Metode de Explicabilitate în Recunoașterea Acțiunilor Umane bazata pe Date Scheletice

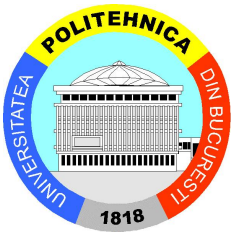
---

Autor:
Cristi Andrei Prioteasa

Supervizor:
As. Dr. Ing. Mihai Nan

Proiect de diplomă

Departamentul Calculatoare
Facultatea de Automatică si Calculatoare



Universitatea Politehnica din București
București, România

Iulie 2023

Explainability Methods in Human Action Recognition based on Skeletal Joint Data

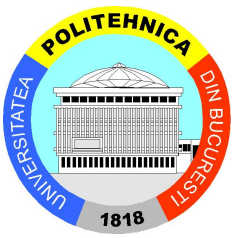
---

Author:
Cristi Andrei Prioteasa

Supervisor:
As. Dr. Ing. Mihai Nan

Diploma Project

Computer Science and Engineering Department
Faculty of Automatic Control and Computer Science



University Politehnica of Bucharest
Bucharest, Romania

July 2023

Author:
Cristi Andrei Prioteasa

Supervisors:
As. Dr. Ing. Mihai Nan

Institute:
University Politehnica of Bucharest, Romania

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF LISTINGS

SINOPSIS

Progresele recente în Învățarea Automată și, mai ales, în Învățarea Profundă, au creat un mediu fără precedent pentru cercetare și inovare. Succesul din spatele Rețelelor Neuronale Profunde depinde de învățarea unor caracteristici conceptuale într-o reprezentare interna care pot reprezenta cei mai importanți factori din datele de antrenare. Cu toate acestea, lipsa de interpretabilitate a acestor caracteristici complexe poate genera probleme greu de diagnosticat. Când aceste sisteme eșuează, de obicei o fac fără avertisment sau explicație. Eșecurile inexplicabile reprezintă o provocare semnificativă în contextul utilizării Inteligenței Artificiale în sectoare critice, cum ar fi industria sănătății sau aplicațiile militare. Înțelegerea motivelor care stau la baza predicțiilor este un pas cheie în evaluarea încrederii și garantarea unor rezultate corecte și etice.

În această lucrare, abordăm nevoia de explicabilitate în contextul problemei recunoașterii acțiunii umane pe baza datelor scheletice. Propunem o adaptare a *Gradient-weighted Class Activation Mapping*, o metodă concepută inițial pentru a crea explicații vizuale pentru imagini, pentru a funcționa eficient pe date scheletice. Rezultatele noastre evidențiază cele mai informative articulații din fiecare cadru al scheletelor de intrare, ilustrând astfel că reprezentările profunde se aliniază cu intuiția umană. Apoi, examinăm influența stratului țintă ales pentru extragerea gradienților și importanța modulului de atenție într-o arhitectură *ResGCNv2.0*. În cele din urmă, efectuăm o analiză comparativă între *Class Activation Maps* și *Gradient-weighted Class Activation Mapping* și prezentăm rezultatele studiului nostru de acord.

ABSTRACT

The recent advances in Machine Learning and, in turn, Deep Learning, have created an unprecedented environment for research and innovation. The success behind Deep Neural Networks depends on learning hidden state high-level features that can represent the most important factors in the training data. However, the lack of interpretability of these complex features can generate hard to diagnose problems. When these systems fail, they usually do so without a warning or explanation. Unexplained failures pose a significant challenge in the context of utilizing AI in critical sectors such as the Health Industry or Military Applications. Understanding the underlying reasons behind predictions is a key step in assessing trust and guaranteeing fair and ethical outcomes.

In this thesis, we address the need for explainability in the context of Human Action Recognition based on Skeleton Data. We adapt Gradient-weighted Class Activation Mapping, a method originally designed to create visual explanations for images, to function effectively with Skeleton Data. Our results highlight the most informative joints in each frame of the input skeletons, in this way revealing that deep representations align with human intuition. We then examine the influence of the chosen target layer and the importance of the attention module in a *ResGCNv2.0* architecture. Finally, we conduct a comparative analysis between Class Activation Maps and Gradient-weighted Class Activation Mapping and present results from our agreement study.

STATEMENT OF AUTHORSHIP

I, Cristi Andrei Prioteasa, born July 7, 2000 in Craiova, declare that this thesis titled *Explainability Methods in Human Action Recognition based on Skeletal Joint Data* and the work presented in it are my own.

Except where reference is made in the text of the thesis, this thesis contains no material published elsewhere or extracted in whole or in part from a thesis accepted for the award of any other degree or diploma. No other person's work has been used without due acknowledgment in the main text of the thesis.

Cristi Andrei Prioteasa July 1, 2023

ACKNOWLEDGMENTS

During my time at University Politehnica of Bucharest, I had the opportunity of interacting with many passionate and curiosity driven people, whose hard work often passes unrecognized. This is a thank you to you all!

Firstly, I would like to thank my supervisor **As. Dr. Ing. Mihai Nan** for his expertise, patience, and availability to answer all of my questions that, on many occasions, had little to do with the thesis itself.

I also want to thank my family for their support and for placing their trust in my academic pursuit.

DEDICATION

to my high school mathematics tutor, Piciu Adrian. It is from you that I have learned how to be truly free in my thinking.

ACRONYMS

CNN	Convolutional Neural Networks
AI	Artificial Intelligence
XAI	Explainable Artificial Intelligence
CAM	Class Activation Maps
grad-CAM	Gradient-weighted Class Activation Mapping
HAR	Human Action Recognition
VQA	Visual Question Answering
GAP	Global Average Pooling
GCN	Graph Convolutional Network
SGC	Spatial Graph Convolutional
TC	Temporal Convolutional
ST-JointAtt	Spatial Temporal Joint Attention
FC	Fully Connected

KEY TERMS

Term	Description
AI	Artificial intelligence (AI) is a type of intelligence possessed by machines which allows them to complete tasks which have been previously associated with human or animal intelligence, such as vision perception, natural language processing, social interaction, decision-making, or planning.
ML	Machine learning (ML) involves teaching machines to learn from large amounts of data and improve their performance over time without being explicitly programmed.
XAI	Explainable Artificial Intelligence (XAI) is a set of techniques and methods which allow deep exploration and visualization of the inner working mechanisms of AI models and create insight into how such models reach their prediction.
HAR	Human Action Recognition (HAR) is a visual classification task which aims to recognize and distinguish human actions in the form of a video or skeletal data.
CNN	Convolutional Neural Network (CNN) is a type of Deep Learning Architecture traditionally used for Visual Classification Tasks which employs a series of Convolutional Layers and Pooling Layers. The convolutional layers use the correlation operation to automatically extract the relevant features from the input data. The pooling layers reduce the input of the activations (and thus the number of parameters) and can act as a regularization method, since local patches of the input images tend to have a similar structure.
CAM	Class Activation Maps (CAM [1]) is an activation based XAI method used for visualizing the most important regions of an image that lead to the target prediction. Although, traditionally designed to work with image inputs, the underlying concept of using a weighted sum of activation maps to illustrate the importance of each feature in the classification process can be adapted to work on any kind of classification task based on CNNs.

Term	Description
grad-CAM	Gradient Weighted Class Activation Maps (grad-CAM [2]) is an XAI method used for visualizing the most important regions of an image that lead to the target prediction. It is a strict generalization of CAM which allows combining the activations with the gradients used in the backpropagation process, in this way generating more fine-grained heatmaps which better illustrate the influence of each region of the input image in the classification process.
Interpretability (XAI)	Interpretability is a characteristic of any XAI method which illustrates the degree to which the visualizations created can be easily understood by the end user. There is a trade-off between the degree of interpretability and faithfulness.
Faithfulness (XAI)	Faithfulness is a characteristic of any XAI method which illustrates how much the created visualization describes the original decision process of the model. There is a trade-off between the degree of interpretability and faithfulness. A highly faithful method (i.e., which describes in detail the inner working process of the model) is harder to interpret (and may require a more advanced target audience) and vice versa.
Black-box model	A Black-Box model is a category of Deep Learning Architectures whose inner working process is difficult to explain or interpret because of the incredibly large number of parameters, layers and general level of complexity. Thus, these model fail to provide an interpretable explanation for their prediction.
GCN	Graph Convolutional Networks are neural networks that leverage the properties of graph structured data. GCNs are able to capture both local and global characteristics of the graph, making them well suited for Human Action Recognition based on Skeletal Joints.

1 INTRODUCTION

With the ever-increasing presence of Artificial Intelligence based systems in many critical parts of our society, such as the Healthcare Industry, Finance, or Education, there has been a growing interest towards creating ways to gain insight into how these systems work, in order to build trust and diagnose failure modes. The task of understanding and interpreting the decision process of Artificial Intelligence models and, more recently, the ubiquitous Deep Neural Networks models has come to be known as Explainable Artificial Intelligence (XAI). This task is relatively easy for some types of models which have inherent interpretable characteristics (e.g. Linear Regression, SVM, Decision Trees or Bayesian Networks), but has proven to be a complex and interdisciplinary problem in the case of Deep Neural Networks, mainly because of the "Black Box" characteristic and the high number of dimensions of the input data which interact in intricate ways. This requires profound understanding of both statistical techniques and the context of the proposed problem.

In this chapter we introduce the need for explainability in the case of Human Action Recognition systems as a way to create trust, diagnose failure modes and identify biased results. We present the social implications of the lack of transparency in AI models, general objectives of explainability techniques and the motivation for creating visualizations of the decision process of such models. We establish the goals of our technique and describe our proposed solution. We present the main problems that arise from trying to create semantic interpretation of the high-level features learned by Deep Learning models and explore ways to address them. We state the identified problem and describe our proposed solution. We outline the results of our approach and present relevant metrics. The end of the chapter describes the paper structure.

1.1 CONTEXT

Although some [1] critique the need for Explainable AI [3], arguing that an evidence-based approach is sufficient even when the inner working mechanisms of the model are not fully understood, there is a growing consensus [4, 5] that Explainability should become a priority if AI agents are ever to act independently of human oversight or be trusted with critical decisions such as medical diagnosis or terrorism detection.

1 https://hai.stanford.edu/news/should-ai-models-be-explainable-depends, last accessed on 31 March 2023

Human-Action Recognition has been a long-standing important research direction in Computer Vision. Due to the recent availability of affordable commercial sensors for estimating human skeletons and the remarkable evolution of computing devices, especially GPUs, there is a growing interest in developing new models which can address problems such as Patient Analysis, Video Indexing or Security Surveillance. Skeleton data has become the widely accepted type of representation for Human Action Recognition due to its simplicity, its ability to capture the motor behaviour and dynamics of real persons and memory efficiency. Skeleton data captures the trajectories of human skeleton joints and can be both view and illumination invariant. Older HARs based on machine learning had an irreversible information loss in their prediction process (from the input data to the output features), which made it impossible to visualize the reason behind the prediction. Most HARs today are based on a Deep Learning architectures, which holds the high-level conceptual feature maps inside the network, thus an inverse propagation process is theoretically possible, which may offer insight into the prediction process. Nevertheless, proper methodology for semantic interpretation of these high-level extracted features and visualization techniques is still in early stages and existing methods are highly dependent on the chosen architecture.

1.2 PROBLEM

Human Action Recognition remains a hard to solve problem, because of its many degrees of freedom, action, and subject flexibility. Although the field of Explainable AI has seen an increasing interest over the past 10 years, most available visualization and extraction techniques were traditionally designed to work on image or video input data. We note the limited available model-agnostic visualization methods for Human Action Recognition based on Skeleton Joint Data. Many existing techniques, either completely discard the spatial or the temporal component of such models, in this way loosing faithfulness, or compute visualizations based on intermediary image-like representations of the skeleton data ([6, 7]).

As Skeleton Joint Data presents many advantages in terms of performance and anonymity (i.e., skeletal data represents a way to model and predict the action performed by an individual, without disclosing characteristic features that may lead to identification) there is an emerging need to adapt the existing state-of-the-art explainability techniques, traditionally designed to work on image date, for skeleton data. This process requires both understanding of the intuition behind state-of-the-art XAI techniques and understanding of the traded-off between the spatial and the temporal component in Human Action Recognition (HAR) architectures.

1.3 OBJECTIVES

In this paper, our objective is to establish an adaptation of Gradient-weighted Class Activation Mapping that works directly on Skeleton Data, without employing an intermediary pseudo-image representation of the input data. We aim to create an adaptation of grad-CAM that works primarily on models of the EfficientGCN family (e.g. *ResGCNv2.0*), but that can applied to different Graph Convolutional Network with Skeleton Data as input with little modification. In creating our adapted version of grad-CAM we propose the following objectives:

1. To analyse the most important differences between Skeleton and Image Data and understand the trade-off between Interpretability and Faithfulness.
2. To explore the influence of the selected target layer, from which we extract the gradients that are used as the weights in the grad-CAM computation process, in terms of spatial and temporal localization.
3. To explore the spatial and temporal attention capabilities added by the Spatial Temporal Joint Attention module proposed by Yi-Fan Song et al. [8].
4. To compare the visualizations generated by Class Activation Maps with those generated by Gradient-weighted Class Activation Mapping.
5. To provide intuitive explanations on the most common failure cases of our model.
6. To conduct an agreement analysis involving non-STEM subjects, tasked with identifying the correct performed action based solely on skeleton data.

1.4 SOLUTION

In this paper, we aim to address the lack of specific visualization techniques for the Human Action Recognition based on Skeletal Joints, by adapting state-of-the-art explainability methods, traditionally designed to work on image data. Existing methods [6, 7] create an intermediary pseudo-image representation of the skeletal joint data which they feed to their specific network architecture, in this way facilitating the use of image-level explainability methods. We propose a different method of adapting grad-CAM to work directly on a skeletal input data and test our approach on a ResGCNv2.0 model [8]. In this sense, our work represents a novel approach.

We decide to use grad-CAM for our visualization, since it is a model-agnostic technique (it requires no additional architectural changes, in contrast to CAM) and creates smoother visualizations. This is desirable in the case of HAR as we want to be able to visualize the general importance of each joint over the entire duration of the action and do not require extremely precise computation of the associated weights. Through our results, we create valuable insight into the decision process of the model, we offer intuitive explanations

on the failure modes, and we are able to determine how the temporal and spatial focus is correlated with the target layer of a ResGCNv2.0 architecture. We test our approaches on multiple configurations of the ResGCNv2.0 architecture [8] and conduct a comparative analysis between CAM and Gradient-weighted Class Activation Mapping illustrating the differences between these visualization techniques. Finally, we conduct an agreement analysis which aims to illustrate the ambiguous nature of certain action classes represented through skeletal data.

1.5 RESULTS

We successfully adapted grad-CAM to work directly on Skeleton Data in the case of two distinct pre-trained configurations of the *ResGCNv2.0* architecture and created visualizations that help explain the decision process of the model. Using the generated visualizations, we interpreted the failure modes of the model, offering intuitive explanations. We interpreted the visualizations created by CAM and grad-CAM and note their differences. We analysed the influence of the selected target layer and confirmed the importance of the Spatial Temporal Joint Attention module [8]. We conducted an agreement analysis that illustrates how certain action classes are hard to identify both by the model and human subjects, based solely on skeleton data.

1.6 THESIS OUTLINE

The remainder of this thesis is organized as follows:

Chapter 2 provides general background information about the state of Artificial Intelligence today and the most important aspects regarding Explainability. We begin by briefly explaining why Deep Neural Networks have become the widely accepted standard for most Machine Learning tasks, their advantages and inherent lack of interpretability. We continue to present the motivation behind XAI and the trade-off between Interpretability and Faithfulness. We outline the importance of XAI at any stage in the development process and how it can be used to drive research.

Chapter 3 details the advantages and limitations of Convolutional Neural Networks (CNN) and two approaches for interpreting the decision process of such architectures. We present Saliency Maps as the simplest gradient-based visualization method and transition to two SOTA activation-based XAI methods: Class Activation Maps and Gradient-weighted Class Activation Mapping.

