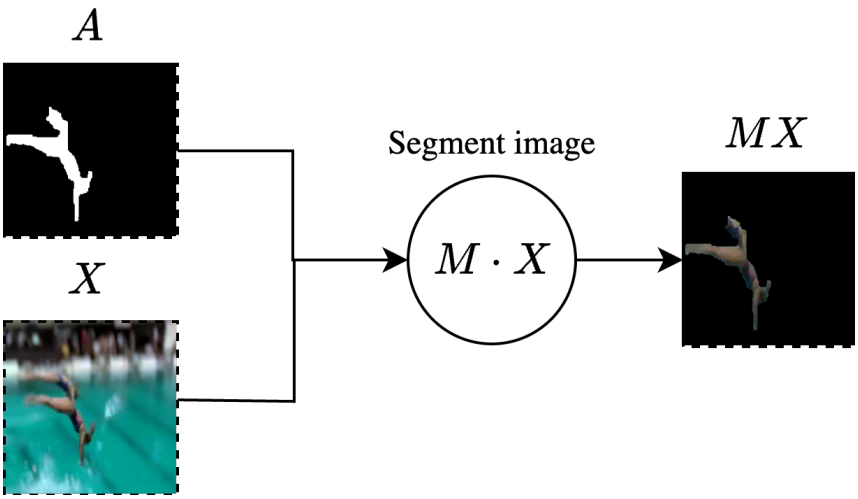

FIGURE 4.5: Diagram of the Transform OGL approach. This method leverages, attention masks obtained through 4.3.1, to mask the diver silhouette for a RGB video X. Segmenting is performed as pre-processing step with a configurable probability. This allows the model to learn relevant features in the presence of bad ground truth attention masks.

function, this approach penalizes the model for focusing on regions that fall within the thresholded attention area.

Importantly, we collect statistics on individual samples across training iterations and apply this penalty selectively to the Nth percentile worst-performing samples—those consistently misclassified by the model. The rationale is that persistent misclassification likely indicates the model is focusing on incorrect features. By applying the binary mask penalty to these specific challenging samples, we force the model to explore alternative visual features that might lead to correct classification. Furthermore, this targeted penalty effectively increases the model's attention on these difficult samples during training, potentially improving overall robustness.

For the second method, we utilize the diver masks obtained through optical flow processing (described in Section 4.3.1) but invert them—transforming previously black pixels to white and vice versa. This inverted mask effectively penalizes high gradients in regions outside the diver, encouraging the model to focus specifically on the subject rather than background elements or contextual cues.

Both approaches implement the core principle of the $\mathcal{L}_{\text{rrr}}$ loss: guiding the model to make correct predictions based on relevant visual features rather than spurious correlations or background context.

4.5 Evaluation

To assess the performance of our proposed methods, we conducted a series of experiments evaluating both classification accuracy and attention quality. We report test scores for several baseline models and our proposed approaches across different temporal resolutions T (16, 32, and 64 frames).

For classification performance, we measure top-1 accuracy on the test set to evaluate how effectively each method recognizes the 48 distinct diving classes. This metric allows

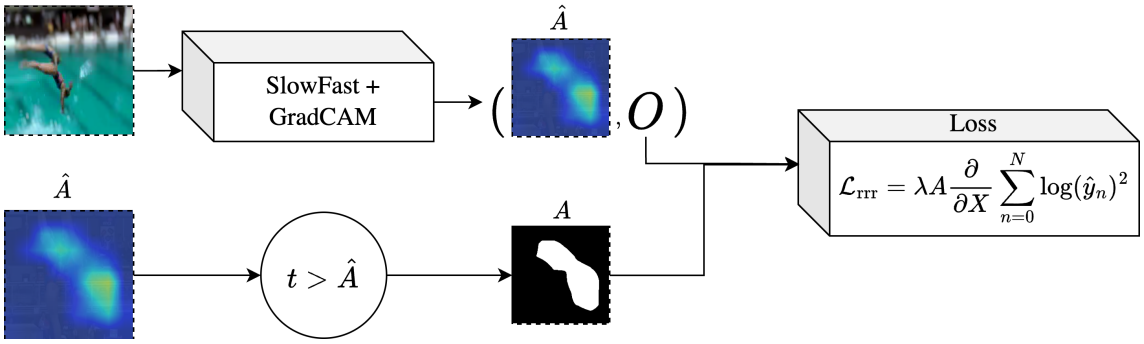

FIGURE 4.6: Diagram of the Right for the Right Reason Learning (RRR) approach. This approach revolves around the RRR Loss function - $\mathcal{L}_{\text{rrr}} = \lambda A \frac{\partial}{\partial X} \sum_{n=0}^{N} \log(\hat{y}_n)^2$ penalises high gradient magnitude in regions marked by $A \in 0, 1$ (a binary mask). The mask A is obtained by processing a RGB video X, and subsequently applying a threshold $t > \hat{A}$.

us to directly compare the benefit of our OGL, and RRR Learning approaches against standard SlowFast implementations without attention guidance.

For our Dice-based approach, we perform an additional evaluation focused on attention map quality. Specifically, we quantify how well the model's GradCAM attention maps $\hat{A}$ align with the ground truth diver masks A derived from optical flow. We measure this alignment using two metrics:

1. Intersection over Union (IoU): This metric calculates the ratio of the overlap area to the union area between the generated attention map and the ground truth mask, providing a strict measure of spatial alignment.

2. Dice coefficient: Consistent with our training objective, we report the Dice similarity score between GradCAM outputs and ground truth masks, which measures the overlap while being less sensitive to small spatial displacements than IoU.

4.6 Training Setup

For this work, several models will be trained under varying configurations, such as Right for the Right Reason Learning (Section 4.4 and Optical Flow Guided Learning 4.3. However, some configuration options remain constant, which will be discussed presently. Each model is trained for 100 epochs using Stochastic Gradient Descent (SGD) as the optimizer with a base learning rate of 0.1, weight decay of 0.0001, and momentum of 0.9.

The training run starts with learning rate warmup for the first 100 *loader batches* B_l - more on what this means later. Beginning at a learning rate of 0.01, the warmup gradually scales the learning rate up to the base learning rate of 0.1. Models are trained with varying numbers of frames T sampled per video, using values of 16, 32, and 64. Cross-entropy serves as our loss function. To account for class imbalance, we employ weighted cross-entropy where each class's weight is inversely proportional to its frequency in the training data. Specifically, the weight for class i is calculated as $w_i = 1 - \frac{n_i}{N}$ for $i \in \{0, 1, ..., 48\}$, where N is the total number of samples in the training dataset, and n_i is the number of samples belonging to class i.

The batch size is set to $B = 256$. For completeness, a distinction must be made between *batch size* (B) and *loader batch size* (B_l). The *loader batch size* B_l represents the number of samples that fit into GPU memory at once. Meanwhile, the batch size B represents

the total number of samples the model processes before the optimizer performs an update step.

To perform mini-batch SGD on batch sizes that don't fit into GPU memory, we choose a smaller B_l that fits into memory, where B is a multiple of B_l. At each iteration, the loss is scaled by $\frac{B}{B_l}$ before the backward pass. This gradient accumulation approach allows us to effectively train with larger batch sizes than would normally fit in GPU memory. In practice, this should have little to no effect on the final outcome compared to training with the full batch at once, but is included for reproducibility purposes.

4.6.1 Dynamic Temporal Stride

We uniformly sample T frames from a video x which has N_x frames. In practice, we sample every $\lceil N_x/T \rceil$ frames, starting at frame zero, where $\lceil \cdot \rceil$ denotes the ceiling function.

This approach addresses a key limitation in standard action recognition, since sampling at a fixed stride can omit significant portions of the video content. For instance, sampling at a 4×16 interval would cause the model to miss a substantial number of frames. Since the model needs the complete sequence to classify actions like diving, it's crucial to observe frames across the full temporal range of the video. Additionally, sampling a consistent number of frames (T) enables efficient batch processing during training.

