Explainable Deep Learning for the Application to Multimodal Data

Rupayan Mallick

THÈSE PRÉSENTÉE

POUR OBTENIR LE GRADE DE

DOCTEUR
DE L'UNIVERSITÉ DE BORDEAUX

Ecole Doctorale de Mathématiques et Informatique

SPECIALITE EN INFORMATIQUE

Par **Rupayan Mallick**

Explainable Deep Learning for the Application to Multimodal Data

Sous la direction de: **Prof. Jenny Benois-Pineau**
Co-Directeur : **Prof. Akka Zemmari**

Soutenue le 20 October 2023

Membres du jury :

Professor Jenny Benois-Pineau	Professeur	Université de Bordeaux	Directrice
Professor Akka Zemmari	Professeur	Université de Bordeaux	Co-Directeur
Professor Alexandre Benoit	Professeur	Université Savoie Mont Blanc	Rapporteur
Professor Chaabane Djerba	Professeur	Université de Lille	Rapporteur
Professor Pascal Desbarats	Professeur	Université de Bordeaux	Président
Professor Cathal Gurrin	Associate Professeur	Dublin City University	Examinatrice
Professor François Bremond	Directeur de Recherche	INRIA Sophia-Antipolis	Invité
Professor Helene Amieva	Professeur	INSERM BPH	Invitée

Dedicated to Maa, Baba and Dada

Abstract

The progress of deep neural networks in the last decade across the domains has led to concern about the black-box nature of these models. For the trustworthiness of deep neural networks, as deep neural networks are inherently considered opaque and black-box in nature explanation of the decisions in a human-understandable manner is an open problem. Domains with high-stakes decisions such as judicial crimes, healthcare, social media, and finance, are extremely vulnerable to the decision by deep neural models. Recently, with the advent of deep neural models such as transformers, the increasing complexity and number of parameters make explainability in a human-understandable manner more important.

The work presented in this thesis, can be divided into two parts, first developing a multimodal network targeted towards the application of risk detection. The data for risk detection consists of egocentric videos and signal data acquired from various physiological and motion sensors. As the acquisition of the data is in a real-world scenario, there are several challenges that arise for the use of this multimedia

data using multimodal networks, i) weak synchronization of the data between the modalities, ii) data missingness, iii) understanding the representation between the modalities. To develop the multimodal network, at first we study the signals from various sensors, to benchmark our model for the use of sensors we use sensor-based human activity recognition datasets. Next, we develop our multimodal networks for visual and sensor data. For the video data, we benchmarked using a large-scale human action recognition dataset.

For our next part, we develop explainability methods for the transformers, more specifically the vision transformers (saliency-based), in this, we evaluate our method w.r.t, the human-attention-based gaze fixation system. For the video-based system, we developed a model for highlighting the temporal importance of the frames. This developed model is used on the visual data of the risk detection system and benchmarked on a large-scale human action dataset. Next, we take leverage of our explanability method and extend to use this method for better generalization of our multimodal system. The two forms of multimodal data representation have been tested, one the intermediate fusion in the feature space and the next late fusion in the decision space. In this work, we also have touched upon robustness and domain generalization using the interpretation of the models.

Résumé

Le travail présenté dans cette thèse peut être divisé en deux parties. La première partie concerne le développement d'un réseau multimodal destiné à l'application de la détection des risques des personnes fragiles dans l'environnement à domicile. Les données consistent en des vidéos égocentriques et des signaux acquis à partir de divers capteurs physiologiques et de mouvement. Comme l'acquisition des données se fait dans un scénario réel, l'utilisation de ces données complexes dans des réseaux multimodaux pose plusieurs problèmes : i) la faible synchronisation des données entre les modalités, ii) l'absence de données, iii) la compréhension de la représentation entre les modalités. Pour développer un réseau véritablement multimodal, nous nous concentrons d'abord sur les composants uni-modaux, concevons et évaluons nos modèles sur des ensembles de données uni-modales libres d'accès. Ensuite, les modèles sont fusionnés dans une architecture multimodale pour prendre des décisions sur des données multimodales réelles. L'une des configurations que nous avons proposées est un transformer multimodal. Les deux formes de fusion d'informations ont

été étudiées : i) la fusion intermédiaire dans l'espace des caractéristiques et ii) la fusion tardive dans l'espace de décision.

Dans la deuxième partie de la thèse, nous développons des méthodes d'explicitation pour les transformers, plus particulièrement les transformers visuels. Nous avons évalué notre méthode en termes de plausibilité des explications obtenues par rapport aux cartes de densité de fixations du regard humain. Cette partie du travail a été réalisée sur un ensemble de données d'images fixes. Notre objectif étant de développer des solutions pour l'analyse d'informations temporelles, telles que la vidéo, et sur la base de la philosophie de l'importance par l'explication, nous avons proposé un modèle pour mettre en évidence l'importance temporelle des images dans la vidéo. Ce modèle a été utilisé sur les données visuelles du système de détection des risques et comparé à un ensemble de données à grande échelle sur les actions humaines. Ensuite, nous tirons parti de notre méthode d'explicabilité proposée et l'utilisons pour une meilleure généralisation du transformer multimodal proposé. En effet, l'utilisation de techniques d'explicabilité dans les transformers multimodaux permet d'augmenter la précision de ces classificateurs sur des données complexes du monde réel et ouvre des perspectives intéressantes pour les études sur l'éparcité et la robustesse de ces approches.