Chapter 4 explores visualization and XAI methods in the case of Human Action Recognition based on Skeleton Data. We present Motion History Images, as a starting point into interpreting the characteristics of Skeleton Data through an intermediary pseudo-image representation. Two distinct approaches which successfully use grad-CAM to highlight the most informative joints of the skeleton in the performed action are presented.

Chapter 5 details how grad-CAM, traditionally designed to work on image input data, is adapted to work on skeleton data directly, without using an intermediary pseudo-image representation. Firstly, CAM is presented as a more architectural restrictive explainability method. Then, the adaptation of grad-CAM is described. We end the chapter with a comparative analysis between image and skeleton data.

Chapter 6 details characteristics of the chosen target model. The main building blocks of ResGCNv2.0 [8] are presented, its advantages over other overly-parametrized SOTA models and dimensions worth exploring using our methods. A brief overview of the grad-CAM extraction and visualization process is discussed.

Chapter 7 details testing of our proposed adaptation of CAM and grad-CAM in an experimental setting. We explore the influence of the chosen target layer in the gradient extraction process, we present how visualizations generated by grad-CAM provide valuable intuitive explanations for the failure cases. We discuss the differences between CAM and grad-CAM in terms of spatial and temporal localization. We conduct an agreement analysis for the most common misclassified predictions, illustrating that both human subjects and the model have difficulties in predicting certain action classes based solely on skeleton data.

Chapter 8 concludes this thesis. The main achieved objectives are outlined and the most important conclusions are discussed.

2

BACKGROUND

In this chapter, we present a short overview of the evolution of Artificial Intelligence and outline the most important ideas in XAI. We illustrate the motivation behind Explainable Artificial Intelligence and how it can help in diagnosing failure modes that would otherwise be incredibly difficult to detect. We discuss the characteristics of XAI methods and explore the trade-off between Interpretability and Faithfulness.

2.1 EVOLUTION OF ARTIFICIAL INTELLIGENCE

Before the shift towards black box models used today, such as Artificial Neural Networks or Deep Neural Networks, which offer little to no explanation for their decision process, the Artificial Intelligence (AI) research in the late 1970s was oriented towards rule-based expert systems. These were systems that would follow precise rules according to some criterion, but never attained expert-level performance [9]. Peter J. Denning and John Arquilla [10] conclude that this type of failure is bound to happen because humans do not use known rules to guide their actions and, moreover, in many cases the actions they take cannot be described by any rule. Since then, the widely-accepted AI model has become some variation of Neural Network, which outperforms the older rule-based systems, but lacks in interpretability. Figures 2.1 and 2.2 illustrate a conceptual view of model-agnostic interpretability.

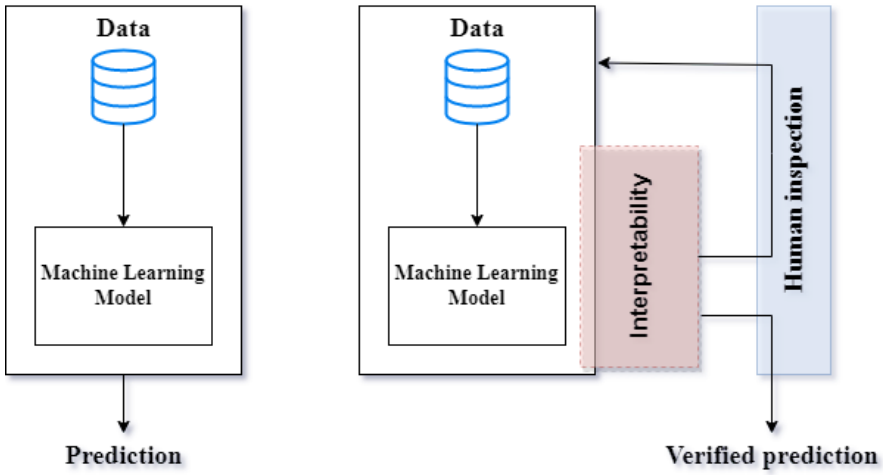


Figure 2.1: Traditional training pipeline vs Interpretable training pipeline.

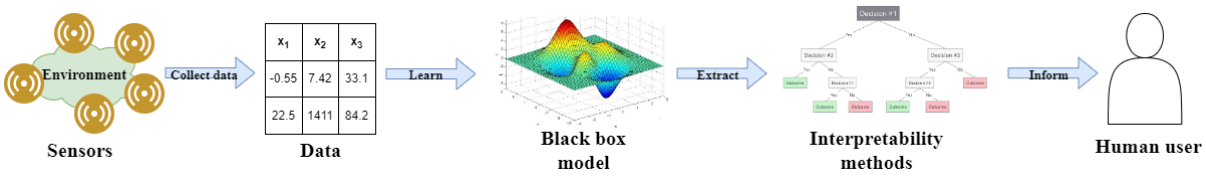


Figure 2.2: High level model-agnostic interpretability overview

2.2 DIAGNOSING FAILURE MODES

The lack of transparency in today's ubiquitous Deep Neural Networks can generate hard to diagnose problems. When these systems fail, they usually do so without a warning or explanation. A good example for the need of an explainable model is described in the work of Kaufman et al. [11]. They discovered that, in a model designed for medical diagnosis, there was a heavy correlation between the target class and the patient's ID (Data leakage). Such an issue is considerably hard to identify just by looking at the predictions and the raw training data, but would be relatively easy to diagnose using an interpretable model, which would suggest that its decision process was influenced by the patient's ID. Selvaraju et al. [2] illustrate how their technique, called grad-CAM, can be used in the context of CNNs to identify biased models. Grad-CAM visualizations revealed that a model trained to distinguish between doctors and nurses learned to look at a person's face and hairstyle in order to determine the correct class, thus promoting a gender stereotype (Figure 2.3). Through the intuition gained from this visualization technique, they identified that the dataset was gender-biased (78% of images with doctors were men and 93% of images with nurses were women [2]). By correcting the dataset they were able to obtain an 8% increase in accuracy, but, most importantly, they were able to guarantee fair and ethical outcomes for the model.

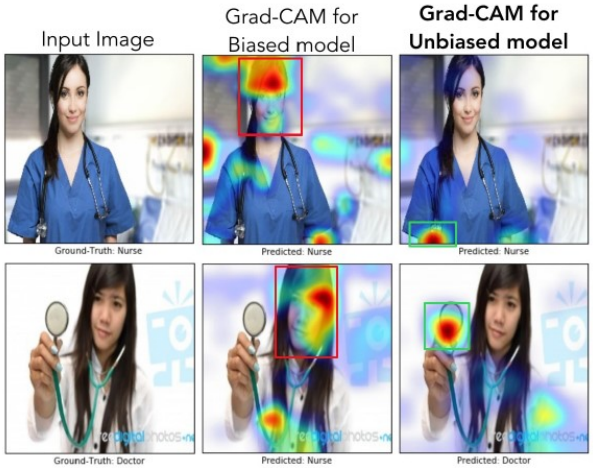


Figure 2.3: The visualization provided by grad-CAM is able to show that the biased model is looking at the person's face in order to distinguish between a doctor and a nurse, thus promoting a gender-bias, while the unbiased model looks at the coat and the stethoscope. Extracted from 'Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization'

2.3 INTERPRETABILITY – FAITHFULNESS TRADE-OFF

Ribeiro et al. [5] define an explanation model as $g \in G$, where G is the superset of interpretable models (linear models, decision trees etc.). The model g can act over the whole domain of the initial model $\{0,1\}^d$, or on a subset of interpretable components $\{0,1\}^{d'}$. The complexity of g is defined as $\Omega(g)$ (e.g. for decision trees it may be the depth and for linear models it may be the number of weights). As a general rule, the more complex an explainable model is (Faithfulness) the harder it is to interpret (Interpretability). We may argue that a completely faithful explanation is the entire model, but such a holistic approach is hardly interpretable in the case of most Neural Network Architectures.

Let the model that we want to explain be denoted by $f : R^d \to R$. In classification f is the probability that x belongs to a certain class. Let $\pi_x(z)$ be a proximity measure between an instance z to x. This represents the vicinity of x. Finally, the measure of unfaithfulness in approximating f with g in the vicinity defined by π_x will be denoted by:

$$L(f, g, \pi_x)$$

Using these notations, finding an explanation for the model f that is both locally faithful and interpretable is equivalent to minimizing L while having $\Omega(g)$ low enough to be interpreted by humans. We now have a general mathematical model for dealing with interpretability. This model can be used with different explanations and complexity thresholds:

$$\zeta(x) = \text{argmin}_{g \in G}(L(f, g, \pi_x) + \Omega(g)) \quad [5]$$

Different thresholds for $\Omega(g)$ may be used, accounting for the knowledge of the end user. For example, a Machine Learning Engineer may prefer a model with a higher degree of faithfulness, while an end user should be presented with a more interpretable explanation.

Ribeiro's work [5] offers a new way of thinking about interpreting a complex model. Although the model itself may be too complex to explain and interpret globally, we can create different local explanations which capture how the model behaves in a local vicinity and are easier to interpret (for example, a linear approximation).

2.4 EXPLAINABILITY AT DIFFERENT STAGES

R. Selvaraju et al. [2] describe three separate stages of a model's evolution where the ability to interpret its decision is useful, thus illustrating the importance of interpretability regardless of how performant the model may be:

1. **Incipient stage**: the model is significantly weaker than a human, and it is not yet ready to be deployed (e.g. Visual Question Answering (VQA)). In this stage, the purpose of the explanations are to guide researches in the right direction for improvement.

2. **On par stage**: the model's performance is on par with a human (e.g. Image classification [1]). Here, the goal of explainability is to establish trust and confidence in the model.

3. **Superior stage**: the model outperforms any human capable of doing the same task (e.g. AlphaGo [2]). In this stage, finding ways to interpret and explain the model creates valuable insight that help humans make better decisions.

We can also think of interpretability techniques as dependent or independent to the model's architecture. A model dependent technique would involve modifying an existing architecture to make it more transparent and interpretable, but may yield worse results, whereas a model agnostic approach would work without any modification of the model on which it is applied. The latter is usually much harder to implement, as different architectures require different ways of interpreting the result, thus, a compromise is usually to create a technique which works for a range of models, such as Convolutional Neural Networks.

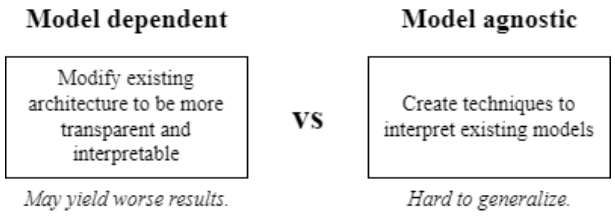


Figure 2.4: Model dependent vs model agnostic explainability.

Adadi et al. [12] conclude that there are many dimensions worth exploring in the case of Explainability. Their proposal for a new taxonomy aims to describe at which stage of the prediction process, interpretation should be conducted:

1. **Justification**: aims to offer a justification for the outcome without a description of the inner-working mechanisms of the model. Considering that numerous cases of biased decisions against groups of individuals have been reported [13] and new regulations have been established under the *"Right to explanation"* [3], a need for justification of any result has become essential.

2. **Control**: aims to offer insight that enhances the control over the model. This comes to particular interest in the case of failure modes, which are particularly hard to diagnose for Neural Networks. Provided a visualization for the model's decision, predictions that seem unreasonable may, very well, have a reasonable explanation [2].

3. **Improvement**: aims to offer insight on the model's inner working mechanisms as a way to identify redundant or inefficient characteristics and improve them. Having a

1 http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/ last accessed: 6 April 2023
2 https://www.deepmind.com/research/highlighted-research/alphago last accessed: 6 April 2023
3 https://gdpr-info.eu/art-22-gdpr/ last accessed: 31 March 2023

strong understanding of why the model works in the way it does lays a safe road for future improvement and innovation.

4. **Discover**: aims to offer intuition on the behaviour of the model with respect to the input, in this way revealing inherent characteristics of the data they work on. This kind of valuable knowledge can shed light on new unknown connections and can illustrate complex patterns in the structure of the known universe.

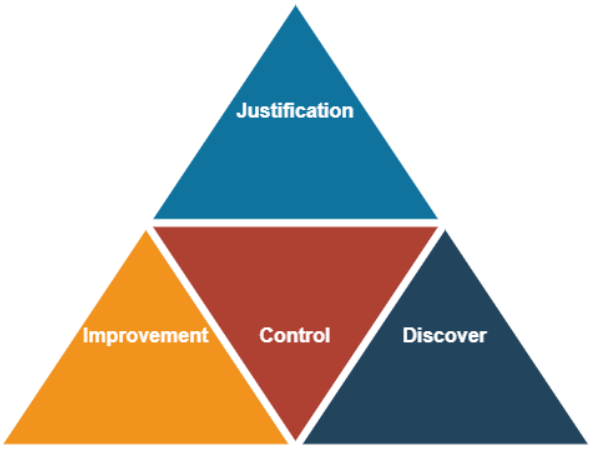


Figure 2.5: Reasons for explainability (Adadi's [12] taxonomy).

2.5 SUMMARY

In this chapter we explored the key ideas and motivation behind interpretability, we presented important research directions and objectives of XAI methods and illustrated how such methods can help correct biased models and diagnose failure modes, in this way creating trust. Finally, we discussed the trade-off between Faithfulness and Interpretability. In the next chapters, we will discuss state-of-the-art XAI techniques for CNN and we will propose a method to adapt Gradient-weighted Class Activation Mapping for Human Action Recognition based on skeleton data.

3

EXISTING METHODS

In this chapter, we summarize the key concepts of Convolutional Neural Networks, we illustrate key advantages and limitations of such architectures and transition towards methods of visualization. We explore the simple intuition behind Saliency Maps (Section § 3.3) and transition towards two state-of-the-art methods: Class Activation Maps (Section § 3.4.1) and Gradient-weighted Class Activation Mapping (Section § 3.4.2). We will later use the intuition gained from analyzing Gradient-weighted Class Activation Mapping in order to build our solution for visualizations in the context of Skeleton Based Action Recognition.

3.1 CONVOLUTIONAL NEURAL NETWORKS FOR VISUAL CLASSIFICATION

Convolutional neural networks became the widely accepted choice for any visual classification tasks due to their remarkable ability to capture high-level semantic concepts in an image. They achieve this through a series of hierarchical layers. Studies such as those of Hubel & Wiesel [14] in 1968 and, more recent, Michael A. Silver et al. [15] show that the human brain employs a similar hierarchical structure when processing visual stimuli: simple cells respond to light orientation, while more complex cells respond to movement with respect to an end-point. Moreover, nearby cells in the cortex are responsible for nearby regions in the visual field (Figure 3.1). Although the biological neuron analogy should be utilized cautiously, this stands to show that developing a model that resembles the cognitive process of the human brain is remarkably powerful.

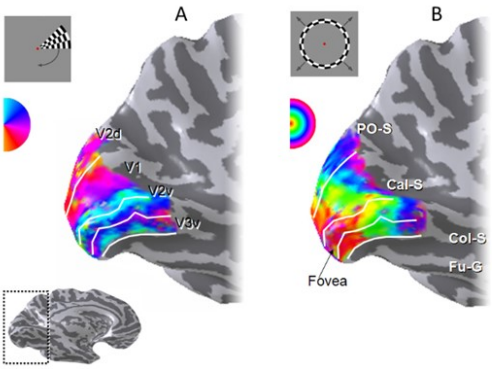


Figure 3.1: Nearby cells in the cortex are responsible for nearby regions in the visual field. 'Topographic maps in human frontal and parietal cortex' [15]

A general CNN architecture consists of a feature extraction component and a classification part (Figure 3.2). The feature extraction part, which consists of multiple layer of convolution, pooling and activation, acts as a concept discovery mechanism and learns semantic correspondences across images [1].