However, even with uniform sampling, we remain restricted to a fixed subset of frames. To further increase sampling diversity, we randomly offset the starting point by applying a shift t to each frame index i, where t is randomly sampled from the range $[0, \lceil N_x/T \rceil]$. The actual frame indices selected become $(i \cdot \lceil N_x/T \rceil + t)$ for $i \in \{0, 1, ..., T-1\}$. This randomization shifts the entire regular sampling pattern, allowing us to access frames that would otherwise be missed by the fixed interval sampling. Here, T consistently represents the number of frames to be sampled, and N_x represents the total number of frames in the video.

4.6.2 Augmentation & Pre-processing

The data processing pipeline implements separate augmentation strategies for training and evaluation to ensure both effective learning and reproducible results. This pre processing pipeline is typical for action recognition tasks [7, 10], though it contains two notable additions, namely: 1) RandAugment, and 2) OGL Transform.

During training, we apply the following sequence of preprocessing steps:

1. Pixel values are scaled to the range [0,1]
2. Random short side scaling with minimum=256 and maximum=320 pixels to maintain aspect ratio
3. Random crop to 224 × 224 pixels
4. RandAugment, if applicable - see Section 4.6.3, [8]
5. OGL Transform, if applicable - see Section 4.3.3
6. Normalize the RGB pixel values around $\mu = [0.31, 0.47, 0.5]$ and $\sigma = [0.2, 0.2, 0.23]$
7. Random horizontal flip with 50% probability

For evaluation, we use a more deterministic approach to ensure consistent results:

1. Pixel values are scaled to the range [0,1]

2. Short side scaling to 320 pixels
3. Center crop to 256 $\times$ 256 pixels
4. Normalisation using the same μ and σ values as in training

4.6.3 RandAugment

RandAugment is a preprocessing technique that applies random augmentations to training data, consistently shown to improve model performance [8]. These augmentations include translation-xy, shear-xy, contrast adjustments, rotation, and other transformations. We used the implementation from the PyTorchVideo Python package.

We hypothesized that these diverse augmentations might simulate some generalization benefits typically gained from larger datasets. Since RandAugment transforms each sample uniquely, a smaller dataset effectively becomes more varied during training, potentially achieving generalization properties similar to those of larger datasets.

Unfortunately, RandAugment could only be applied to the baseline models, not to the attention-guided approaches. This limitation stems from technical constraints in the PyTorchVideo library, which does not provide sufficient control over transformation parameters. For our attention-guided methods, precise alignment between input frames and ground truth attention maps is critical, requiring synchronized transformation parameters that the library couldn't guarantee.

Chapter 5

Results

This section presents a comprehensive evaluation of the proposed Explanation Guided Learning methods compared to baseline approaches described in Chapter 4). In Section 5.2 we investigate some unexpected performance characteristics of our Dice based approach. Section 5.3 evaluates influence of the λ and *worst performer percentile*, on the right-reason based approaches. The following section compares our methods with the SoA on the Diving48 benchmark were we make two interesting observations about temporal sampling and the temporal resolution of our model (Section 5.4). This chapter concludes by analysing model complexity in terms of GFLOPs and training cost, Sections 5.5, and 5.6 respectively.

5.1 Model Comparison

This analysis examines performance across multiple temporal resolutions and contrasts our methods with standard augmentation techniques transfer learning approaches, and advanced augmentation strategies like RandAugment (described in Section 4.6.2). We evaluate seven distinct model configurations, each representing different learning and augmentation strategies to the diving classification task, as described below:

- **Vanilla** 4.2: Our baseline SlowFast network trained from scratch without any additional techniques or pre-training.
- **Vanilla + Kinetics** 4.2: The SlowFast network initialized with pre-trained weights from the Kinetics dataset, leveraging transfer learning to improve performance.
- **RandAug + Kinetics** 4.6.3: Extends the Kinetics pre-trained model with RandAugment data augmentation to increase sample diversity during training.
- **OGL + Dice** 4.4: Our Optical Flow Guided Learning approach that uses the Dice loss to align model attention with diver silhouettes derived from optical flow, initialized with Kinetics pre-trained weights.
- **OGL + Transform** 4.3.3: The transform-based variant of our optical flow method that directly masks input frames based on diver silhouettes, initialized with Kinetics pre-trained weights.
- **RRR + GradCAM** 4.4: Implements the Right for the Right Reason approach using GradCAM-derived attention maps to guide the model's focus, initialized with Kinetics pre-trained weights.

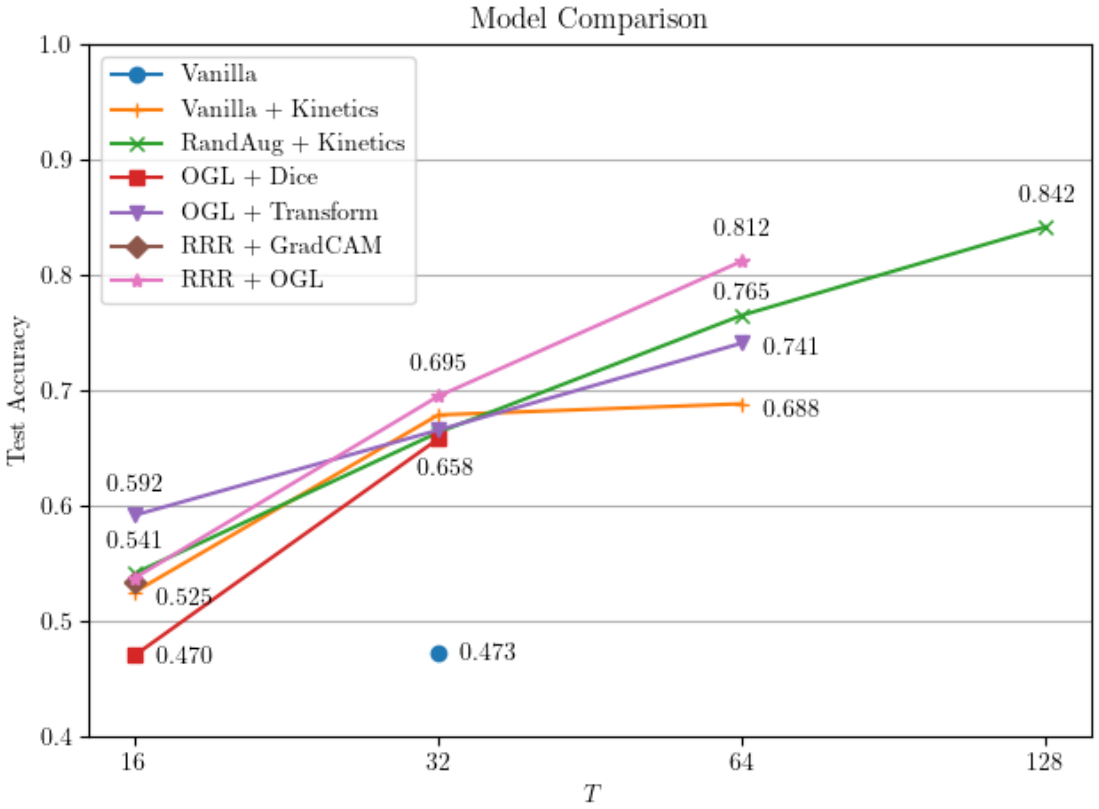

FIGURE 5.1: Plot showing training configurations at temporal resolutions T along the x-axis and their accuracies along the y-axis.

- **RRR + OGL** 4.4: Combines Right for the Right Reason with optical flow-derived attention maps instead of GradCAM, initialized with Kinetics pre-trained weights.

It's important to note that the Vanilla configuration is the only method that does not use pre-trained Kinetics weights.

Figure 5.1 and Table 5.1 present a comprehensive comparison of model performance across different temporal resolutions ($T \in \{16, 32, 64, 128\}$). The baseline SlowFast model ("Vanilla") demonstrates the lowest performance, achieving only 0.473 accuracy at $T = 32$. Adding Kinetics pretraining ("Vanilla + Kinetics") substantially improves performance, reaching 0.679 at $T = 32$ and 0.688 at $T = 64$. This confirms the value of transfer learning from larger action recognition datasets, even when the target domain is specialized.

RandAugment combined with Kinetics pretraining ("RandAug + Kinetics") demonstrates the strongest overall performance trajectory, particularly at higher temporal resolutions. This configuration achieves 0.541 at $T = 16$, 0.664 at $T = 32$, 0.765 at $T = 64$, and peaks at 0.842 with $T = 128$, the highest performance across all methods. This validates our hypothesis that strong augmentation provides generalization benefits that approximate those of larger datasets.

Our proposed Optical Flow Guided Learning approaches show variable performance. The Dice-based method ("OGL + Dice") performs poorly at $T = 16$ with only 0.470 accuracy, lower than even the baseline model. This suggests the approach struggles to align attention maps effectively with limited temporal information. In contrast, the Temporal Mask Transform approach ("OGL + Transform") shows strong performance at $T = 16$

Method	T=16	T=32	T=64	T=128
Vanilla	-	0.4732	-	-
Vanilla + Kinetics	0.525	0.678	0.688	-
RandAug + Kinetics	0.541	0.663	0.765	**0.842**
OGL + Dice	0.470	0.657	-	-
OGL + Transform	0.591	0.665	0.740	-
RRR + GradCAM	0.533	-	-	-
RRR + OGL	0.537	0.694	0.812	-

TABLE 5.1: Performance comparison across different methods and temporal resolutions (T). The best overall performance is highlighted in bold. Values are rounded to 4 decimal places for readability.

with 0.592 accuracy—the highest among all methods at this resolution—outperforming RandAug + Kinetics. At higher resolutions, this method maintains competitive performance with 0.665 at $T = 32$ and 0.740 at $T = 64$.

The Right for the Right Reason methods demonstrate interesting temporal scaling behaviour. The GradCAM variant ("RRR + GradCAM") achieves 0.534 accuracy at $T = 16$, while the optical flow variant ("RRR + OGL") shows remarkable improvement with temporal resolution. Starting at 0.537 for $T = 16$, it substantially outperforms other methods at $T = 32$ (0.695) and $T = 64$ (0.812). The Temporal Mask Transform method achieved a 2.47% improvement over the "Kinetics + RandAug" approach and a 5.06% improvement over "Vanilla + Kinetics".

Temporal resolution proves to be a critical factor across all models. Performance consistently improves as the number of frames increases, with the most dramatic gains observed in the RandAug + Kinetics configuration, which improves by approximately 0.30 in accuracy from $T = 16$ to $T = 128$. Similarly, RRR + OGL shows exceptional scaling, improving by 0.27 from $T = 16$ to $T = 64$.

The influence of temporal resolution highlights the importance of capturing sufficient motion information for accurate diving classification. While methods perform similarly at $T = 16$ (ranging from 0.470 to 0.592), the performance gap widens considerably at higher resolutions, where methods better equipped to leverage temporal information demonstrate superior performance. This suggests that the temporal dynamics captured at higher frame rates are crucial for distinguishing between similar diving techniques.

In the interest of time, we conducted most experiments at temporal resolution $T = 16$, with only select configurations evaluated at higher resolutions. Training these models is computationally intensive, with a single experiment requiring several days to complete. This constraint is particularly significant for the RRR and OGL methods, which involve second-order gradients that substantially slow the back-propagation process compared to other approaches. These computational constraints are explored further in Section 5.6.

The strong performance of OGL + Transform at $T = 16$ and the exceptional scaling of RRR + OGL at higher temporal resolutions demonstrate the potential of optical flow-guided approaches, while the poor performance of the Dice-based method warrants further investigation to understand the underlying limitations.

5.2 Dice Loss Evaluation

This section evaluates the characteristics of the Dice loss in terms of the Dice factor and IoU. Furthermore, we investigate the influence of hyperparameters α and β on test accu-

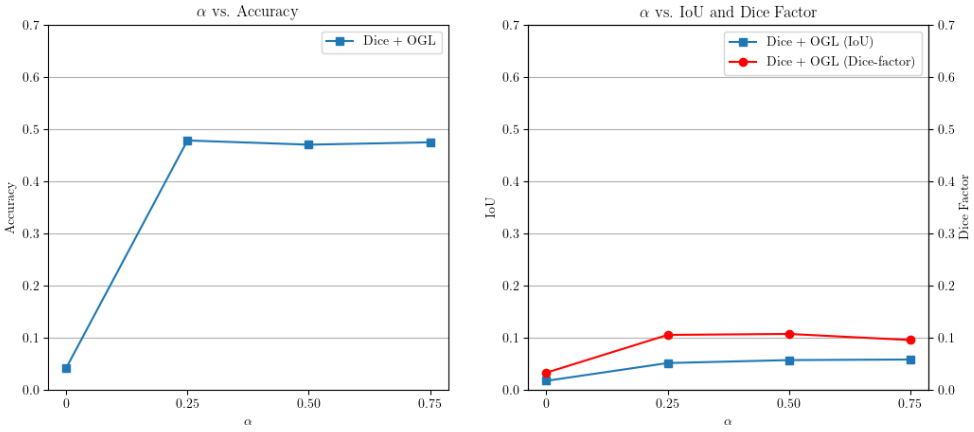

FIGURE 5.2: Impact of α parameter on model performance and attention alignment. (a) α vs. Accuracy shows a sharp improvement when increasing α from 0 to 0.25 (prioritizing cross-entropy loss), followed by performance plateau with further increases. (b) α vs. IoU and Dice Factor reveals consistently low attention alignment metrics across all α values, even when the Dice component is heavily weighted in the loss function ($\alpha = 0.25$). This suggests a fundamental incompatibility between GradCAM-generated attention maps and binary segmentation masks as learning targets, rather than simply an issue of loss weighting.

racy, IoU, and Dice factor values. Our primary goal is to develop a clear understanding of why the Dice-based approach underperforms, examining whether the issue stems from loss function weighting, fundamental misalignment between GradCAM attention and ground truth masks, or inherent limitations in using binary segmentation masks as attention targets.

Recall that α and β control the contribution of the Cross-entropy and Dice loss components respectively. As shown in Figure 5.1, the **OGL + Dice** method had a negative effect on final performance. To understand this behavior better, we examined varying values of $\alpha \in \{0, 0.25, 0.5, 0.75\}$ with $\beta = 1 - \alpha$.

Figure 5.2 illustrates the influence of α on test performance, IoU, and Dice factor. When $\alpha = 0.0$, the cross-entropy loss is effectively disabled, leaving only the Dice loss for training. In this configuration, test accuracy drops to nearly zero (0.0418), just slightly above random guessing. This makes intuitive sense - without cross-entropy loss, the model lacks the signal needed to learn class distinctions. When α is increased to 0.25, accuracy improves dramatically to 0.478. However, further increases to α yield no additional performance improvements.

Notably, as seen in Figure 5.2b, the IoU and Dice factor values remain between 0.0 and 0.1 regardless of changes to α. This is particularly surprising for lower values of α (e.g., $\alpha = 0.0$), where we would expect significantly higher Dice factor and IoU values since the Dice component receives more weight in the loss function and should send a stronger signal to the model to align its attention maps with ground truth masks. However, we see the opposite, as α becomes higher we observe a small increase in IoU and Dice factor.

These consistently low values across all α settings indicate that our method for learning from GradCAM is fundamentally flawed - the model is not successfully aligning its attention maps with the ground truth segmentation masks even when the loss function heavily prioritizes this alignment.