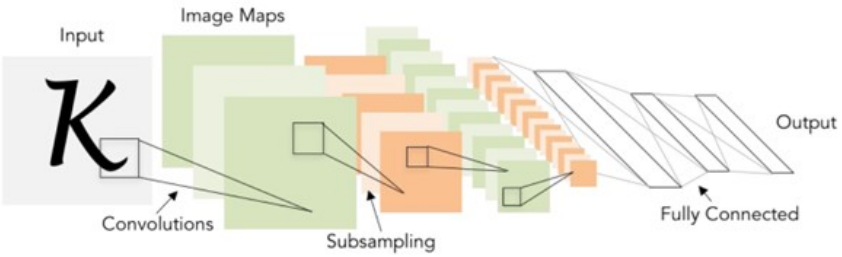


Figure 3.2: General architecture of a Convolutional Neural Network. Extracted from "Convolutional Neural Networks for Visual Recognition" Stanford, Fei-Fei Li & Justin Johnson & Serena Yeung

The classification module outputs the scores based on the high-level features learned by the last convolutional layer. While the first convolutional layers are activated by low-level semantic features, such as edges, changes in shape or contrast, the features learned by the last convolutional layers represent more abstract concepts. It is remarkable to observe that different types of CNN architectures end up learning the same filters for the first convolutional layers (Figure 3.3). These type of features describe general characteristics of shape, color, and texture transition and topology of objects. Moreover, recent work has shown that CNNs possess the incredible ability to localize objects, despite being trained on only image labels [1].

Moreover, in a Convolutional Neural Networks a neuron is connected only to each neuron from a local region of the input volume, called the receptive field, which can be fine-tuned to emphasize on larger or smaller regions. Thus, the similarity to the human visual processing system and performance advantages (such as minimal pre-processing time, automatic feature extraction etc.) have made Convolutional Neural Networks the standard choice for any type of visual classification task.

3.2 LIMITATIONS OF CONVOLUTIONAL NEURAL NETWORKS

Although, Convolutional Neural Networks are remarkably powerful, their architectural design poses certain limitations that may be too restrictive for future tasks. In his 2012 lectures [2], Prof. Geoffrey Hinton identifies the problems that arise from the current architecture of CNNs and highlights the key differences between how the human vision system and CNNs process information, proposing an alternative called Capsule Networks ([16, 17]). He identifies the following problems in current Convolutional Neural Networks:

1 http://cs231n.stanford.edu/2017/ Last accessed on 07.03.2023
2 https://www.youtube.com/watch?v=rTawFwUvnLE Last accessed on 17.06.2023

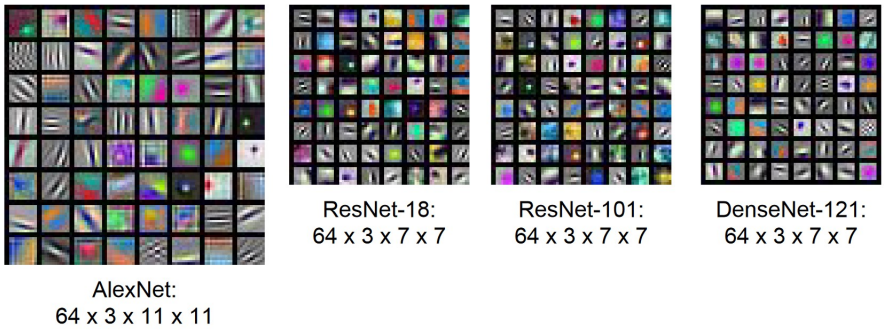
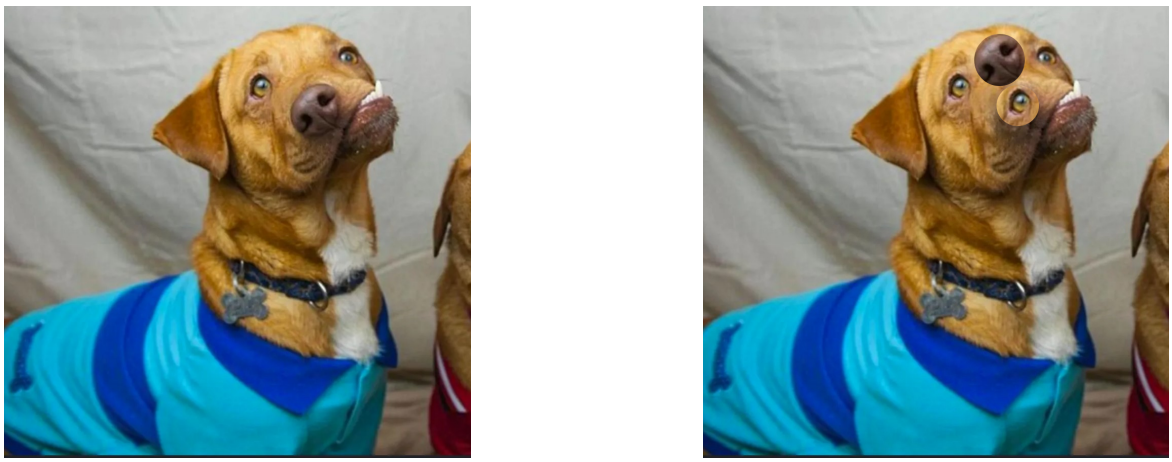


Figure 3.3: First layer learned features for different architectures used for Object Recognition. Extracted from "Convolutional Neural Networks for Visual Recognition" Stanford, Fei-Fei Li & Justin Johnson & Serena Yeung

1. **Pooling looses information**: Under the assumption that the local structure of an image in usually uniform, the pooling layers reduce redundant information, but, at the same time, loose precise spatial localization.

2. **Spatial relationship between part of the image is disregarded**: Because of the flat nature of the layers of neurons, these layers are activated by the presence of the targeted concept in the input image, but deeper in the network, they are unable to recognize the relative relationship between different concepts in the image. This becomes a liability in the context of adversarial attacks (Figure 3.4).



(a) Original image. Target class prediction confidence: 40%.

(b) Modified Image. Target class prediction confidence: 37%..

Figure 3.4: Although there is a clear visual distinction between the original image and the modified image, an AlexNet architecture predicts the correct class for both images with a difference in confidence of only 3%. We expected that the modified image would achieve a significantly smaller score, but because of the lack of relative spatial relationship discovery, both images achieve similar scores.

3.3 GRADIENT-BASED APPROACHES: SALIENCY MAPS

Since gradients play a key-role in the learning process through backpropagation it is only natural that they should offer an insight on the model's decision. Simonyan et al. [18] introduced one of the first approaches for visualizing gradients in a Visual Task based on Convolutional Neural Networks: Saliency maps. First, they proposed the idea of incrementally modifying the input image as to maximize the output score for a certain class, a visualization they call *Class Appearance Model* (Figure 3.5).

Figure 3.5: Class Appearance models for different target classes. Extracted from *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps* [18]

Although this offers some intuition on how the model expects the generic characteristics of each class to look like, it offers no information about which part of the input image the model is basing its decision on. The idea behind saliency maps is to use the gradient of the loss function with respect to the input pixels and map it on the original image. The pixels which have a large absolute value for the gradient at that location are those which, when slightly perturbed, affect the class score significantly. In this way, we create a heatmap which shows the importance attributed to each pixel of the original image in the prediction process (Figure 3.6).

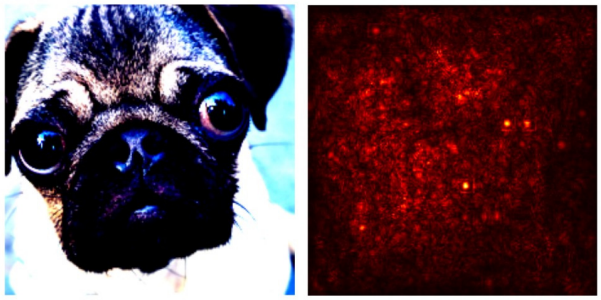


Figure 3.6: Saliency heatmap for the target class "Mops".

More formally, since the output score of a class c for an image I is a non-linear function $S_c(I)$ of the input image, we can use the gradients, which are the derivates of the score with respect to the input image, to approximate the score function in a Taylor expansion [18]:

$$S_c(I) \approx w^T I + b \tag{3.1}$$

$$w = \frac{\partial S_c}{\partial I}\bigg|_{I_0} \tag{3.2}$$

Large positive gradients for a pixel in the original image correspond to parts of the target object which have significant importance in the prediction process, while large negative gradients correspond to the competing class object. Although this can create valuable insight in the case of binary image classification, that is rarely the case. In most scenarios, there is more than one competing class, thus visualizing the pixels for the counterfactual explanation is hard and, often, incomprehensible with saliency maps. Moreover, saliency maps are not class discriminative and have a saturation problem [19]. In order to solve the saturation problem, a new approach called Guided-Backpropagation has been proposed, that restricts the backpropagation of the gradients through the ReLU activation function only to the activations that have positive influence on the output (Figure 3.7).

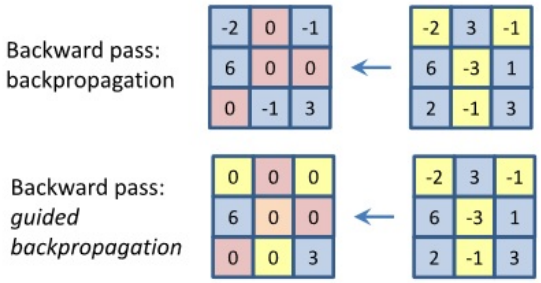


Figure 3.7: Backpropagation pass vs guided backpropagation pass. Extracted from *Striving for Simplicity: The All Convolutional Net* [20]

With this simple improvement, Guided-Backpropagation is able to create more interpretable representations of the decision process. More importantly, Guided-Backpropagation is model-agnostic i.e., works with any type of convolutional architecture.

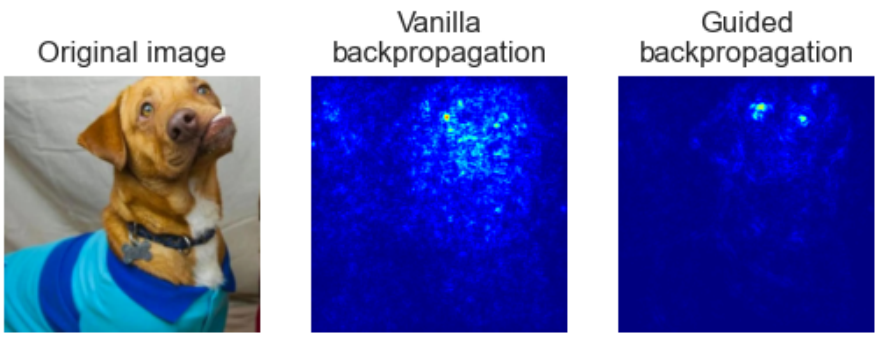


Figure 3.8: Vanilla vs Guided-Backpropagation. Guided-Backpropagation offers a more interpretable heatmap that distinguishes general characteristics of a dog: ear shape, eye position etc.

Method	Formula	Faithfulness
Vanilla Backpropagation (Saliency Map)	$\frac{\partial f}{\partial L_n} = \frac{\partial f}{\partial L_{n+1}} \cdot I(L_n > 0)$	+
Guided-Backpropagation	$\frac{\partial f_L}{\partial L_n} = \frac{\partial f}{\partial L_{n+1}} \cdot I(L_n > 0) \cdot (f^L > 0)$	-

Table 3.1: Different methods for propagating an output activation backwards through a ReLU unit. I is the sign function and f^L is the activation after the L^{th} layer.

In this section, we presented two visualization techniques for Convolutional Neural Networks: Vanilla and Guided-Backpropagation. These are model-agnostic [3] backpropagation based methods, which offer some valuable insight into the model's decision process by propagating the gradients of the output score function with regard to the input image. Although weakly supervised class saliency maps can be used for object localization, despite being trained on image labels only, they are not class discriminative and do not offer a high resolution visualizations. These limitations have driven the research behind Class Activation Maps.

3.4 ACTIVATION-BASED APPROACHES

In this section, we discuss two types of activation based approaches for visualizing the decision process of CNN architectures: Class Activation Maps and Gradient-weighted Class Activation Mapping, respectively. Activation based approaches compute a weighted sum of the feature maps after the target layer, which is then interpolated and overlapped onto the original image and highlights the most informative parts of the image for a given predicted class.

3.4.1 *Class activation maps*

Class Activation Maps aim to solve the shortcomings of saliency maps. Proposed by Zhou et al. [1], Class Activation Maps are class-discriminative saliency maps, which are able to illustrate with high resolution discriminative image regions used by the network to predict that class. The idea behind CAM is to use a weighted sum of the feature maps of the last convolutional layer (which represent activations of high level concepts) in order to compute the activation F_c for class c (Figure 3.9):

$$F_c = w_1^c A_1 + w_2^c A_2 + ... + w_n^c A_n$$

where A_i represents the i^{th} feature map of the last convolutional layer and w_i^c represents the learned weight for that specific feature map for the class c. Zhou et al. [1] propose using a Global Average Pooling (GAP) layer after the last convolutional layer and then learning a linear model from the n resulting weights to the c output classes. The intuition behind using Global Average Pooling is that this encourages the network to learn to identify the

3 Restricted to Convolutional Architectures.

extent of an object, since maximizing the value of the average is finding all discriminative parts of an object.

By projecting back the learned weights of the last fully connected layer on to the last convolutional layer (before the GAP) we create a linear combination of heatmaps of relevant features of the last convolutional layer employed in predicting the target class. The intuition behind using the features of the last convolutional for the visualization is that this layer has the best trade-off between high resolution spatial information and high-level semantic concepts [2]. The spatial information is then lost after the fully-connected layers.

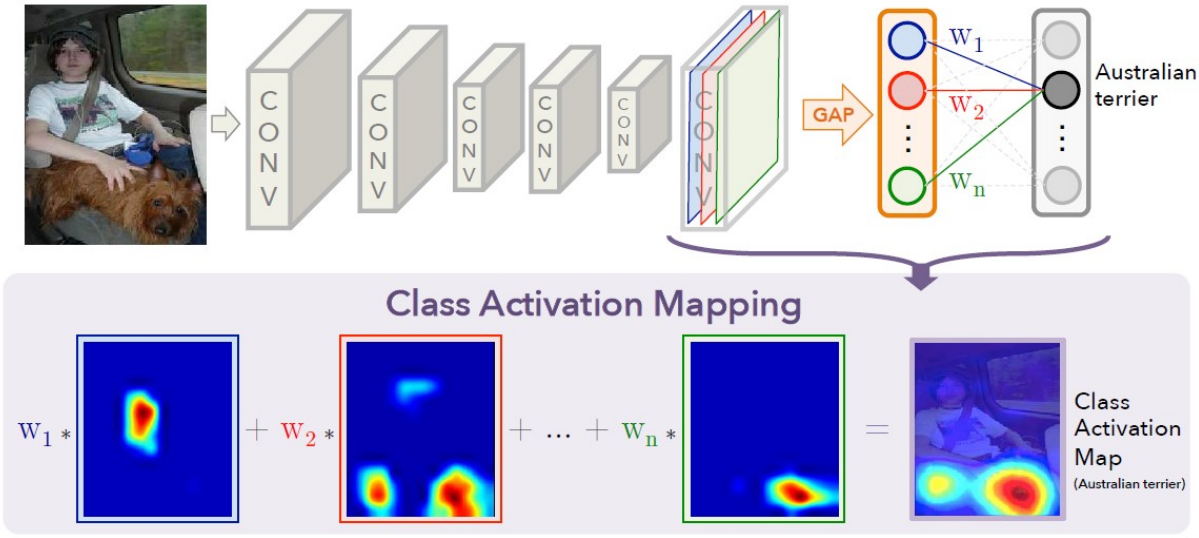


Figure 3.9: The predicted class score is back propagated to the last convolutional layer to generate the activation maps. For a specific class, the activation map represents a weighted sum of visual patterns at different spatial locations. Extracted from 'Learning Deep Features for Discriminative Localization.' [1].