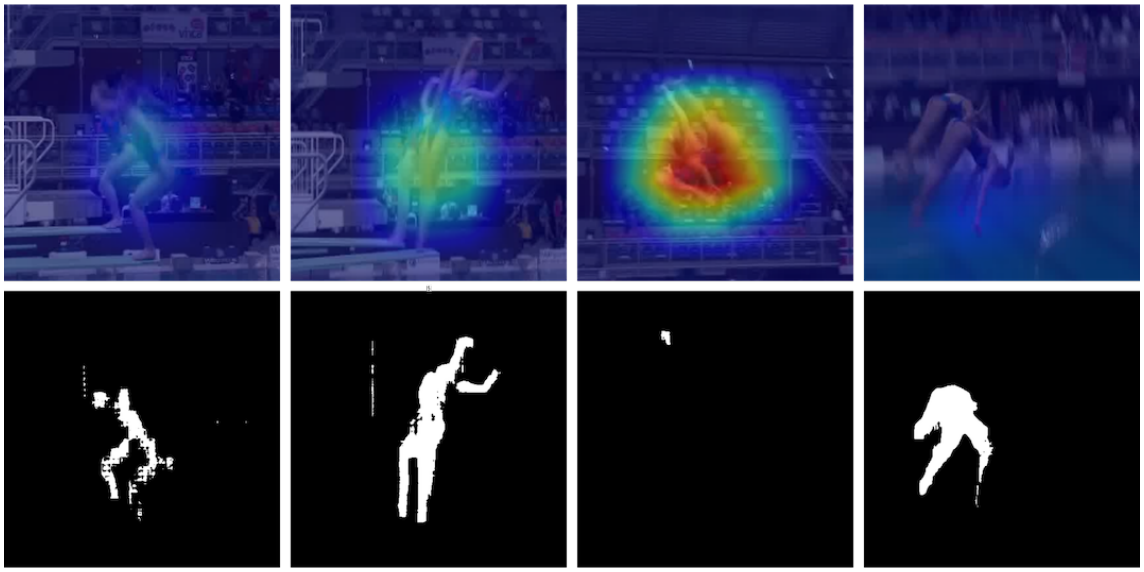

FIGURE 5.3: Temporal comparison of attention maps across four consecutive frames from a diving sequence. Top row: GradCAM attention maps generated by our model, with blue indicating low activation and red indicating high activation. Bottom row: Corresponding ground truth diver segmentation masks derived from optical flow, with white indicating diver presence. Note how the model correctly ignores the first frame (minimal GradCAM activation) where no significant action occurs, despite the diver's presence in the segmentation mask. This temporal misalignment between classification-relevant features and diver presence explains why the Dice loss provides counterproductive training signals.

These results suggest that adding the Dice loss as a secondary objective negatively impacts overall performance. To understand why, we can examine the GradCAM outputs $\hat{A}$ and the diver ground truth masks A in Figure 5.3. The figure shows a sequence of GradCAM outputs and segmentation masks across four frames. In the first frame, activations are minimal in the GradCAM output (blue regions), while the corresponding ground truth mask shows significant diver presence (white regions). This disparity yields a high loss value for this particular frame. From a XAI perspective, these GradCAM frames are good explanation, as the model has correctly learned not to focus on initial frames where no significant action occurs. Only in the second and third frames do we see increasing GradCAM activations that better align with the diving action, and our own understanding of when and where a model should look. This misalignment demonstrates how the Dice loss can provide counterproductive signals, effectively penalizing the model for correctly ignoring non-informative frames and instead forcing it to attend to all frames where a diver is present, regardless of action relevance.

5.3 Right for the Right Reason

We evaluated two key parameters: the penalty strength (λ) and the worst performer percentile threshold for sample selection. For λ optimization, we fixed the worst performer percentile at 10; for percentile optimization, we set $\lambda = 0.1$.

Table 5.2 shows that increasing λ had minimal or slightly negative effects on performance, with $\lambda = 0.01$ yielding the best results (53.4% accuracy). Higher λ values

also caused training instability in both loss and accuracy. Ross et al. [34] recommended $\lambda = 1000.0$ to balance the Right Reason loss magnitude with cross-entropy loss, but our diving dataset required substantially lower values. Similarly, varying the worst performer percentile showed limited impact, with the 20th percentile performing marginally better (51.4% accuracy).

The core premise of this method — allowing models to discover alternative explanations by penalizing structurally misclassified samples — did not translate to improved test accuracy when using GradCAM attention maps. This provides further evidence of a fundamental disconnect between GradCAM's apparent representations and its actual function, consistent with our findings in Section 5.2 regarding the mismatch between ground truth diver masks and GradCAM attention.

However, optical flow-derived masks showed promise in other contexts. The RRR + OGL method (Figure 5.1) achieved accuracy improvements of 3.14% and 4.70% at $T = 32$ and $T = 64$ respectively, despite a slight decrease at $T = 16$. Similarly, the OGL + Transform method's positive performance at $T = 16$ suggests these diver masks contain valuable information, even if the attention-based alignment approach proves ineffective. Therefore, these diver ground truth attention masks together with the RRR Loss serve as a good soft inductive bias.

Parameter	**Accuracy (%)**	**Setting**
Lambda (λ) Values		
$\lambda = 0.01$	53.4	-
$\lambda = 0.1$	49.4	-
$\lambda = 0.5$	48.9	-
$\lambda = 1.0$	49.9	-
Worst Performers Percentile		
10th percentile	49.4	$\lambda = 0.1$
20th percentile	51.4	$\lambda = 0.1$
30th percentile	49.4	$\lambda = 0.1$

TABLE 5.2: Right for the Right Reason (RRR) hyperparameter evaluation results on Diving48. The table shows accuracy results for different lambda penalty strengths and percentile thresholds for selecting worst-performing samples to apply the penalty.

5.4 Comparison with SoA

Table 5.3 presents a comparison of our model against state-of-the-art architectures on the Diving48 dataset [21, 3, 49]. Our implementation of the SlowFast network (highlighted in bold) demonstrates performance that exceeds both TimeSformer and the original SlowFast implementation. However, direct comparisons warrant careful interpretation due to variations in evaluation protocols and differences in architectural complexity. Notably, our implementation utilizes fewer parameters. More precisely, SlowFast-R50 with 50 convolutional layers, whereas the comparison SlowFast-R101 employs a deeper 101-layer architecture as reported by Bertasius et al. (2021) in their TimeSformer paper. Indicating, our method is a substantial improvement since far fewer parameters are used.

The third column reports the temporal resolutions (T) the models were trained at. Note that, for the SlowFast architectures we report the slow/fast temporal resolutions re-

spectively. The general trend is that higher temporal resolutions yield greater performance, as indicated by our own results 5.1.

Two key factors differentiate our approach from the **SlowFast-R101 16×8** implementation: First, we double the slow pathway's temporal resolution from 16 to 32 frames, and second, we employ our *dynamic temporal stride* sampling method described in Section 4.6.1. Our superior performance despite using a smaller backbone (R50 vs. R101) suggests that temporal resolution in the slow pathway plays a more critical role than model depth for diving classification. This finding is particularly noteworthy given that diving sequences feature rapid movements that would theoretically benefit from the fast pathway's higher frame rate processing. The performance gap may indicate that while the fast pathway captures motion efficiently, it may lack sufficient channel capacity to fully interpret the complex temporal patterns in diving sequences. This interpretation aligns with our observation that increasing temporal resolution in the "spatial-focused" slow pathway yields better performance gains than simply scaling up model depth. The original SlowFast paper assumes that temporal features require fewer channels than spatial features. However, for highly technical activities with complex and rapid motion patterns this may not hold, as the fast pathway lacks representation capacity.

The lower performance of methods reported in Li et al. (2018) asks for examination within the context of the Diving48 dataset's design objectives. The authors intentionally constructed this benchmark to minimize environmental cues that could be exploited by deep learning model, creating what they termed "a controlled environment" where models must focus on temporal action dynamics rather than contextual shortcuts [21]. This design philosophy stands in contrast to other action recognition datasets where incidental features (e.g., background, equipment, clothing) often correlate strongly with action classes, potentially enabling models to achieve high accuracy without truly understanding the temporal dynamics of the actions themselves.

Method	**Accuracy (%)**	T
TSN (RGB)[21]	16.77	N/A
TSN (Flow)[21]	19.64	N/A
TSN (RGB+Flow)[21]	20.28	N/A
C3D [21]	11.51	8
C3D [21]	16.43	16
C3D [21]	21.01	32
C3D [21]	27.60	64
TimeSformer[3]	74.90	8
TimeSformer-HR[3]	78.00	16
TimeSformer-L[3]	81.00	96
SlowFast-R101 16x8 [3]	77.6	16/128
SlowFast-R50 32xτ^* (Ours)	**84.2**	**32/128**
BEVT [49]	87.2	N/A

TABLE 5.3: Comparison of action recognition accuracies, and temporal training resolution T across different architectures on Diving48 cited from different publications [21, 3, 49]. Ours in bold is comparable to what reported in the Bertarius et al (2021) in the TimeSformer paper. BEVT is the best and current SoA, in terms of performance for Diving48. Note that for the SlowFast architecture we report the slow and fast temporal resolutions respectively.