More formally, for a given image, the class activation map for class c is defined by:

$$\begin{aligned} M_c(x,y) &= \sum_{k=1}^{N} w_k^c f_k(x,y) \\ where \quad f_k(x,y) &= \frac{1}{Z}\sum_i \sum_j A_{ij} \end{aligned} \tag{3.3}$$

where N is the number of feature maps in the last convolutional layer and $f_k(x,y)$ represents the activation of the k^{th} feature map. Since the score of the output class after the GAP is:

$$S_c = \sum_k w_k^c \sum_{x,y} f_k(x,y) = \sum_{x,y} M_c(x,y) \tag{3.4}$$

we can observe that CAM explicitly illustrates the importance during the classification process of class c for the activation at the location (x,y). Although CAM requires no backward pass to generate the activation maps, training an additional Fully Connected layer is required. This imposes a constraint on the architecture of the model and turns out to be too

restrictive for tasks such as VQA. Nevertheless, the valuable insight of CAM represents the foundation for a state-of-the-art visualization method called grad-CAM.

3.4.2 *Gradient-weighted Class Activation Mapping*

The limitations of Class Activation Maps lead to the development of Gradient-weighted Class Activation Mapping [2], a more general technique which requires *no architectural changes* (in the case of CNNs) and can be used for reinforcement learning, as well. The idea of using the spatial information of the last convolutional layer, remains the same. The difference lies in calculating the weights of these feature maps. If, in the case of CAM the weights were learned through a linear model, in the case of grad-CAM, the new weights are calculated based on the gradients (thus, a forward and a backward pass are required). More formally:

$$\begin{aligned} L^c_{grad-CAM} &= ReLU(\sum_{k=1}^{N} w^c_k A^k) \\ where \quad w^c_k &= \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A^k_{ij}} \end{aligned} \tag{3.5}$$

The ReLU function acts as a filter that capture only the pixels whose increase in intensity results in an increase of the score of class c, y^c. We can see that Equation 5.2 is a strict generalization of Equation 5.1. This becomes even more clear when we compute the gradient of the output score with respect to each feature map of the last convolutional layer (although any convolutional layer earlier in the network can be used):

$$\frac{\partial M_c}{\partial f_k} = \frac{\frac{\partial M_c}{\partial A^k_{ij}}}{\frac{\partial f_k}{\partial A^k_{ij}}} \tag{3.6}$$

We observe that:

$$\frac{\partial f_k}{\partial A^k_{ij}} = \frac{1}{Z}$$

$$\frac{\partial M_c}{\partial f_k} = w^c_k$$

Substituting this in equation (3.6) we get:

$$\frac{\partial M_c}{\partial f_k} = \frac{\partial M_c}{\partial A^k_{ij}} Z$$

$$w^c_k = \frac{\partial M_c}{\partial A^k_{ij}} Z$$

Summing over each pixel in the activation map and rewriting the last equation, we get:

$$\sum_i \sum_j w_k^c = \sum_i \sum_j \frac{\partial M_c}{\partial A_{ij}^k} Z$$

$$Z w_k^c = Z \sum_i \sum_j \frac{\partial M_c}{\partial A_{ij}^k}$$

$$w_k^c = \sum_i \sum_j \frac{\partial M_c}{\partial A_{ij}^k} \tag{3.7}$$

Thus, we have shown that the weights w_k^c, which were learned through a linear model in the case of CAM, actually evaluate to the gradients of the output scores with respect to the k^{th} feature map. As Selvaraju et al. [2] conclude, grad-CAM represents a strict generalization of CAM. In its current form, grad-CAM is model-agnostic, class-discriminative and successfully localizes areas of interest in the input image for a given class target (and does this without the need for training a linear model; moreover, the gradients are backpropagated just up to the targeted convolutional layer), but it lacks the ability to highlight precise pixel-level details [4]. Thus, these visualizations (vanilla grad-CAM and guided backpropagation) are fused together via element-wise multiplication, which acts as an enhancing mechanism which focuses the heatmap created by grad-CAM onto the fine-grained details of the original image. The resulted visualization is high-resolution, class-discriminative and model-agnostic.

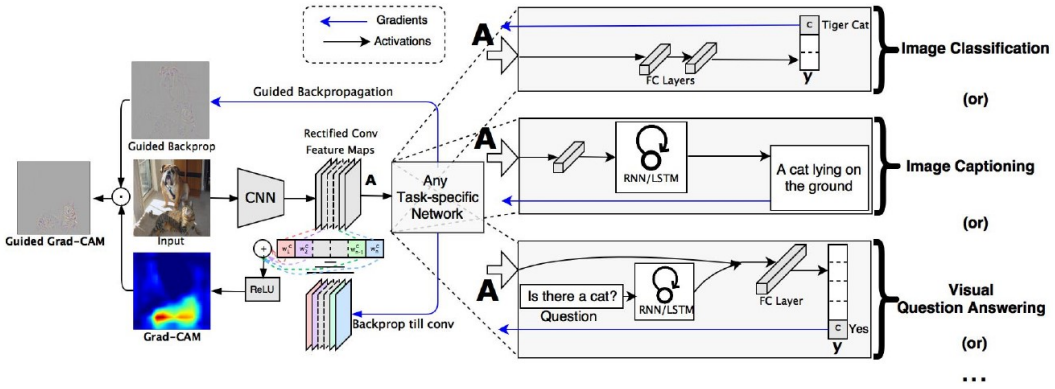


Figure 3.10: Fusion between grad-CAM and Guided Backpropagation visualizations. 'Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization' [2]

Moreover, by simply using the negation of the gradients flowing back into the targeted convolutional layer, we can obtain counterfactual explanations i.e., explanations that illustrate the regions which contribute the most to the inhibition of the neuron corresponding to that particular class and thus to the misclassification of the target class. Consequently, removing such concepts from the original image would increase the confidence of the model in its prediction.

Further improvements have been proposed for grad-CAM such as grad-CAM++, which uses individual weights proportional to each of the object's spatial footprint for each gradi-

4 This is in turn a result of the upsampling of the resulted activation map to the resolution of the input image.

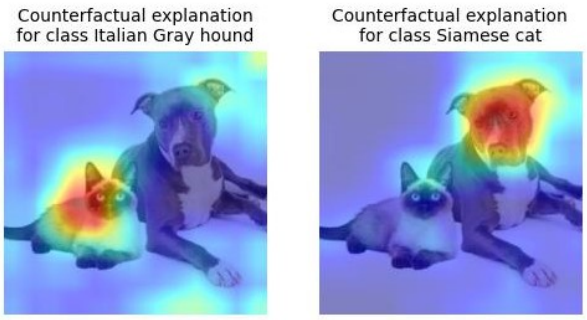


Figure 3.11: Counterfactual explanation generated by grad-CAM in an AlexNet architecture for target class Siamese Cat and Italian Gray hound.

ent flowing into the k^{th} after the last convolutional layer, thus providing an even higher-resolution visualization. We will not explore this method in this paper, since it is not suited for Skeleton Data, where each joint has an equal footprint in the original sequence.

3.5 SUMMARY

In this chapter, we firstly present the general context of Convolutional Neural Networks, their advantages and limitations (Section § 3.1, Section § 3.2). We introduce Saliency Maps (Section § 3.3), as the simplest gradient-based method for pixel-level visualizations that provide insight into the prediction process of the model. We acknowledge the limitations of Saliency Maps and then transition towards state-of-the-art visualization methods. We analyse the intuition behind Class Activation Maps (Subsection § 3.4.1) and Gradient-weighted Class Activation Mapping (Subsection § 3.4.2) and use this insight as a foundation for our proposed solution in the context of Human Action Recognition based on skeleton data.

4

RELATED WORK

In this chapter, we explore visualization techniques in the context of movement pattern identification based on skeletal data. Movement pattern identification is a type of HAR, used for determining types of activities performed by a person (or multiple persons). These may range from everyday activities (such as brushing teeth, eating or drinking water) to specific activities related to a field of work (such as mining, aiming, administering medication etc.). We explore both visualization methods for the input data (**R1**) and methods which are able to provide insight into the decision process of the model (**R2**).

For **R1** we present techniques which transform the input data into an intermediary representation which can illustrate specific temporal and spatial characteristics of the action performed (Section § 4.1, Section § 4.2).

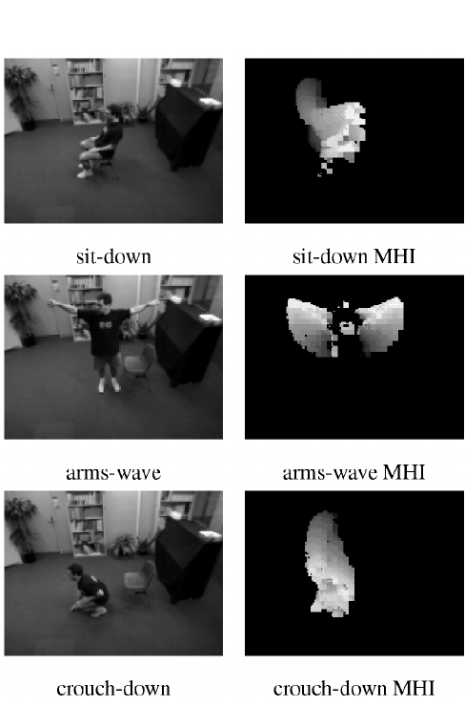
For **R2** we present how an adaptation of Gradient-weighted Class Activation Mapping, traditionally designed to work with image input data, was successfully used by Satoshi and Yukie [7] to create insight into the prediction process in child gross motor skills based on skeletal data. However, Satoshi and Yukie's work create an intermediary representation from the given input skeletal data, which is then fed to a convolutional architecture and do not apply grad-CAM directly on a model which uses skeletal data as input.

We analyze the underlying assumptions, the used fusion mechanisms and their implications on the generated visualizations and discuss on the insight created by such methods. Finally, we note the lack of specific existing work in the area of explainability for HAR based on skeletal joint data.

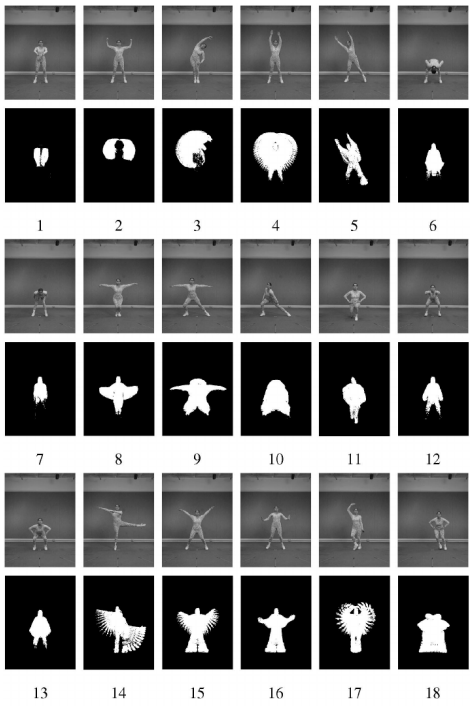
4.1 EARLY WORK: MOTION HISTORY IMAGES

Early works conducted by Davis and Bobick [21] focused on using statistical methods in order to solve these challenges, by creating Motion Energy Images (**MEI**) and Motion History Images (**MHI**) which encode the motion trajectory of the joints over time in a single image which represent action templates. Then, the mean and covariance are used to assign new actions to the most similar template (Figure 4.1). Although, this offers some intuition on the spatial and temporal characteristics of each type of action, it posed many problems such as view sensitivity (*slight changes in the angle of the camera greatly influences the MHI and MEI*) or temporal representation (*the temporal component of repetitive actions is hard to*

represent using MHI or MEI; certain types of action may be similar in the spatial component, but inverse in the temporal component e.g., throwing a ball vs. catching a ball). Moreover, this kind of visualization technique is only suitable for handcrafted features and does not generalize well to Deep Learning Architectures, which extract their own internal representation of the most important features.



(a) Motion History Images



(b) Motion Energy Images for different actions

Figure 4.1: Early visualization methods for Human Action Recognition

4.2 A DIFFERENT DIRECTION: SEQUENCE TO IMAGE (SEQ2IM)

In his recent work, "Deep Learning for Skeleton-Based Human Action Recognition" [6], Laraba proposes a different direction for interpretability in the context of HAR. His approach, called *"Sequence to Image (Seq2Im)"*, transforms the original sequence of skeleton joints into a single *RGB* image which encodes both the spatial and the temporal component. His proposed idea is to encode the spatial coordinates of each joint as a *RGB* color. More formally, each relative coordinate (X, Y, Z) at frame F and belonging to joint J is transformed through normalization into values belonging to a RGB color. Thus, the input tensor shape is transformed into a *RGB* image of height J (25 in the case of *NTU RGB+D 60* or *NTU RGB+D 120* datasets) and width equal to the number of sampled frames in the action. Since the number of frames is usually much greater than the number of individual joints and in order to create a more interpretable visualization, an interpolation is performed which transforms each pixel in a $128x128$ square.

This transformation is able to illustrate sudden change in motion or repetitive behaviour through the change in gradient. Moreover, as the input data is now represented as an *RGB* image, traditional CNN architectures (e.g., *Inception V3, VGG19, ResNet152, DenseNet121*) and traditional explainability methods (e.g., Class Activation Maps, Gradient-weighted Class Activation Mapping, Guided-Backpropagation etc.) can be used without any further adjustments.

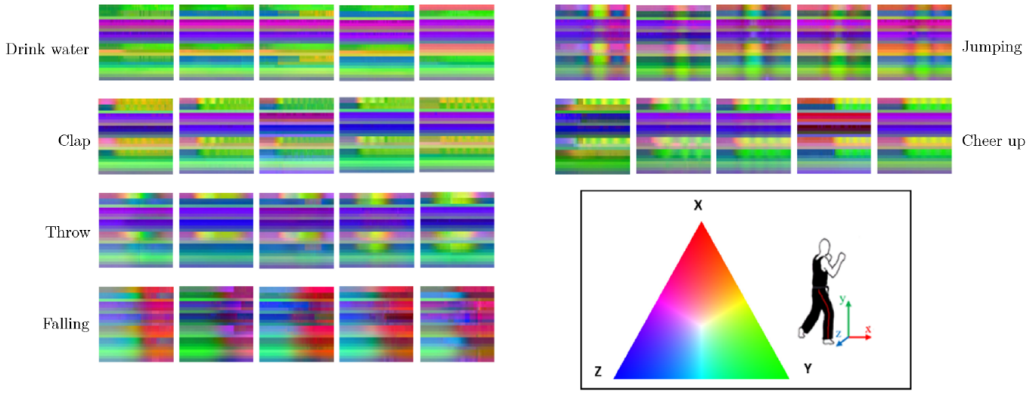


Figure 4.2: The transformed input data is able to capture repetitive temporal characteristics across joints and spatial motion across frames (e.g., in the "Jumping" action, vertical bars illustrate the repetitive motion; in the "Falling" action, the sudden gradient shift of all joints toward red is able to capture both the exact moment the action begins and the change in spatial coordinates). Extracted from 'Deep Learning for Skeleton-Based Human Action Recognition' [6].

The generated heatmaps illustrating the most important joints can be then mapped onto the original skeletal data using a similar inverse transformation (Figure 4.3). Nevertheless, this approach fails to achieve SOTA results, but serves as an interesting alternative to interpreting the characteristics of the input data and the predictions in the case of Human Action Recognition.

4.3 DERIVING PSEUDO-IMAGE REPRESENTATIONS FROM SKELETON DATA

Satoshi and Yukie [7] successfully adapt grad-CAM to work with skeleton input data, in this way creating insight into the decision process of their model. However, their adaptation consists in creating an intermediary representation from the skeletal data in the form of a grayscale image, which is then fed to their network. In this way, they are able to apply the original grad-CAM procedure, that is backpropagating and averaging the gradients from the output to the last convolutional layer and creating a weighted sum of the feature maps (which, in this case, are tensors of only two dimensions $[H, W]$).