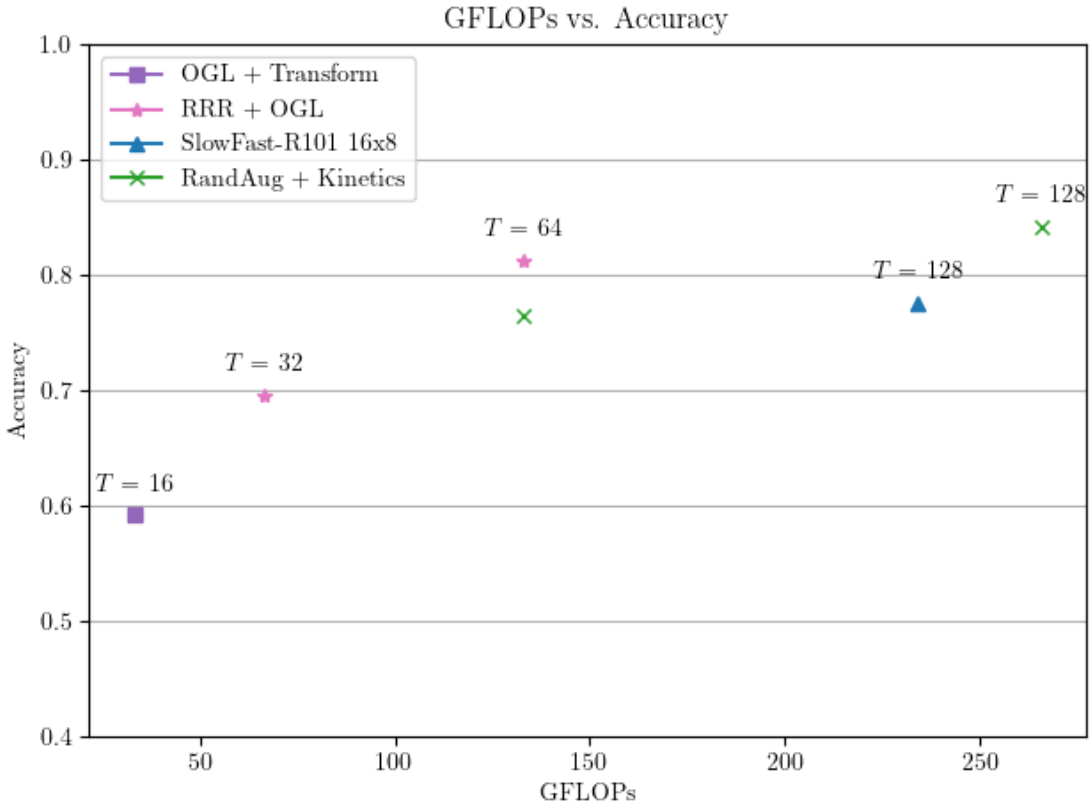

FIGURE 5.4: Accuracy/Complexity trade-off across methods and temporal resolutions. The horizontal axis shows computational complexity in GFLOPs, while the vertical axis shows classification accuracy on Diving48. Our RandAug+Kinetics model at $T = 64$ achieves comparable accuracy (0.765) to SlowFast-R101 16x8, from Bertasius et al (2021) [3] (0.776), while requiring only half the computational resources. At $T = 128$, our model reaches 0.842 accuracy, significantly outperforming prior work. This improvement can be attributed to our denser temporal sampling in the slow pathway. The OGL + Transform and RRR + OGL methods show competitive performance at lower temporal resolutions and computational costs.

5.5 Accuracy/Complexity Trade-off

Figure 5.4 visualizes the accuracy-complexity trade-off across our models and temporal resolutions. The results for Diving48 from Bertasius et al.'s (2021) were included to highlight the effectiveness of our methods. Our "RRR + OGL" model demonstrates exceptional efficiency, achieving 0.812 accuracy at $T = 64$ with only 132.84 GFLOPs, compared to Bertasius et al.'s (2021) model which requires 234 GFLOPs to reach a 0.776 accuracy. While our method only uses approximately 56.0% fewer computations. Our model at $T = 128$ achieves 0.842 accuracy at 265.69 GFLOPs, representing a significant 6.6% improvement over prior work. At lower computational budgets, our Temporal Transform method achieves 0.592 accuracy with just 33.21 GFLOPs ($T = 16$), while our "RRR + OGL" approach reaches 0.695 accuracy at 66.42 GFLOPs ($T = 32$). These results demonstrate that our approach not only improves classification performance but does so with substantially better computational efficiency, highlighting the effectiveness of our tempo-

ral sampling strategy, and random augmentations. Even though the presented methods show favourable inference cost. They are much more costly in terms of training time.

Method	**Hours/Epoch**
Vanilla	0.236
RandAug	0.289
RRR + GradCAM* ($T = 16$)	0.467
Temporal Mask Transform	0.603
RRR + Temporal Mask	0.911
OGL + Dice	1.183

TABLE 5.4: Comparison of hours per epoch performance across different methods. Ordered from shortest to longest. The tests where recorded at $T = 32$, on a single NVIDIA L40 GPU. The RRR + GradCAM* ($T = 16$) row was recorded at $T = 16$, since there were no runs at $T = 32$ for this method. Therefore, the time was doubled to give an estimate of this method $T = 32$. Our methods impose a significant additional training cost, this training cost can mostly be attributed to second order gradients, and additional IO costs of loading the ground truth attention masks from disk.

5.6 Training Cost

Table 5.4 presents the training cost of the different methods explored in this work expressed hours/epoch. The EGL based approaches - "RRR + GradCAM", "Temporal Mask Transform", "RRR + Temporal Mask", and "OGL + Dice" - impose significant training cost over the non-EGL based approaches - "Vanilla", and "RandAug". The additional training cost can mostly be attributed to second-order gradients during backpropagation in the Dice and RRR loss approaches. The $\mathcal{L}_{\mathcal{RRR}}$ in Eq 4.1 contains a gradient term $\frac{\partial}{\partial X}\sum_{n=0}^{N}\log(\hat{y}_n)^2$ as part of the loss function itself. During the forward pass, computing this loss requires calculating gradients with respect to the input X, which creates a computational graph where gradient operations become nodes. Subsequently, during the backward pass, computing $\frac{\partial \mathcal{L}_{\text{rrr}}}{\partial \theta}$ (where θ represents model parameters) requires differentiating through these embedded gradient computations. This creates higher-order gradients - gradients of gradients - making the backward pass significantly more computationally expensive, approximately tripling the training time compared to standard loss functions.

Furthermore, reading temporal masks from disk adds significant IO overhead, as demonstrated by the "Temporal Mask Transform" timing. This explains why "OGL + Dice" performs worst in terms of training time, as it combines both second-order gradients in the loss function and disk-based mask loading. The "RRR + GradCAM" method takes approximately twice as long as the "Vanilla" method. However, this does not present the complete picture, since we only apply the RRR loss to 10% of samples. Therefore, "RRR + OGL" provides a more realistic estimate of the computational cost imposed by the RRR loss when applied more broadly.

The substantial training time penalties caused by second-order gradients make the EGL loss functions explored in this work computationally impractical for large-scale applications, highlighting a critical limitation that must be addressed in future research.

This concludes the results section. While the Dice-based approach did not yield the anticipated improvements, our investigation provided valuable insights into the challenges of aligning attention mechanisms with segmentation-based guidance. The RRR-based methods demonstrated performance comparable to the baseline, and notably, the OGL Transform approach showed meaningful performance gains. Furthermore, we demonstrated that the dynamic temporal stride sampling strategy combined with increased slow pathway temporal resolution enabled our model to outperform larger SlowFast architectures despite using fewer computations. The next chapter presents the discussion and conclusions, where we analyse the limitations of our approaches, theorise about the underlying causes, and propose directions for future research that could address these challenges.