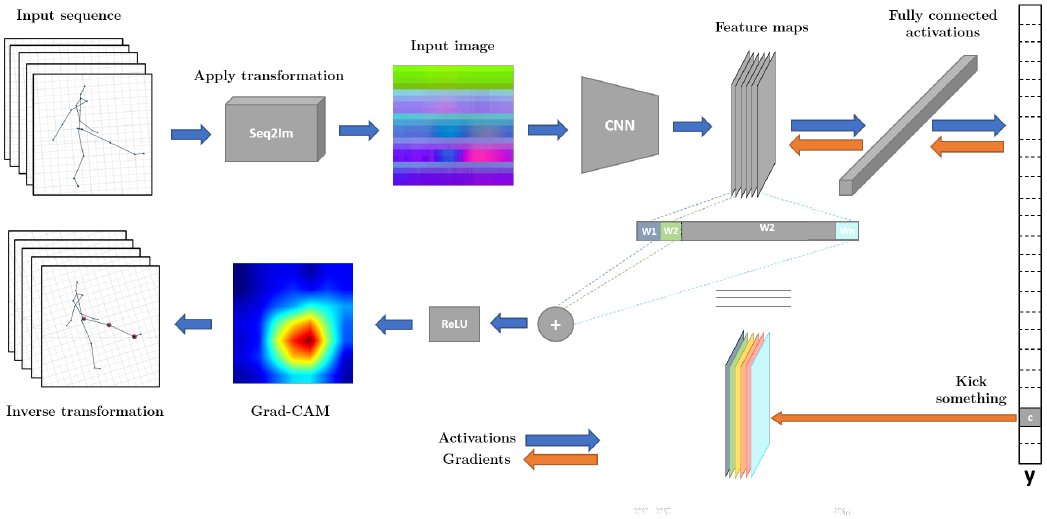


Figure 4.3: Seq2Im architecture overview. Skeleton data in transformed into a RGB image that is fed to a traditional convolutional architecture. In the backward process, the grad-CAM heatmap, representing a *RGB* image is transformed into skeletal data. Extracted from 'Deep Learning for Skeleton-Based Human Action Recognition' [6].

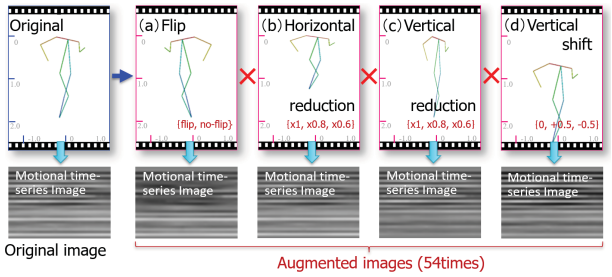


Figure 4.4: Input data transformation. Skeletal data is transformed into a combined grayscale image, which is then fed as the new input for a VGG. Extracted from 'Skeleton-based explainable human activity recognition for child gross-motor assessment' [7].

The generated visualization represents a pseudo grayscale image which captures the most important joints (spatial component) and the moments of their activation (temporal component). Although it is not possible to visualize the skeletal motion directly from this resulted pseudo-image, it is possible to reproduce the skeletal motion from a transformation which uses both the resulted grad-CAM heatmap and the original transformed skeletal data. This is possible because both the original input data transformation and grad-CAM preserve the spatial characteristics of the input data i.e., image patches of the new input image and of the generated grad-CAM heatmap correspond to the same location. Figure 4.5 illustrates the inverse transformation, which takes the pseudo-image generated by grad-CAM and the intermediary skeleton representation and reproduces skeleton data highlighting the most informative bones in each frame.

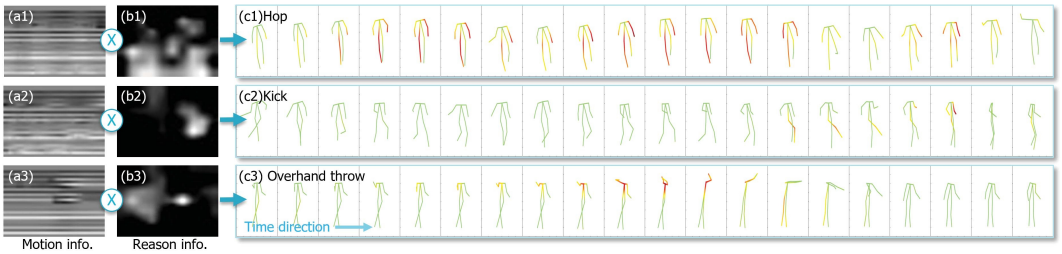


Figure 4.5: Intermediary representation of the skeletal data (a). Grad-CAM generated heatmaps (b). Reproduced skeleton motion visualization (c). Extracted from 'Skeleton-based explainable human activity recognition for child gross-motor assessment' [7].

4.4 SUMMARY

In this chapter, we discussed existing methods for interpreting skeleton data and the decision process of different HAR architectures. We began by exploring traditional methods for visualizing skeleton data: Motion Energy Images (MEI) and Motion History Images (MHI) respectively. These visualizations create insight into the trajectory of the joints over time of each action class. We then present two distinct approaches (Section § 4.2, Section § 4.3) which successfully use grad-CAM to illustrate that the model is concentrating on the most informative joints. Nevertheless, both approaches rely on creating an intermediary image-like representation of the skeleton data. In the Chapter 5, we will discuss how grad-CAM can be adapted to work directly on skeleton data.

5

PROPOSED SOLUTION

In this chapter, we describe how Class Activation Maps and Gradient-weighted Class Activation Mapping can be adapted to work directly on skeleton data. In Section § 5.1, we begin by demonstrating the application of CAM on HAR models (*based on skeletal joints*) that utilize a GAP layer in their classification module. In Section § 5.2, we outline the necessary modifications that must be made to grad-CAM to enable its functionality with skeleton data. We conclude this chapter by presenting the key distinctions between image and skeleton input data (Section § 5.3).

5.1 CAM ON SKELETON DATA

Since most of the available configurations of the ResGCNv2.0 use a Global Average Pooling layer in their classification module, we are able to compute the Class Activation Maps without any further architectural change. CAM uses the weights of the last Fully Connected (FC) Layer from the classification module as follows (Figure 5.1):

$$L^c_{CAM} = \sum_f w^c_{kf} A^k_f \tag{5.1}$$

where f is the common dimension on which the tensor contraction is performed. Since w will be a tensor of size $[C, f]$, where C is the number of output classes (60 in the case of the *NTU RGB+D* dataset, and 120 in the case of *NTU RGB+D* 120 dataset) the generated heatmaps will be a tensor of size

$$[C, F', J, P]$$

C : Number of output classes (*60 or 120*)

F' : Number of output frames

J : Number of joints for describing the skeleton data (*25 in the case of NTU RGB+D*)

P : Number of persons or individuals in a multi-person setting

Because of the Temporal Convolutional (TC) operation, F' is always smaller than the original number of frames. Since we are only interested in those weights our the target class c, we index the generated heatmap tensor after the first dimension.

The advantage of CAM is that it does not require an additional backward pass through the network. Nevertheless, if the model does not have a Global Average Pooling layer in its architecture, then a massive overhead is created by inserting a GAP layer and learning the new weights and may not even be possible in some cases. That is why, in the next section, we discuss how we can adapt grad-CAM, a more general technique, to work on skeletal data.

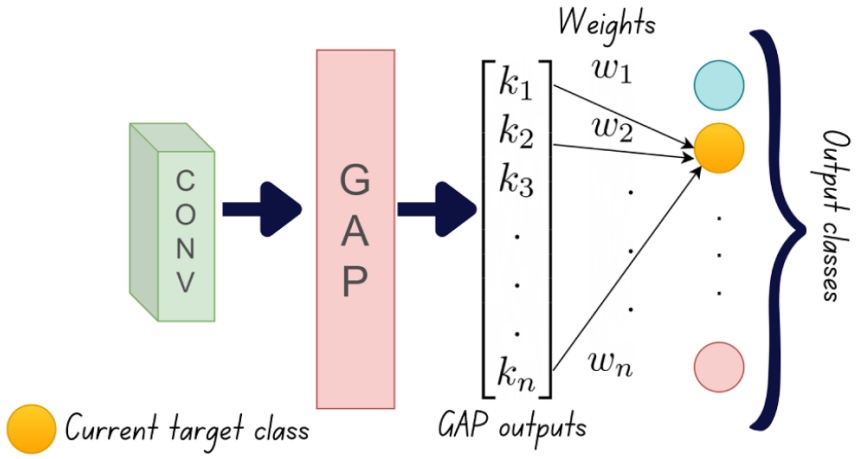


Figure 5.1: Weight learning process in the CAM method. In CAM the learned weights of a linear model are used in Equation 5.1. Image by Bala Priya C.

5.2 ADAPTING GRAD-CAM FOR SKELETON DATA

In Section § 3.4.2 we discuss how we derive grad-CAM from CAM and how it represents a strict generalization of CAM. We keep the underlying insight of the original paper [2] when designing our adapted version, but we account for working with skeleton data (which has four dimensions) instead of image data (which has only two dimensions). More formally, the original formulation of grad-CAM is:

$$L^c_{grad-CAM} = ReLU(\sum_{k=1}^{N} w^c_k A^k) \tag{5.2}$$

In the case of skeletal data, the activation maps after the last convolutional layer (before the classification module) are tensors of shape $[C, F, J, P]$ where C is the number of channels, F is the number of output frames, J is the number of individual joints in the skeletal data (25 for the *NTU RGB+D* dataset) and P is the number of persons performing the action (can be 1 or 2). The shape of the gradients extracted depends on the specific convolutional layer being targeted. Although their shape may vary, they **consistently have the channel dimension C in common** with the activation maps. Thus, we compute an average on all the other dimensions, resulting in a vector of C real numbers which will be used as the weights for each feature in the feature maps. Since we are only interested in the influence of the desired class (and not the competing classes), we apply the ReLU activation function

over the generated heatmaps. The result encodes the importances of each joint of both skeletons at each frame.

5.3 IMAGE VS SKELETON INPUT DATA

Table 5.1 illustrates the different tensor shapes in the case of grad-CAM applied on a traditional convolutional architecture with images as input and in the case of HAR with an ResGCNv2.0 architecture and skeleton data as input. The resulting heatmap is always smaller than the original input, because of the irreversible information loss in the forward pass (e.g., from Pooling). Thus, in the case of grad-CAM for image inputs, an interpolation is performed which transforms the output heatmap to be the same size as the original image, in this way allowing it to be overlapped and visualized. In the case of HAR the number of frames in the generated heatmap is significantly smaller than the original number of frames (e.g., 72 vs 288 for our test configuration in Chapter 7) and this is because of the Temporal Convolutional Layers which reduces the temporal dimension. We chose not to interpolate the resulting heatmap to the original number of frames in order to avoid over-smoothing.

	Image input data	**Skeleton input data**
Feature maps shape	$[C, H, W]$	$[C, F, J, P]$
Gradients shape (before average)	$[C_1, C_2, H, W]$	$[C_1 \dots C \dots C_n]$
Grad-CAM heatmap shape	$[H', W']$	$[F', J, P]$

Table 5.1: Comparison between tensor shapes in the case of grad-CAM used for image and skeleton input data. The shape of the gradients is strictly dependent on the convolutional layer used as target.

5.4 SUMMARY

In this chapter, we presented how Gradient-weighted Class Activation Mapping can be adapted to work directly on a Graph Convolutional Network architecture, namely ResGCNv2.0 [8]. Nevertheless, the proposed method can be easily adapted to work on any kind of convolutional architecture based on skeletal data. We began by illustrating how CAM can be used in this situation because of the presence of the GAP layer in the given architecture (Section § 5.1). We acknowledged the limitations of CAM and transitioned towards grad-CAM, a strict generalization of CAM (Section § 5.2). We showed how grad-CAM can be adapted to work on skeleton-data and, finally, we discussed the main differences between using grad-CAM on image and skeleton data (Section § 5.3).

6

IMPLEMENTATION DETAILS

In this chapter, we present implementation details of our proposed solution, technologies used, and an adapted version of Gradient-weighted Class Activation Mapping suited for Skeleton-Based Human Action Recognition. We employ a top-down approach when designing our extraction process, which allows us to use our technique with little modification on different configurations of ResGCNv2.0. In Section § 6.1 we describe our base framework, made available by Yi-Fan Song[1], which we use as a starting point for our explainability methods. In Section § 6.2 we describe the primary architecture used when designing our algorithms. In Section § 6.3 we describe the main algorithms of our implementation and their purpose.

6.1 FRAMEWORK SETUP

We use the framework made available by Yi-Fan Song [2]1 as a basis for our implementation. The framework, written in pyTorch, contains the pretrained model configurations, scripts for generating data, training and evaluating and their own visualization technique, which is based on Class Activation Maps. The framework allows for easy switching between the two datasets (*NTU RGB+D 60* and *NTU RGB+D 120*) between the pretrained available configurations (*EfficientGCN-B0, EfficientGCN-B2, EfficientGCN-B4*) and between benchmarks (*Cross Subject* or *Cross View*).

6.2 MODEL ARCHITECTURE

EfficientGCN is a Graph Convolutional Network architecture proposed by Yi-Fan Song et al. [8] which aims to address the complexity of the recent Human Action Recognition SOTA models, which require numerous parameters. In order to significantly reduce the number of parameters of the model, the three input branches of the preprocessing module (Joint, Velocity, and Bone) are fused together at an early stage, instead of using a multi-stream network. In this way, *EfficientGCN* achieves SOTA performance (92.1 % accuracy on the *X-Sub (cross subject) NTU 60* dataset) with a number of parameters of only 0.29M (5.82× smaller and 5.85x faster than the SOTA *MS-G3D*) [8]. The authors conclude that *EfficientGCN* is able to obtain SOTA performance with a significantly lower number of parameters because overly-complex HAR models have many redundant parameters.

1 https://gitee.com/yfsong0709/EfficientGCNv1 Last accessed on 05.06.2023

The basic components of the *EfficientGCN* model are the Graph Convolutional Network (GCN) blocks, which are composed by stacking Spatial Graph Convolutional (SGC) layers, Temporal Convolutional layers, and Attention Modules. Residual connections are introduced in order to facilitate the optimization process.

6.2.1 *Attention Module*

Yi-Fan Song et al. [8] propose a new attention module which aims to address the high correlation between the spatial and the temporal dimension in Human Action Recognition tasks. The module, named Spatial Temporal Joint Attention is designed to distinguish the most informative joints in certain frames. The module works by averaging the input features, both at frame-level and at joint-level. Then, the resulted feature vectors are fed through a series of FC layers which output two sets of attention scores, for the temporal (frames) and spatial component (joints) (Figure 6.1).

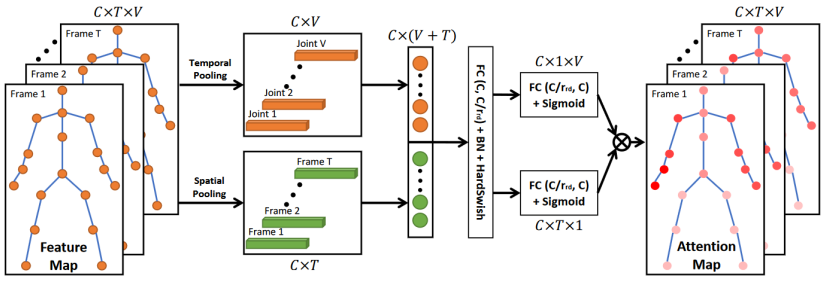


Figure 6.1: Overview of the ST-JointAtt module. Extracted from 'Constructing Stronger and Faster Baselines for Skeleton-Based Action Recognition' [8]

6.2.2 *Temporal Convolution Layer*

Authors [8] propose 4 different implementation for the Temporal Convolutional layer. These layers have been widely used in CNN literature for visual recognition tasks.

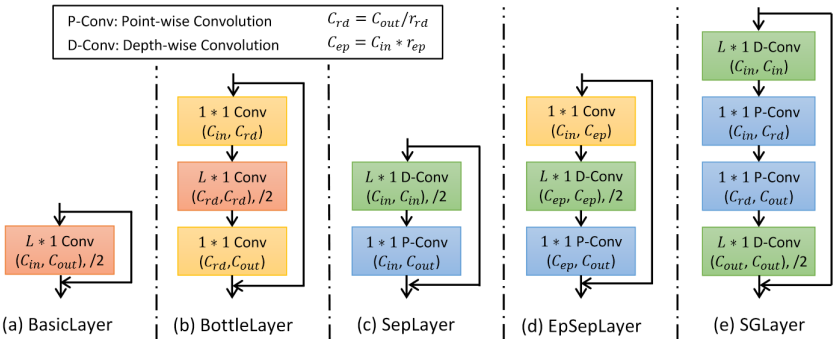


Figure 6.2: Types of Temporal Convolutional layers used in different configurations of the *EfficientGCN* architecture. r_{rd} and r_{ep} represent reduction or expansion factors for the inner channels. Figure extracted from 'Constructing Stronger and Faster Baselines for Skeleton-Based Action Recognition' [8].