Chapter 6

Discussion

Our investigation reveals insights about the application of EGL to action video action recognition. While the primary hypothesis regarding Dice-based attention alignment proved incorrect, this negative result provides important theoretical insights. In contrast, the "OGL + Transform" method achieved a 6.67% improvement over "RandAug + Kinetics" at lowest temporal resolution. Similarly, the "OGL + RRR" method improved upon the baseline at temporal resolutions of 32 and 64 frames. Moreover, this approach required far fewer computation compared to much larger SlowFast models. The Temporal Mask Transform ("OGL + Transform") approach did not show the same scaling behavior as "RRR + OGL", performing worse at higher temporal resolutions, when compared to "RandAug + Kinetics". This suggests that the soft inductive bias imposed by RRR is superior to masking features altogether, especially when mask quality is poor. Together, these methods demonstrate the efficacy of optical flow as ground truth masks in the EGL framework, though they represent strong feature engineering that contradicts the premise of autonomous feature discovery. Nevertheless, FlowSAM generated useful attention masks despite notable artifacts in some frames (Figure 5.3) and the unsophisticated diver mask identification method described in Section 4.3.1. However, there are several other factors that could explain these results.

For instance, the temporal nature of diving videos creates a fundamental mismatch between our static segmentation masks and dynamic diving actions. As shown in Figure 5.3, high GradCAM activations appear primarily during acrobatic manoeuvrers, while our ground truth masks lack temporal specificity, marking the diver's presence regardless of action relevance. This creates competing optimization objectives: Dice loss pushes toward uniform attention on the athlete, while cross-entropy loss promotes selective attention on discriminative temporal features. The inability to satisfy both objectives simultaneously could explain why varying loss weights had little effect on attention alignment. This finding aligns with our prediction that GradCAM and our ground truth attention maps maybe fundamentally misaligned; see section 4.3.2.

Furthermore, additional tests reveal (Figure 5.2b) that changing β (the Dice loss contribution) had no effect on attention map alignment metrics, pointing to fundamental misalignment between the model's GradCAM outputs and ground truth attention maps. While GradCAM produces interpretable visualizations of model attention, it only provides a simplified representation of the model's internal state and may not align with the model's actual internal representations.

To solve this misalignment, future work could enhance the attention-based approach by jointly predicting segmentation masks and diving classes, transforming the output from $\hat{y} \in \mathbb{R}^{1\times 48}$ to $\hat{y} \in \mathbb{R}^{1\times T\times H\times W\times 48}$ by adapting UNet to video data [33]. This approach would

be particularly valuable for fine-grained action recognition benchmarks like FineDiving [52] and FineGym [38], where dive actions are segmented change within a video sample.

The Temporal Mask Transform approach offers a practical alternative to complex attention-based methods, while demonstrating the efficacy of targeted segmentation masks. However, these masks contain artifacts from macro blocking in source videos—segmentation quality deteriorates during rapid movement while remaining crisp during minimal motion. The simplistic mask selection algorithm also requires improvement. Future work could address these limitations by: 1) adopting newer versions of the Segment Anything Model, 2) utilizing higher-quality source data with fewer encoding artifacts, and 3) exploring textual prompting through CLIP to select proper masks [32, 48]. Combining textual prompts' expressiveness could unlock guiding models to learn human-object interactions in complex environments.

This work also makes an interesting observation about assumptions made by Feichtenhofer et al. [10]. For the SlowFast architecture they proposed that the fast pathway requires fewer channels while the slow pathway needs lower temporal resolution. Our findings challenge this assumption, as increasing the slow pathway's temporal resolution significantly improved performance (Figure 5.4). More precisely, our "RRR + OGL" approach achieves and accuracy of 0.812 compared to 0.776, at reduction of 56% computations. This suggests that the fast pathway may lack sufficient channel depth to encode complex motion patterns in competitive diving. While the original SlowFast channel capacity likely suffices for simpler movements in datasets like Kinetics, domains featuring rapid, technical motions may benefit from architectural adjustments. Future work could investigate whether similar improvements emerge in other fast-motion domains such as skateboarding and gymnastics. Moreover, future work could confirm our suspicion that the fast pathway lacks channel depth to capture complex motion patterns in diving acrobatics, by increasing the fast pathway's channel depth.

This work also suffers the curse of working with high-dimensional data and limited resources, imposing significant computational constraints. These constraints and training cost limited evaluation of EGL methods at higher temporal resolutions. While we explored alternative XAI approaches like LIME and SHAP, these proved computationally intensive for video data [39]. Furthermore, the proposed EGL approaches proved to be computationally expensive due to second-order gradients. Future work could explore more efficient attention mechanisms, such as incorporating a U-Net style segmentation branch that allows the model to directly predict diver segmentations for each frame, thereby eliminating the computational overhead of GradCAM attention while enabling the model to explicitly demonstrate its own spatial attention patterns. Additionally, future work could focus on optimizing these methods for high-dimensional temporal data, making advanced explanation techniques more practical for RGB video applications.

Chapter 7

Conclusion

This work addressed fundamental questions about the effectiveness of Explanation Guided Learning for action recognition in sports video analysis. This chapter concludes by reviewing the four research questions from Section 3, providing practical recommendations for future research directions, and highlighting the significant contributions this work makes to the fields of explainable AI and video understanding.

Q1: *How well does EGL improve performance on diving action recognition tasks?* EGL's performance varies significantly based on implementation: while the Dice-based approach underperformed, the Temporal Mask Transform ("OGL + Transform") method achieved a 6.67% improvement over "Kinetics + RandAug" at $T = 16$. However, the most notable improvement is that of "RRR + OGL". Which performs better than our baseline methods across all temporal resolutions. Specifically, "RRR + OGL" achieves a 2.47% improvement over "Kinetics + RandAug" and 5.06% improvement over "Vanilla + Kinetics", when averaged across temporal resolutions. This demonstrates EGL's potential while highlighting its sensitivity to implementation details.

Q2: *Can optical flow-derived segmentation masks serve as effective ground truth explanations for video-based EGL without human annotation?* This work created a novel approach using FlowSAM that derives diver masks from optical flow frames, providing an automated method for generating ground truth attention maps without the need for human annotators. Thereby addressing an important gap which limits the adoption for EGL. While the generated masks are of limited quality, especially during fast motion. And the simplistic mask selection algorithm requires improvement. The auto-generated masks proved effective in multiple contexts: the Temporal Mask Transform approach showed strong performance improvements at lower temporal resolutions, and the "RRR + OGL" method outperformed traditional augmentation techniques at higher temporal resolution. Although applying attention masks constitutes a form of feature engineering, these masks can still serve a valuable purpose in specialized domains such as sports, where fast inference time is crucial for adoption in real-time applications. This work demonstrated the potential for significantly smaller models to outperform models that require twice the computations, by training smaller models using EGL. Thus, despite limitations in mask quality and selection algorithms, our optical flow-derived masks demonstrated clear potential as a scalable alternative to human annotation for video-based EGL applications.

Q3: *To what extent does constraining model attention to align with ground truth diver segmentation masks improve both classification accuracy and explanation quality?* The attention alignment approach using Dice loss revealed fundamental challenges in constraining model attention to match segmentation masks. Despite varying loss weights (α values from 0.0 to 0.75), IoU and Dice factor metrics remained consistently low (0.0-0.1), indicating

that the model failed to align its GradCAM attention patterns with ground truth diver masks. This failure most likely stems from a temporal mismatch between GradCAM and the ground truth masks. GradCAM naturally focuses on discriminative action moments during acrobatic manoeuvres, while segmentation masks uniformly highlight diver presence regardless of action relevance. The competing optimization objectives—Dice loss promoting uniform attention on the athlete versus cross-entropy loss encouraging selective temporal attention—created irreconcilable conflicts that prevented effective learning. These findings reveal that static segmentation masks may be fundamentally incompatible with the dynamic attention patterns required for temporal action recognition.