6.3 IMPLEMENTED ALGORITHMS

In this section, we present the most important parts of our visualization process. We describe our implemented algorithms, the data flow and additional modifications which allow our solution to work on different type of models.

6.3.1 *Extracting the gradients and activation maps*

As described in Chapter Chapter 5, in order to compute the Gradient-weighted Class Activation Mapping, we first need to select the target layer in our model, get the subsequent activation maps and backpropagate the gradients of the output class with regard to those activation maps. In order to extract the activation maps, we used pyTorch's mechanism of hooks[3]. Hooks are a mechanism of executing predefined functions triggered by a forward or a backward pass of a tensor or *nn.Module* object. Hooks allow for easy extraction of the necessary activation maps after the target layer. Instead of extracting the gradients of the output probability function w.r.t the selected target layer, we create a one-hot encoded tensor with the same shape as the output tensor which we backpropagate through the network.

$$\textbf{one\_hot} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \quad \textbf{one\_hot}[i] = \begin{cases} 0, & \text{if argmax(output)} \neq i \\ 1, & \text{otherwise} \end{cases}$$

We backpropagate the one-hot encoded tensor instead of the original output, since we want to only account for the influence of the predicted class. In the case of our chosen configuration, we select the activation maps after just before the classification module, since they offer the best trade-off between feature discovery and spatio-temporal localization. We then explore three different target layers for computing our gradients: the last two convolutional layers of the Spatial Temporal Joint Attention module and the last convolutional layer of the *Sandglass Layer* (*SGLayer*) [22]. We observe that using the last convolutional layer of the *Sandglass Layer* preserves the spatial localization, but entirely looses the temporal localization, i.e., the correct joints are activated but in the wrong frames. We further elaborate on this in 7.

After the extraction process has been completed, we obtain the activation maps of shape $[C, F, J, P]$ where C is the number of channels, F is the number of output frames, J is the number of individual joints in the skeletal data (25 for the *NTU RGB+D* dataset) and P is

3 https://pytorch.org/tutorials/beginner/former_torchies/nnft_tutorial.html Last accessed on 05.06.2023

the number of persons performing the action (can be 1 or 2). In order to compute a tensor contraction (a weighted sum on the common dimension) we first average all the other dimensions in the gradient tensor, obtaining a vector of C real values (*in our experiments on the aforementioned selected configuration, we obtained activation maps of size* $[128, 72, 25, 2]$ *and gradients of size* $[128, 64]$). Lastly, we obtain the grad-CAM maps of size $[F, J, P]$, on which we apply the ReLU activation function, since we only want to extract the influence of the target class. There is an irreversible loss of temporal information during the forward process, thus the number of resulted frames F is less than the number of frames in the original action.

We note that our proposed approach is both model-independent (we only need to select the target layer) and efficient (only a forward pass and a partial backward pass are required).

6.3.2 *Extracting the joint locations*

In order to visualize the generated heatmaps, we overlap them on a 2D representation of the input data. The skeleton data is a tensor of size $[F, J, P, B, C, A]$, where:

1. **F** is the number of frames (*288 in the case of NTU RGB+D 60 dataset*)
2. **J** is the number of joints (*25 joints in the case of NTU RGB+D 60 dataset*)
3. **P** is the number of persons performing the action (can be 1 or 2)
4. **B** is the batch size
5. **C** is the number of channels of the XYZ joint coordinate
6. **A** is the number of relative angles

We are only interested in the spatial and temporal dimensions of the input tensor, which we then project onto a 2D space to be plotted. We create a visualization function which takes as input the heatmap (generated either by CAM or grad-CAM) and overlaps it onto the 2D joint locations (*after a projective transformation*), assigning each joint (or bone) a color range which represents the importance of that joint, at that frame, in the decision process of the model.

6.4 SUMMARY

In this chapter, we presented aspects related to the implementation of our proposed solution. We started by discussing the target model architecture for our implementation and its main components. We continue to briefly explain the extraction process of the gradients and activation maps and how we overlap the heatmaps generated by our technique onto the original skeleton data.

7

EXPERIMENTS AND EVALUATION

In this chapter, we present the experimental results of our work. We test our adapted version of grad-CAM on two different configurations of the ResGCNv2.0 [8]. The first architecture relies on Sandglass Layer (*SGLayer*) [22] as the main block of the Temporal Convolutional modules, while the second relies on the Expanded Separable Layer (*EpSepLayer*) [23]. We use different target layers for extracting the gradients and explore the influence of inner layers in terms of the trade-off between spatial and temporal localization. Through our visualizations, we are able to evaluate the importance of the Spatial Temporal Joint Attention module. We analyse the classes with the lowest confidence scores and the misclassifications, and try to offer an explanation through our method. We use actions performed by one and two persons from the NTU RGB+D 60 dataset. Finally, we present results from our agreement analysis.

7.1 DATASET

We run our experiments on the NTU RGB+D 60 dataset [24]. NTU RGB+D 60 dataset contains 56880 human action videos collected using a *Kinect v2* camera. There are 60 individual actions (50 individual actions and 10 actions performed by two subjects). Each skeleton is encoded using a 25 joints configuration (Figure 7.1). There are two available benchmarks: **Cross-subject**, which splits the dataset by the subjects performing the actions and **Cross-view** which splits the dataset by the views of different cameras used in the collection acquisition process.

7.2 MODEL CONFIGURATION

When designing our implementation, we used two EfficientGCN-B0 configurations with *Sandglass Layer* and *Expanded Separable Layer* as the main building blocks, using the Cross Subject benchmark of the *NTU RGB+D 60* dataset. Nevertheless, our approach requires little to no modification to accommodate for different model architectures or datasets.

7.3 TARGET LAYER INFLUENCE

In this section, we explore the influence of the selected target layer for computing the gradients used in our adapted grad-CAM method. Traditionally, the target layer from which

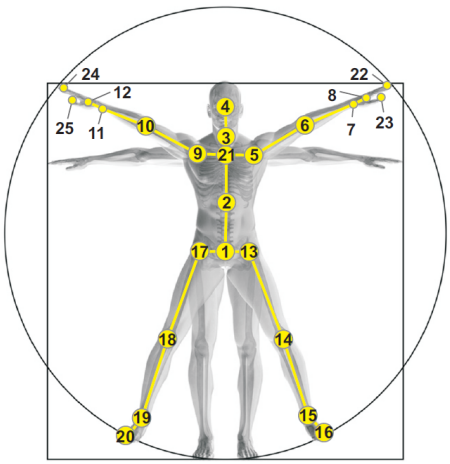


Figure 7.1: Configuration of the joints in the NTU RGB+D 60 dataset. Figure extracted from 'NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis' [24]

the gradients are extracted is the same as the layer used for the activation maps i.e., the last convolutional layer before the classification module, since that layer offers the best trade-off between concept discovery and spatial (and, in this case, temporal) localization. Through our experiments, we explore the influence of selecting different layers for computing our gradients, while using the last convolutional layers to extract the feature maps.

In the case of our first chosen configuration, we explore three different layers: the last two convolutional layers before the Classification Module ($conv_v$ and $conv_t$ respectively) and the last convolutional layer before the Spatial Temporal Joint Attention block proposed by Yi-Fan Song et al. [8]. A representation of the target blocks can be found in Chapter 9. Figure 7.2 illustrates the activations for each frame of the action *"Salute"*. Row a) and row b), respectively, corresponding to the target layer $conv_v$ and $conv_t$, show little difference in the spatial or temporal characteristic of the visualization (there is a slight difference in the intensity of the activation). It is interesting to observe that, in the case of row c), corresponding to the gradients of the last convolutional layer in the *Sandglass Block*, the spatial component is correctly activated (right hand), but in the wrong frame. Nevertheless, using the gradients from *c)* introduce artefacts in the visualization and thus, does not represent a good choice as a target layer.

The same behaviour is consistent with the visualizations of different actions using the gradients from the last convolutional layer in the *Sandglass Block*. Moreover, we observe through our experiments that several action classes present no activation when using the aforementioned gradients (Figure 7.3, row *c*), thus we conclude that the convolutional layers of the Spatial Temporal Joint Attention block create the most interpretable visualizations. This is contrary to what we observed in simpler CNN architectures (e.g. Alexnet) used for Image Classification, where we prefer intermediate convolutional layers as target for our grad-CAM visualization, rather than the very last layers, which creates a fine-grained heatmap harder to interpret. It is speculated that a model with many intermediate layers

overly subdivides the semantic information to a degree difficult to interpret for humans (see § 2.3).

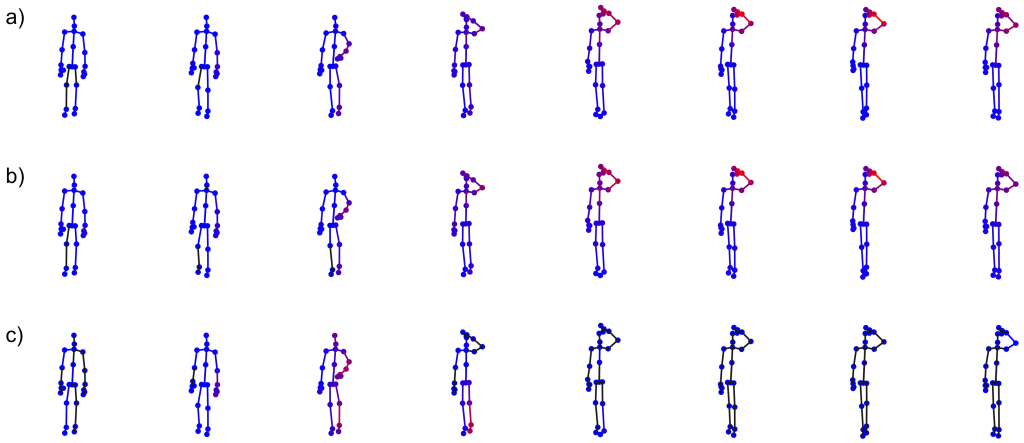


Figure 7.2: Predicted class: salute. a) using gradients from the $conv_v$ layer b) using gradients from the $conv_t$ c) using gradients from the *residual* layer

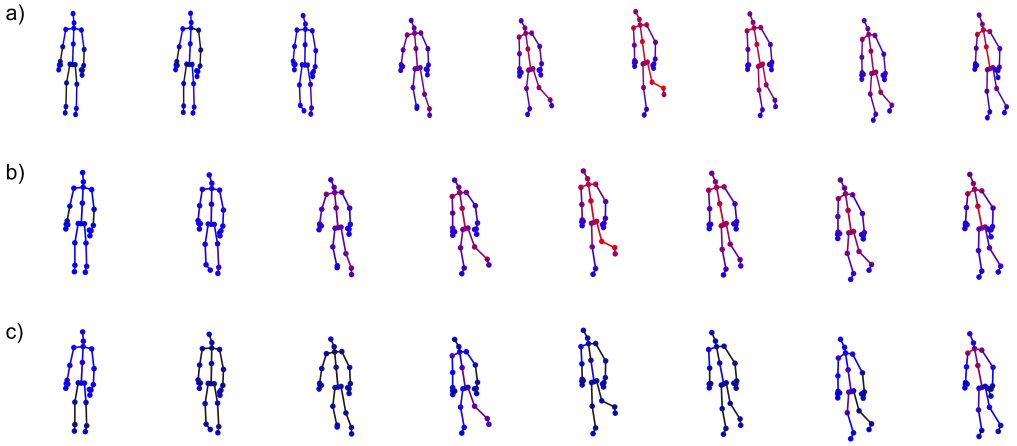


Figure 7.3: Predicted class: Hopping. a) using gradients from the $conv_v$ layer b) using gradients from the $conv_t$ c) using gradients from the *residual* layer

We conclude that our generated visualizations illustrate similar behaviour to the visualizations generated by Yi-Fan Song et al. [8] in their paper 'Constructing Stronger and Faster Baselines for Skeleton-Based Action Recognition' (Figure 7.4).

7.4 EXPLAINING FAILURE MODES

In this section, we present how our visualizations offer intuitive explanations for seemingly inexplicable failure cases. Figure 7.5 illustrates the activations for a misclassified. Although, our model predicts the action class *using a fan* (instead of *clapping*), we observe that the model correctly attributes high importance to the joints in the hands (spatial component) and the repetitive action of the hands (temporal component). Our agreement analysis

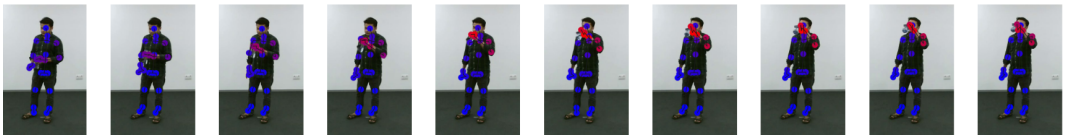


Figure 7.4: Predicted class: Drinking water. Visualization generated with CAM. Extracted from 'Constructing Stronger and Faster Baselines for Skeleton-Based Action Recognition'

(Section 7.6) suggests that humans also have difficulties identifying similar actions based solely on the skeleton data.

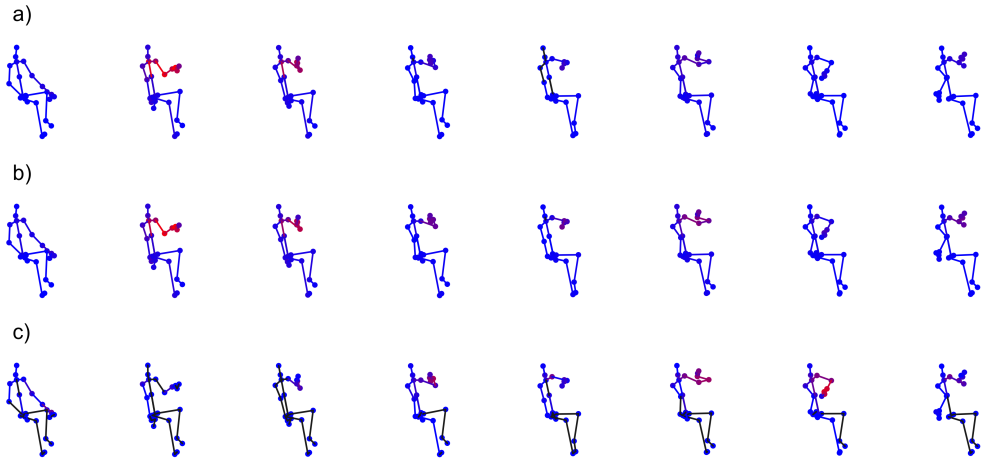


Figure 7.5: Misclassified action. Predicted action is using a fan, while the real class is clapping. a) using gradients from the $conv_v$ layer b) using gradients from the $conv_t$ c) using gradients from the *residual* layer

Another experiment we conducted shows two important aspects regarding the misclassified actions: (a) the false skeletons generated by the data acquisition module are not activated in our generated heatmaps, i.e. the model does not attribute any importance to those joints and (b) the spatial localization of the false skeletons can contribute to the misclassification of an action. Figure 7.6 shows how the correct joints are activated (right hand), but, because of the presence of the false skeleton, the model misclassifies the action as being *touch pocket*. This behaviour may exist because of the close spatial relationship between the moving hand of the first skeleton and the false skeleton. Similar experiments can be reviewed in Chapter 9.

7.5 COMPARATIVE ANALYSIS: CAM VS GRAD-CAM

In this section, we examine the heatmaps generated by CAM and grad-CAM for several action classes and examine the trade-offs between the two. Since grad-CAM combines the gradients with the activation maps extracted after the desired layer and, since gradients represent continuous change, we expect the generated heatmap to be smoother than in the case of CAM. We inspect this behaviour through our comparative analysis.