Q4: *How do different explanation guidance approaches (attention alignment vs. input masking vs. gradient penalisation) compare in their effectiveness for fine-grained action recognition?* The three EGL strategies demonstrated different levels of effectiveness. Attention alignment through Dice loss proved least effective, suffering from the temporal mismatch issues described above and consistently underperforming across all temporal resolutions. Input masking via the Temporal Mask Transform ("OGL + Transform") approach showed strong performance at lower temporal resolutions $T = 16$, achieving the highest accuracy among all methods at this setting, but lacked the scaling behaviour observed in other approaches. Gradient penalisation through "RRR + OGL" emerged as the most promising approach, demonstrating consistent improvements over baseline methods at $T = 32$ and $T = 64$, with 3.14% and 4.70% accuracy gains respectively. The RRR approach's soft inductive bias proved superior to hard masking, particularly when mask quality was imperfect, allowing the model to selectively utilize relevant features while avoiding complete feature elimination. However, RRR is limited by high training costs induced by second-order gradients. These results suggest that gradient penalisation provides the most effective balance between guidance and model flexibility for action recognition tasks. Yet they impose significant training cost, making it difficult to recommend.

Additionally, experiments with denser temporal sampling strategies in the slow pathway revealed important insights about the SlowFast architecture's design assumptions. Decreasing the slow pathway's temporal resolution by reducing α improved performance across all methods, challenging the original architectural premise that lower temporal resolution suffices for the slow pathway. These findings suggest that while the fast pathway's limited channel depth remains appropriate for datasets containing simpler motion patterns like Kinetics, domains featuring complex, technical movements such as competitive diving require higher temporal resolution in the slow pathway to capture the nuanced motion dynamics essential for fine-grained action recognition.

7.1 Recommendations

People interested in applying EGL to video action recognition should take away the following insights. First, avoid using methods that include second-order gradients in the compute graph. For example, our Dice-based approach and our application of the RRR Loss from Ross et al. (2017) [34] show that training is 2-4 times slower compared to our baseline. This effect was not highlighted in the existing literature, likely due to the use of more "toy-like" datasets where these computational costs are less noticeable on modern hardware. These effects become exacerbated due to the high-dimensional nature of video data. While this work demonstrates the efficacy of auto-generated ground truth attention maps, the quality remains poor. This is mostly constrained by compression in the source data and a rather simple diver selection algorithm. Upon inspection of another diving dataset called FineDiving, it is clear that it contains higher quality video with fewer artifacts, though it

has fewer training examples [52]. Future work should also investigate the effect of upgrading to a newer version of Segment Anything and including CLIP for textual prompting [32, 48]. Specifically, CLIP could unlock moving object segmentation relevant objects to the domain. For example, in basketball, it could be prompted to segment the ball, hoop, and players. Therefore, combining FlowSAM with CLIP could be a powerful method for generating ground truth attention masks without the need for human annotators.

This work makes several contributions to the field: 1) the first application of EGL to sports action recognition, 2) novel automated attention map generation using optical flow, 3) incompatibilities between spatial segmentation and temporal attention, 4) comparative analysis establishing gradient penalisation as the most effective EGL strategy for video tasks, and 5) architectural insights challenging established assumptions about temporal resolution requirements in temporally challenging domains such diving.

These contributions advance our understanding of EGL's applicability to video tasks while identifying critical areas for future research, particularly in developing explanation methods that respect the temporal dynamics essential for action recognition.

Chapter 8

Acknowledgements

For this work I'd like thank my supervisor Alexia Briassouli for her guidance, and feedback during this thesis, as well as helping me with obtaining the necessary resources to make this work possible. I'd like to thank my chair Tom van Dijk for his time and effort and assisting me and Alexia throughout the whole process of the Master Thesis. Ewout van Nimwegen for the endless supply of free coffee, and fun breaks during hard thesis work. In extension, I'd like to thank everybody who sat at the thesis table in Langezijds, for all the thesis related/unrelated discussions and good times that made all the hard work a bit easier (you know how you are!). Special thanks to the friends who read my thesis and gave feedback: Nick Wolters, and Bas van der Kaaden. And last but not least, my girlfriend for all the emotional support.

Appendix A

Supplementary Figures

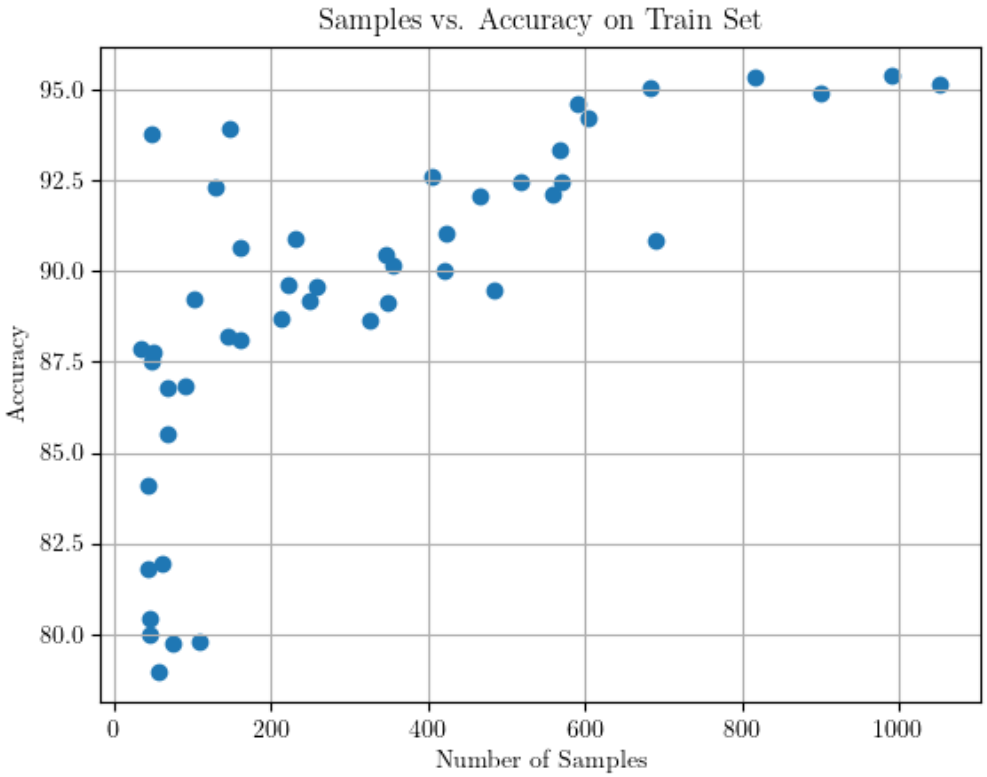

FIGURE A.1: This figure shows the the number of samples vs the accuracy on the training split. In essence this shows the expected behaviour as more samples reduce variance in the performance [31]. Furthermore, more samples also give higher accuracy. Interesting, is that beyond 600 samples the accuracy does not increase that much.

FIGURE A.2: This figure displays several sample frames from the Diving48 dataset. The figure displays for frames from a single dive sequence. For each frame we display a RGB video frames top-left, optical flow frames, top-right, candidate masks bottom left, and diver binary masks bottom right. This figure is an extended version of Figure 4.2

Bibliography

[1] Shivangi Anthwal and Dinesh Ganotra. An Overview of Optical Flow-based Approaches for Motion Segmentation. *The Imaging Science Journal*, 67(5):284–294, 2019.

[2] Reza Azad, Moein Heidary, Kadir Yilmaz, Michael Hüttemann, Sanaz Karimijafarbigloo, Yuli Wu, Anke Schmeink, and Dorit Merhof. Loss functions in the era of semantic segmentation: A survey and outlook. *arXiv preprint arXiv:2312.05391*, 2023.

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is Space-Time Attention All You Need for Video Understanding? In *ICML*, volume 2, page 4, 2021.

[4] Jeremy Bloom. AI Experiment in Halfpipe Judging at X Games Will Give Snowboarders a Glimpse into the Future. X Games Partnership with Google Cloud, January 2025. Reported in Associated Press, NPR, and other outlets.

[5] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A Short Note about Kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018.

[6] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A Short Note on the Kinetics-700 Human Action Dataset. *arXiv preprint arXiv:1907.06987*, 2019.

[7] Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[8] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. RandAugment: Practical Automated Data Augmentation with a Reduced Search Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

[10] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast Networks for Video Recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6202–6211, 2019.

[11] FIFA. Semi-automated Offside Technology to be Used at FIFA World Cup 2022, 2022. URL: https://inside.fifa.com/media-releases/semi-automated-offside-technology-to-be-used-at-fifa-world-cup-2022-tm.