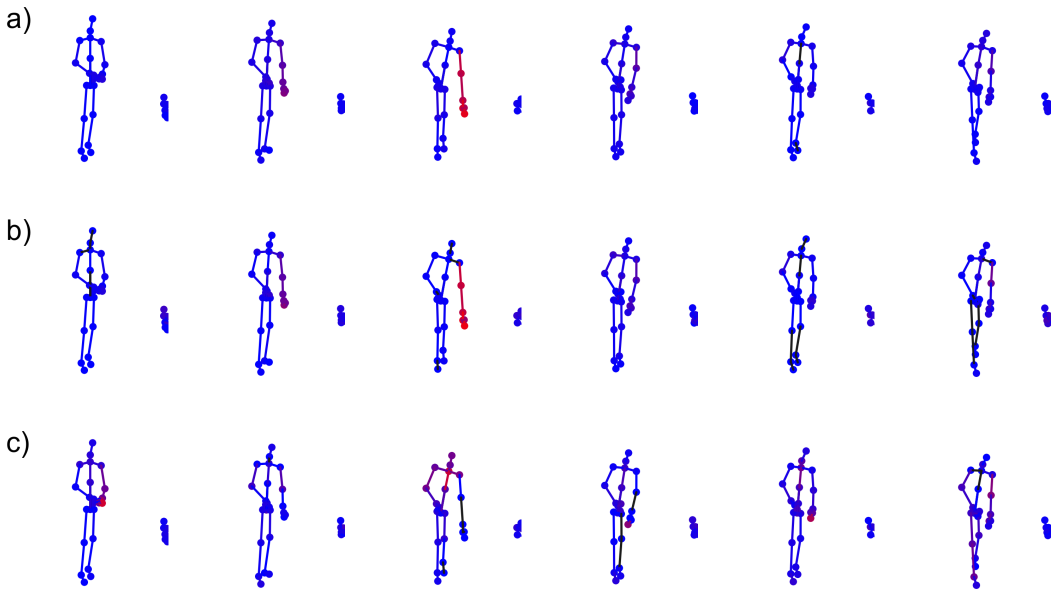


Figure 7.6: Misclassified action. Predicted action is touch pocket, while the real class is dropping something. a) using gradients from the $conv_v$ layer b) using gradients from the $conv_t$ c) using gradients from the *residual* layer

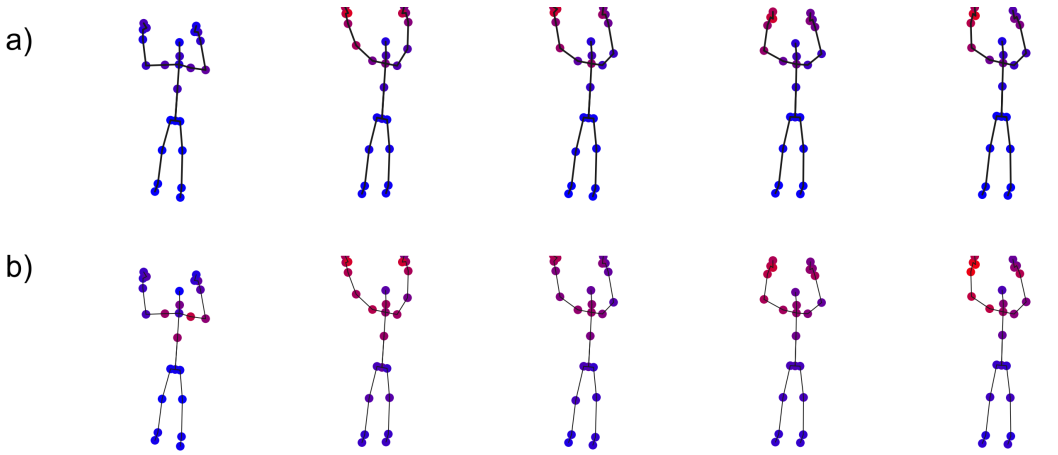


Figure 7.7: Performed action: Cheering up. a) grad-CAM b) CAM

Figure 7.7 illustrates the overlapped heatmaps for grad-CAM and CAM, respectively. We observe that grad-CAM creates more faithful visualizations, which have denser temporal localization and more accurate spatial localization than CAM. In our extended experiments (Chapter 9), grad-CAM did not generate overly-smoothed heatmaps and created more interpretable visualizations than CAM.

7.6 AGREEMENT ANALYSIS

We conduct an agreement analysis with a target group of 33 non-STEM students in order to assess differences between machines and humans in interpreting actions based on skeleton data and to evaluate trust levels in the model. Our study illustrates that without further

contextual information, identifying certain actions based solely on skeleton joints is a hard problem even for human subjects.

From our analysis, we observed that the action class "Eating snack" of the *NTU RGB+D 60* dataset has the most divided answers (Figure 7.8). Inspecting the original visualization we conclude that this is because of the high degree of similarity between all actions involving moving the hands towards the head, since, without any contextual information (e.g., regarding the environment or the object held) nor the model nor the human subjects cannot predict with high certainty the action being performed. This was confirmed by the confusion matrix provided by our pretrained model [8], which suggests that the most similar classes to *"Eating a snack"* are *"Putting on glasses"* and *"Drinking water"* (Figure 9.7). Nevertheless, several action classes were easily identified both by our model and by human subjects (Figure 9.8, Figure 9.9). More than 50% of the subjects confirmed that the provided visualization (grad-CAM) was helpful in creating a stronger sense of trust in the model's prediction, and agreed that the joints activated by grad-CAM correspond to human intuition.

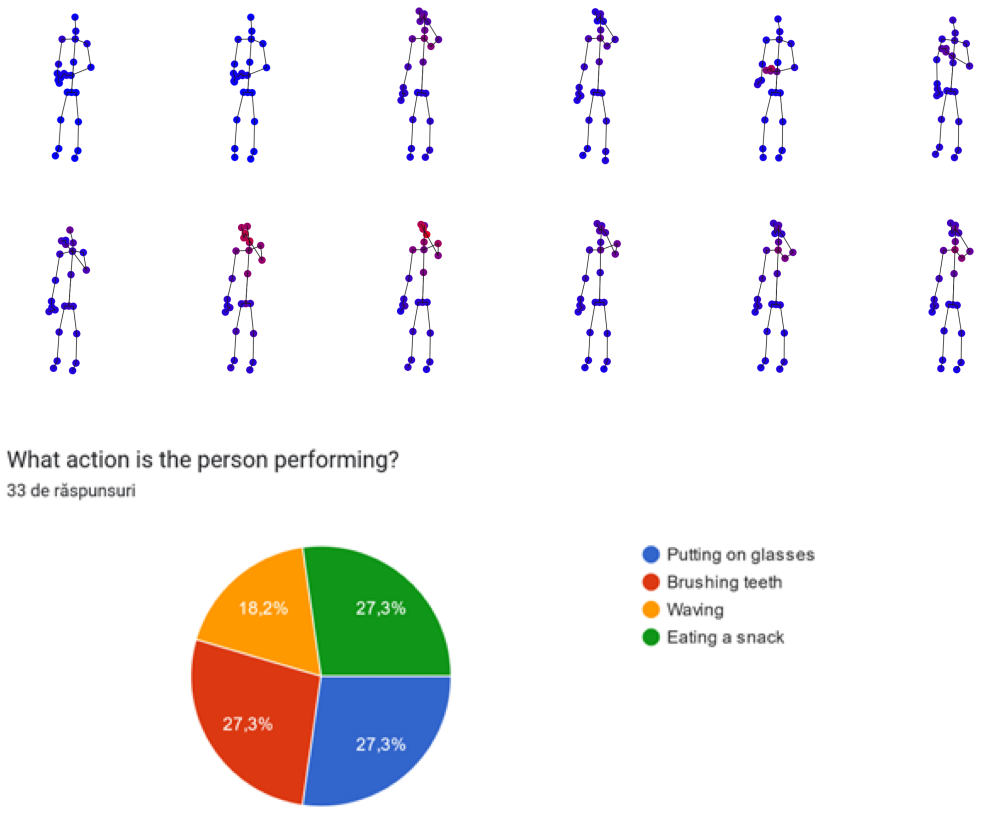


Figure 7.8: Performed action: Eat snack

8

CONCLUSIONS

In this paper, we analysed existing explainability methods for Deep Learning Architectures, traditionally designed to work on image input (Chapter 4) and successfully adapted Gradient-weighted Class Activation Mapping to work directly on skeleton data (Chapter 5). We test our proposed approach on different configurations of the ResGCNv2.0 [8] and explore how the target layer influences the generated visualizations.

8.1 CONTRIBUTIONS

We note that we have successfully achieved the following objectives:

1. We explored the motivation behind Explainable Artificial Intelligence, the use cases and ethical problems that may arise from using models which cannot provide an explanation for their decision process.
2. We reviewed the advantages and limitations of Convolutional Neural Networks. We analysed different Computer Vision XAI methods that were traditionally designed for image input data, such as Saliency Maps, Class Activation Maps and Gradient-weighted Class Activation Mapping and investigated their limitations.
3. Through visualizations provided by grad-CAM we were able to illustrate the incredible localization capabilities of Convolutional Neural Networks, although being trained on image labels only.
4. We explored related literature in the area of explainability methods for Human Action Recognition based on skeletal joints and noted the limited number of appropriate resources.
5. We described how the intuition behind grad-CAM can be used in the context of skeletal data. We proposed an adaptation of grad-CAM that accounts for the high number of dimensions of skeletal data and creates interpretable joints and bones visualizations.
6. We highlighted how grad-CAM can provide intuitive explanations for seemingly unreasonable failure cases.
7. We explored the influence of the chosen target layer to compute the gradients used in grad-CAM for two configurations of a pretrained ResGCNv2.0 model.

8. We analysed the spatial and temporal localization of CAM and grad-CAM and conducted an agreement study which illustrates that certain action classes are hard to identify based solely on skeletal data even for human subjects.

We created visualizations using different target layers (before and after the *Spatial Temporal Joint Attention Module*). We confirmed the importance of the attention module proposed by Yi-Fan Song et al. [8]. Visualizations generated using gradients from layers prior to the attention module lack temporal localization (i.e. the most informative joints are correctly identified, but in the wrong frames).

Our experiments indicate that grad-CAM provides visualizations that better illustrate the temporal dimension (the joints are activated only in the frames where the action is performed) and the spatial dimension (activated joints are the most informative joints) than CAM, in the case of *ResGCNv2.0*. Moreover, our visualizations prove that false skeletons (i.e. skeletons which are wrongly associated by the *Kinect Sensor* with an object in single-person actions) have no influence in the prediction process. Nevertheless, we also obtain inconsistent results that require further investigation.

Through our work, we are able to provide intuitive explanations of the model's prediction process and create trust in its decision. We illustrate that apparently irrational failure cases have intuitive explanations and certain action classes are hard to identify solely based on skeleton data and require further context. Our implementation represents a starting point which can be easily adapted to new convolutional architectures.

8.2 OUTLOOK AND FUTURE WORK

Explainable Artificial Intelligence is still a novel research field with with the greatest breakthroughs still left to be discovered. We believe that, through further research into the inner working mechanisms of Deep Learning models, unknown characteristics of data may be revealed, which can, in turn, reveal more about the nature of our reality.

In the future, we plan to develop a comprehensive framework that allows easy and efficient access to different explainability methods that can work with no further adaptation on the most common types of CNN architectures. We would like to dedicate extensive attention to studying how the inner representations of high-level concepts (weights of the last convolutional layers of the network) relate between different convolutional models and if they are similar.

We also intend to dedicate further attention into how explainability can be used to drive research in Molecular Biology and in the Health Industry. Using AI in such critical areas requires a high degree of trust, only XAI can create. Being able to rely on trustworthy AI systems to interpret large amounts of data and provide insight that would have otherwise been lost to the human's limited computational power, creates an unprecedented envir-

onment for research and innovation. Furthermore, looking into how machines interpret information can be beneficial in rethinking how we evaluate large amounts of data.

9 APPENDIX

9.1 APPENDIX A: GRAD-CAM RESULTS

Figures 9.1, 9.2, 9.3 display the generated grad-CAM visualizations overlapped onto the frames of the original skeleton data. The visualizations are generated using the feature maps of the last convolutional layer before the classification module and the specified gradients.

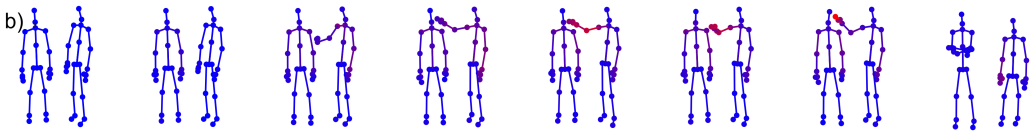


Figure 9.1: Performed action: Pat on back other person. Target layer: *conv_t*

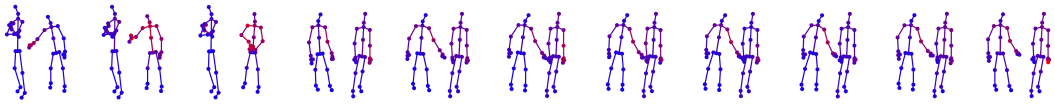


Figure 9.2: Performed action: Touch other person pocket. Target layer: *conv_v*

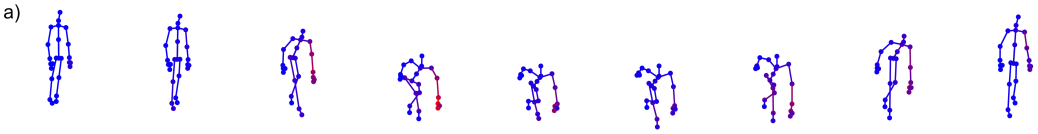


Figure 9.3: Performed action: Pick up. Target layer: *conv_v*

9.2 APPENDIX B: COMPARATIVE ANALYSIS BETWEEN CAM AND GRAD-CAM

In this section, we present comparative visualizations of CAM and grad-CAM for several actions from the NTU RGB+D 60 dataset (Figure 9.4, Figure 9.5, Figure 9.6). We use the Sandglass Layer Configuration for our model [8] and we use the last convolutional layer as the target for extracting the gradients and activation maps.