[12] Felix Friedrich, Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting. A Typology for Exploring the Mitigation of Shortcut Behaviour. *Nature Machine Intelligence*, 5(3):319–330, 2023.

[13] Fujitsu Limited. Fujitsu and the International Gymnastics Federation launch AI-powered Fujitsu Judging Support System for use in Competition for all 10 Apparatuses, October 2023. URL: https://www.fujitsu.com/global/about/resources/news/press-releases/2023/1005-02.html.

[14] Yuyang Gao, Siyi Gu, Junji Jiang, Sungsoo Ray Hong, Dazhou Yu, and Liang Zhao. Going Beyond XAI: A Systematic Survey for Explanation-Guided Learning. *ACM Computing Surveys*, 56(7):1–39, 2024.

[15] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018.

[16] Liam Hiley, Alun Preece, Yulia Hicks, Supriyo Chakraborty, Prudhvi Gurram, and Richard Tomsett. Explaining Motion Relevance for Activity Recognition in Video Deep Learning Models, 2020. URL: https://arxiv.org/abs/2003.14285, arXiv:2003.14285.

[17] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-Scale Video Classification with Convolutional Neural Networks. In *CVPR*, 2014.

[18] 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.

[19] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment Anything. *arXiv:2304.02643*, 2023.

[20] Michael Lewis. *Moneyball: The Art of Winning an Unfair Game*. W. W. Norton & Company, New York, 2003.

[21] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards Action Recognition without Representation Bias. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 513–528, 2018.

[22] Fujitsu Limited. Fujitsu and International Gymnastics Federation Announce Collaboration to Develop AI-based Judging Support System, October 2017. URL: https://www.fujitsu.com/global/about/resources/news/press-releases/2017/1007-01.html.

[23] Zachary C Lipton. The Mythos of Model Interpretability: In Machine Learning, the Concept of Interpretability Is Both Important and Slippery. *Queue*, 16(3):31–57, 2018.

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[25] Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, 30, 2017.

[26] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the Effective Receptive Field in Deep Convolutional Neural Networks. *Advances in neural information processing systems*, 29, 2016.

[27] N. Owens. Hawk-Eye Tennis System. In *International Conference on Visual Information Engineering (VIE 2003). Ideas, Applications, Experience*, pages 182–185. IEE, 2003. doi:10.1049/cp:20030517.

[28] Paritosh Parmar, Amol Gharat, and Helge Rhodin. Domain Knowledge-Informed Self-Supervised Representations for Workout Form Assessment. In *European Conference on Computer Vision*, pages 105–123. Springer, 2022.

[29] Perry Pierce, Matthew Whitrock, and Stuart Cheshire. Developing MLB's Automated Ball/Strike System (ABS). *MLB Technology Blog*, January 2022. URL: https://technology.mlblogs.com/developing-mlbs-automated-ball-strike-system-abs-d4f499deff31.

[30] Will Price and Dima Damen. Play Fair: Frame Attributions in Video Models. In *Computer Vision–ACCV 2020: 15th Asian Conference on Computer Vision, Kyoto, Japan, November 30–December 4, 2020, Revised Selected Papers, Part V 15*, pages 480–497. Springer, 2021.

[31] Simon J.D. Prince. *Understanding Deep Learning*. The MIT Press, 2023. URL: http://udlbook.com.

[32] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment Anything in Images and Videos. *arXiv preprint arXiv:2408.00714*, 2024. URL: https://arxiv.org/abs/2408.00714.

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[34] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. Right for the Right Reasons: Training Differentiable Models by Constraining Their Explanations. *arXiv preprint arXiv:1703.03717*, 2017.

[35] Patrick Schramowski, Wolfgang Stammer, Stefano Teso, Anna Brugger, Franziska Herbert, Xiaoting Shao, Hans-Georg Luigs, Anne-Katrin Mahlein, and Kristian Kersting. Making Deep Neural Networks Right for the Right Scientific Reasons by Interacting with Their Explanations. *Nature Machine Intelligence*, 2(8):476–486, 2020.

[36] 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. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.

[37] Ramprasaath R Selvaraju, Stefan Lee, Yilin Shen, Hongxia Jin, Shalini Ghosh, Larry Heck, Dhruv Batra, and Devi Parikh. Taking a Hint: Leveraging Explanations to

Make Vision and Language Models More Grounded. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2591–2600, 2019.

[38] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. FineGym: A Hierarchical Video Dataset for Fine-Grained Action Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[39] Sameer Singh, Marco Tulio Ribeiro, and Carlos Guestrin. Programs as Black-Box Explanations. *arXiv preprint arXiv:1611.07579*, 2016.

[40] Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting. Right for the Right Concept: Revising Neuro-Symbolic Concepts by Interacting with Their Explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3619–3629, 2021.

[41] Alexandros Stergiou, Georgios Kapidis, Grigorios Kalliatakis, Christos Chrysoulas, Remco Veltkamp, and Ronald Poppe. Saliency Tubes: Visual Explanations for Spatio-Temporal Convolutions. In *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, September 2019. URL: http://dx.doi.org/10.1109/ICIP.2019.8803153, doi:10.1109/icip.2019.8803153.

[42] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, Proceedings 3*, pages 240–248. Springer, 2017.

[43] Huawei Sun, Lorenzo Servadei, Hao Feng, Michael Stephan, Avik Santra, and Robert Wille. Utilizing Explainable AI for Improving the Performance of Neural Networks. In *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1775–1782. IEEE, 2022.

[44] Zachary Teed and Jia Deng. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.

[45] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[46] Tomoki Uchiyama, Naoya Sogi, Satoshi Iizuka, Koichiro Niinuma, and Kazuhiro Fukui. Adaptive Occlusion Sensitivity Analysis for Visually Explaining Video Recognition Networks, 2023. URL: https://arxiv.org/abs/2207.12859, arXiv:2207.12859.

[47] Corne Van Zyl, Xianming Ye, and Raj Naidoo. Harnessing eXplainable Artificial Intelligence for Feature Selection in Time Series Energy Forecasting: A Comparative Analysis of Grad-CAM and SHAP. *Applied Energy*, 353:122079, 2024.

[48] Haoxiang Wang, Pavan Kumar Anasosalu Vasu, Fartash Faghri, Raviteja Vemulapalli, Mehrdad Farajtabar, Sachin Mehta, Mohammad Rastegari, Oncel Tuzel, and Hadi

Pouransari. SAM-CLIP: Merging Vision Foundation Models Towards Semantic and Spatial Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3635–3647, 2024.

[49] Rui Wang, Dongdong Chen, Zuxuan Wu, Yinpeng Chen, Xiyang Dai, Mengchen Liu, Yu-Gang Jiang, Luowei Zhou, and Lu Yuan. BEVt: BERT Pretraining of Video Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14733–14743, 2022.

[50] Meredith Wills. AI Is Helping Referee Games in Major Sports Leagues, but Limitations Remain, May 2024. URL: https://www.scientificamerican.com/article/ai-is-helping-referee-games-in-major-sports-leagues-but-limitations-remain/.

[51] Junyu Xie, Charig Yang, Weidi Xie, and Andrew Zisserman. Moving Object Segmentation: All You Need Is SAM (and Flow). In *ACCV*, 2024.

[52] Jinglin Xu, Yongming Rao, Xumin Yu, Guangyi Chen, Jie Zhou, and Jiwen Lu. FineDiving: A Fine-Grained Dataset for Procedure-Aware Action Quality Assessment. In *CVPR*, pages 2949–2958, 2022.

[53] Jinglin Xu, Sibo Yin, Guohao Zhao, Zishuo Wang, and Yuxin Peng. FineParser: A Fine-Grained Spatio-Temporal Action Parser for Human-Centric Action Quality Assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14628–14637, 2024.

[54] Michael Yeung, Evis Sala, Carola-Bibiane Schönlieb, and Leonardo Rundo. Unified Focal Loss: Generalising Dice and Cross Entropy-based Losses to Handle Class Imbalanced Medical Image Segmentation. *Computerized Medical Imaging and Graphics*, 95:102026, 2022.