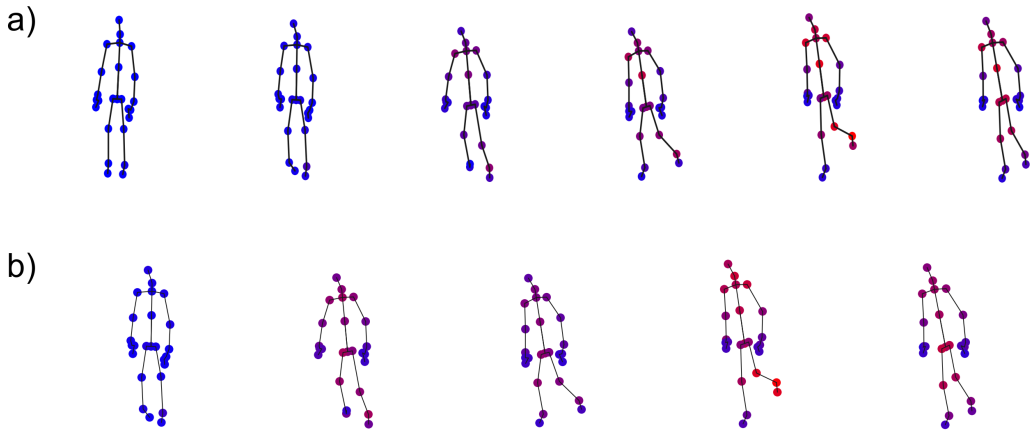


Figure 9.4: Performed action: Hopping. a) grad-CAM b) CAM

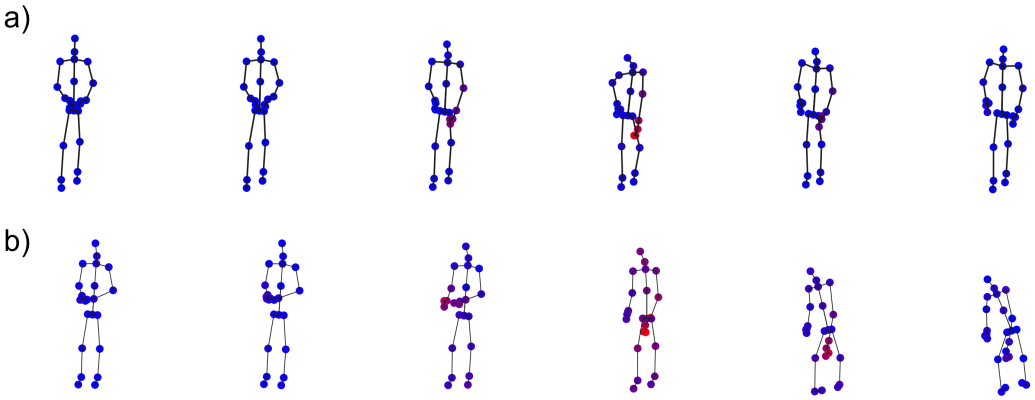


Figure 9.5: Performed action: Drop. a) grad-CAM b) CAM

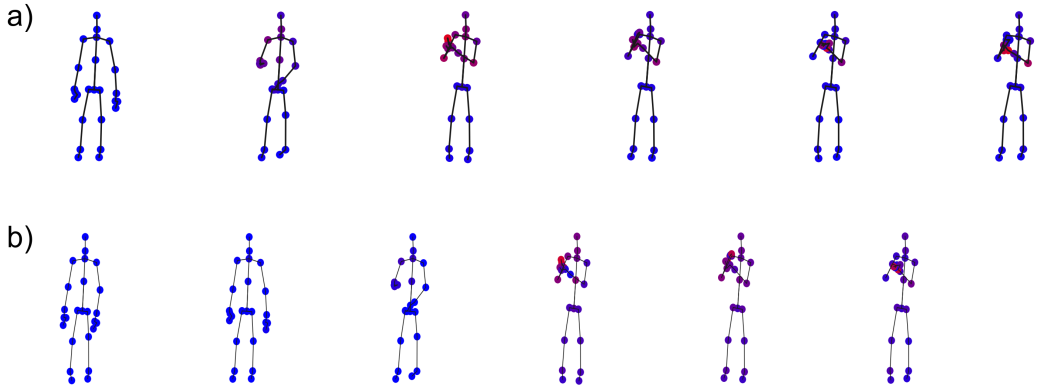


Figure 9.6: Performed action: Clapping. a) grad-CAM b) CAM

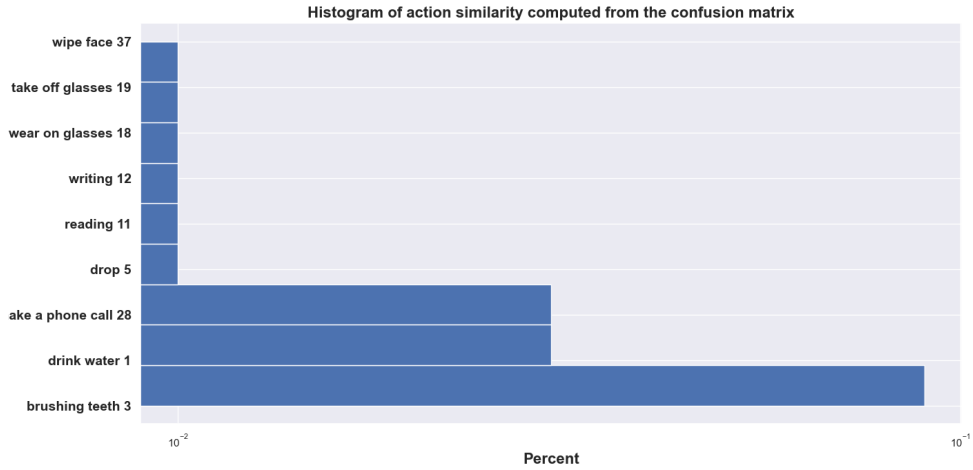


Figure 9.7: Action classes most confused with the action **Eat snack**

9.3 APPENDIX C: TARGET BLOCK CONFIGURATIONS

In this section, we describe the configuration of the two main temporal convolution blocks used in our target configuration (Listing 1, Listing 2) and the proposed Spatial Temporal Joint Attention Layer [8] (Listing 3).

9.4 APPENDIX D: AGREEMENT ANALYSIS

In this section, we present results of our agreement analysis. Our results suggest that some actions are hard to identify both by our model and by human subjects (Figure 7.8), based solely on skeleton data, while other action classes are easily recognizable, due to their characteristic nature (Figures 9.8 9.9).

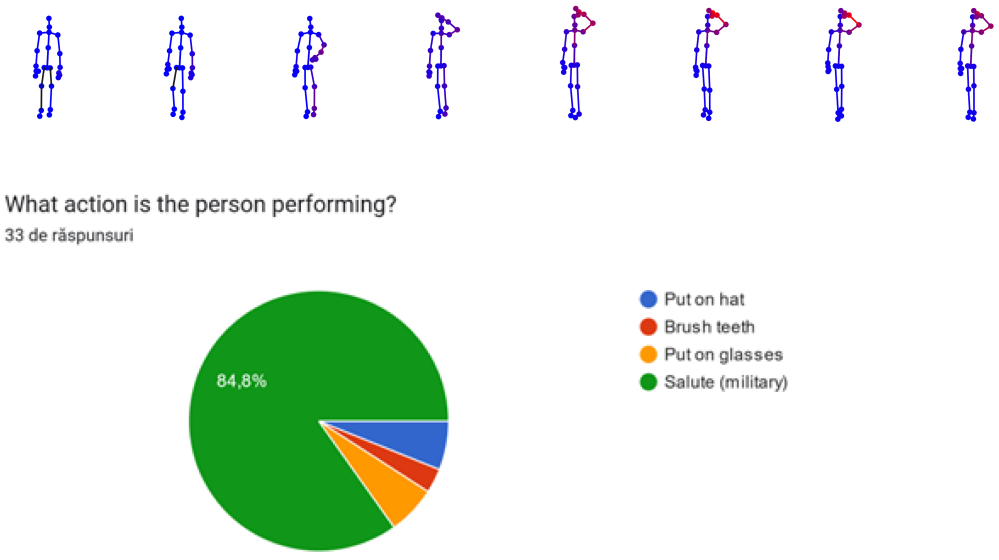


Figure 9.8: Agreement analysis conducted for the action class **Salute**

Listing 1: Expanded Separable Layer Configuration

```
Temporal_Sep_Layer(
  (act): Swish()
  (expand_conv): Sequential(
    (0): Conv2d(128, 256, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
        track_running_stats=True)
  )
  (depth_conv): Sequential(
    (0): Conv2d(256, 256, kernel_size=(5, 1), stride=(2, 1), padding=(2, 0),
        groups=256)
    (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
        track_running_stats=True)
  )
  (point_conv): Sequential(
    (0): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
        track_running_stats=True)
  )
  (residual): Sequential(
    (0): Conv2d(128, 128, kernel_size=(1, 1), stride=(2, 1))
    (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
        track_running_stats=True)
  )
)
```

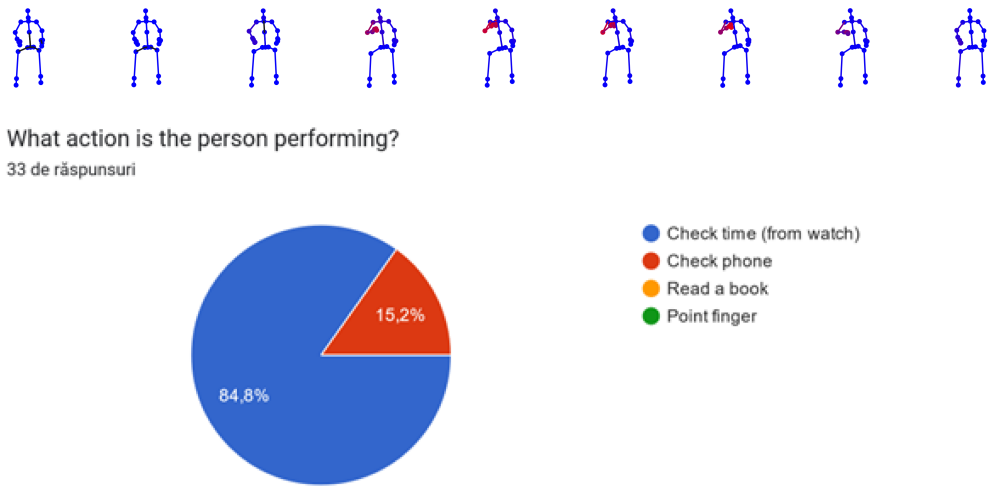


Figure 9.9: Agreement analysis conducted for the action class **Check time (from watch)**

Listing 2: Sandglass Layer Configuration

```
Temporal_SG_Layer(
  (act): Swish()
  (depth_conv1): Sequential(
    (0): Conv2d(128, 128, kernel_size=(5, 1), stride=(1, 1), padding=(2, 0),
        groups=128)
    (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
        track_running_stats=True)
  )
  (point_conv1): Sequential(
    (0): Conv2d(128, 64, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
        track_running_stats=True)
  )
  (point_conv2): Sequential(
    (0): Conv2d(64, 128, kernel_size=(1, 1), stride=(1, 1))
    (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
        track_running_stats=True)
  )
  (depth_conv2): Sequential(
    (0): Conv2d(128, 128, kernel_size=(5, 1), stride=(2, 1), padding=(2, 0),
        groups=128)
    (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
        track_running_stats=True)
  )
  (residual): Sequential(
    (0): Conv2d(128, 128, kernel_size=(1, 1), stride=(2, 1))
    (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
        track_running_stats=True)
  )
)
```

Listing 3: Spatial Temporal Joint Attention Layer Configuration

```
Attention_Layer(
  (att): ST_Joint_Att(
    (fcn): Sequential(
      (0): Conv2d(128, 64, kernel_size=(1, 1), stride=(1, 1))
      (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
          track_running_stats=True)
      (2): Hardswish()
    )
    (conv_t): Conv2d(64, 128, kernel_size=(1, 1), stride=(1, 1))
    (conv_v): Conv2d(64, 128, kernel_size=(1, 1), stride=(1, 1))
  )
  (bn): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
  (act): Swish()
)
```

BIBLIOGRAPHY

[1] B. Zhou, A. Khosla, L. A., A. Oliva and A. Torralba, 'Learning Deep Features for Discriminative Localization.', *CVPR*, 2016.

[2] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, 'Grad-CAM: Visual explanations from deep networks via gradient-based localization', *International Journal of Computer Vision*, vol. 128, no. 2, pp. 336–359, 2019. DOI: 10.1007/s11263-019-01228-7.

[3] E. A. Holm, 'In defense of the black box', *Science*, vol. 364, no. 6435, pp. 26–27, 2019. DOI: 10.1126/science.aax0162. eprint: https://www.science.org/doi/pdf/10.1126/science.aax0162. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.aax0162.

[4] G. Montavon, W. Samek and K.-R. Müller, 'Methods for interpreting and understanding deep neural networks', *Digital Signal Processing*, vol. 73, pp. 1–15, 2018, ISSN: 1051-2004. DOI: https://doi.org/10.1016/j.dsp.2017.10.011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1051200417302385.

[5] M. Ribeiro, S. Singh and C. Guestrin, '"why should i trust you?": Explaining the predictions of any classifier', Feb. 2016, pp. 97–101. DOI: 10.18653/v1/N16-3020.

[6] S. Laraba, 'Deep learning for skeleton-based human action recognition', Ph.D. dissertation, Oct. 2020. DOI: 10.13140/RG.2.2.30723.53283.

[7] S. Suzuki, Y. Amemiya and M. Sato, 'Skeleton-based explainable human activity recognition for child gross-motor assessment', in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, 2020, pp. 4015–4022. DOI: 10.1109/IECON43393.2020.9254361.

[8] Y.-F. Song, Z. Zhang, C. Shan and L. Wang, 'Constructing stronger and faster baselines for skeleton-based action recognition', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 1474–1488, 2023. DOI: 10.1109/TPAMI.2022.3157033.

[9] H. L. Dreyfus, *What Computers Cannot Do*. MIT Press, 1992, ISBN: 9780262540674.

[10] P. J. Denning and J. Arquilla, 'The context problem in artificial intelligence', *Commun. ACM*, vol. 65, no. 12, 18–21, 2022, ISSN: 0001-0782. DOI: 10.1145/3567605. [Online]. Available: https://doi.org/10.1145/3567605.

[11] S. Kaufman, S. Rosset and C. Perlich, 'Leakage in data mining: Formulation, detection, and avoidance', vol. 6, Jan. 2011, pp. 556–563. DOI: 10.1145/2020408.2020496.

[12] A. Adadi and M. Berrada, 'Peeking inside the black-box: A survey on explainable artificial intelligence (xai)', *IEEE Access*, vol. PP, pp. 1–1, Sep. 2018. DOI: 10.1109/ACCESS.2018.2870052.

[13] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm and N. Elhadad, 'Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission', in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '15, Sydney, NSW, Australia: Association for Computing Machinery, 2015, 1721–1730, ISBN: 9781450336642. DOI: 10.1145/2783258.2788613. [Online]. Available: https://doi.org/10.1145/2783258.2788613.

[14] D. H. Hubel and T. N. Wiesel, 'Receptive fields, binocular interaction and functional architecture in the cat's visual cortex', *The Journal of Physiology*, vol. 160, 1962.

[15] M. A. Silver and S. Kastner, 'Topographic maps in human frontal and parietal cortex', *Trends in Cognitive Sciences*, vol. 13, no. 11, pp. 488–495, 2009, ISSN: 1364-6613. DOI: https://doi.org/10.1016/j.tics.2009.08.005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364661309001739.

[16] S. Sabour, N. Frosst and G. E. Hinton, 'Dynamic routing between capsules', *Advances in neural information processing systems*, vol. 30, 2017.

[17] G. Hinton, S. Sabour and N. Frosst, 'Matrix capsules with em routing', 2018. [Online]. Available: https://openreview.net/pdf?id=HJWLfGWRb.

[18] K. Simonyan, A. Vedaldi and A. Zisserman, *Deep inside convolutional networks: Visualising image classification models and saliency maps*, 2014. arXiv: 1312.6034 [cs.CV].

[19] A. Shrikumar, P. Greenside and A. Kundaje, 'Learning important features through propagating activation differences', in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, Eds., ser. Proceedings of Machine Learning Research, vol. 70, PMLR, 2017, pp. 3145–3153.

[20] J. T. Springenberg, A. Dosovitskiy, T. Brox and M. Riedmiller, *Striving for simplicity: The all convolutional net*, 2015. arXiv: 1412.6806 [cs.LG].

[21] A. Bobick and J. Davis, 'The recognition of human movement using temporal templates', *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, pp. 257 –267, Apr. 2001. DOI: 10.1109/34.910878.

[22] Z. Daquan, Q. Hou, Y. Chen, J. Feng and S. Yan, 'Rethinking bottleneck structure for efficient mobile network design', in Dec. 2020, pp. 680–697, ISBN: 978-3-030-58579-2. DOI: 10.1007/978-3-030-58580-8_40.

[23] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, 'Mobilenetv2: Inverted residuals and linear bottlenecks', in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520. DOI: 10.1109/CVPR.2018.00474.

[24] A. Shahroudy, J. Liu, T.-T. Ng and G. Wang, 'Ntu rgb+d: A large scale dataset for 3d human activity analysis', in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1010–1019. DOI: 10.1109/CVPR.2016.115.

COLOPHON

This document was authored using TeXstudio[1] and typeset based on the La Trobe PhD Thesis Template[2] (customization of the classicthesis[3] LaTeX template).

1 https://www.texstudio.org
2 https://github.com/bashimao/ltu-thesis
3 https://bitbucket.org/amiede/classicthesis