Acknowledgements

First and foremost I would to thank my supervisors Prof. Jenny Benois-Pineau and Prof. Akka Zemmari for their continual support throughout my doctoral journey. I am grateful to my supervisors for providing me with this opportunity. I appreciate their time and effort from the very first day in listening and supporting my ideas and developing them. I really appreciate the meeting sessions discussing a number of ideas that have been a very rich experience for me. I would like to especially thank you both for the constant support during the lockdown period as it was just after joining the lab.

I would like to thank my reviewers Prof. Alexandre Benoit, and Prof. Chaabane Djerba for agreeing to review my thesis. My other jury members Prof. Pascal Desbarats, and Prof. Cathal Gurrin as an examiner. I would also like to thank all my mentors and teachers from my Master's for igniting my interest in machine learning, deep learning, and computer vision.

I would humbly extend my gratitude towards all my dear collaborators Prof.

Helene Amieva, and Prof. Laura Middleton, for the collaboration in the research we accomplished together. I would especially like to mention Boris Mansencal for all the technical support he provided during my research.

This thesis would not have been possible without the funding from EDMI. I am extremely thankful to EDMI for the grant during these last three years. I am very grateful to Prof. Laurent Simon and Prof. Romain Bourqui for being on my thesis committee and keeping track of the progress of my thesis.

I am extremely thankful to my parents, Mr. Partha Sarathi Mallick, and Mrs. Mousumi Mallick to whom I am forever indebted for their constant support. I would also like to mention my brother Mr. Deepayan Mallick for being the great support he is.

I would also like to thank AfoDIB, the computer science Ph.D. students association for conducting all the events throughout the year. This journey would not have been possible without my colleagues Dr. Thinhinane Yebda, and Dr. Marion Pech to whom I will always be grateful. I am also thankful to all my lab mates in LaBRI for all the lunches and events we did together.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Deep learning research has been at the forefront of many domains such as computer vision, Natural Language Processing (NLP), etc. for the past decade. A deep neural network model is believed to mimic the biological neurons of humans, allowing networks to extract information from various data modalities in a task of content understanding. With the increase of data acquisition devices, data has been abundant, making deep neural networks highly successful. In the real world, the use of multimodal data is evident, as the human cognitive system uses more than one stimulus/modality for the understanding of the surrounding world. Thus, while using deep neural networks, the use of multimodal data eases the understanding of concepts in the real world. But before combining modalities, it is prevalent to use single modalities. Image and video representation is very important in computer vision, and image analysis constitutes primarily many tasks such as understanding the scene, its variations, and attributes, etc. Deep neural networks have achieved

tremendous results in classification, segmentation, and object detection. This also resulted in the use of these networks for video analysis since videos are fundamentally the combination of image frames along the temporal dimension, i.e., videos have an added temporal dimensionality along with the spatial dimensionality. With recent advancements in language and vision projects such as image and video captioning, this research field brings some unique challenges for multimodal researchers given the heterogeneity of the data and the contingency often found between modalities. In this thesis, we use fusion and alignment as there is a correlation between the modalities for the prediction of concepts.

The challenges for multimodal machine learning are multimodal fusion, co-learning, multimodal representation learning, alignment of modality [11], etc. These include attention models, auto-encoders, and multimodal recurrent networks. The representation of multimodal data is important for the performance of deep neural networks. The ideal form of representation, as stated by Bengio et al. [14], comprises smoothness, temporal and spatial coherence, sparsity, and natural clustering, among others. The representation can be of two types: i) joint and ii) coordinated. These two types of representations are illustrated in Figure 1.1. Joint representations present an unimodal representation in multimodal space. For joint representations, early fusion techniques are used in the input space; i.e., it can be understood as concatenating features from individual modalities. For representing the features from individual modalities, neural networks (NNs) can be used since they are successive blocks of the inner product followed by the non-linear activation functions. That is unimodal

representations can be projected to a joint space by them. In a coordinated space, for the features from the individual modality, separate representations are learned in a coordinated fashion under a constraint. One of the examples of coordinated representation is computing similarity between features in a coordinated space. The use of joint representations is especially when all the modalities are present during inference time. Similarly, coordinated representations are majorly used for applications where only one modality is present in the inference time. The computation of the similarity between the features reduces the distance between the features from different individual modalities. The type of representation in multimodal learning is dependent on the type of information we are trying to extract. For example, in the task of sentiment analysis, semantic information is extracted from multiple modalities for the analysis of the expressed emotions [147]. Similarly, a very common and well-studied topic of action detection may use more than one modality to localize instances of action in the temporal domain [27, 37]. Video captioning uses the video frames and speech transcribed by automatic speech recognition as input and predicts a caption. The captions can be generated using a generative network in a joint space. There are various other examples that can be demonstrated as to how the representation of a multimodal network is used.

In recent years, the **explainability or interpretability** of decisions of DNNs has gained importance due to the nature of the application the networks are used for. There are applications such as in health care, law enforcement systems, finance, etc.,

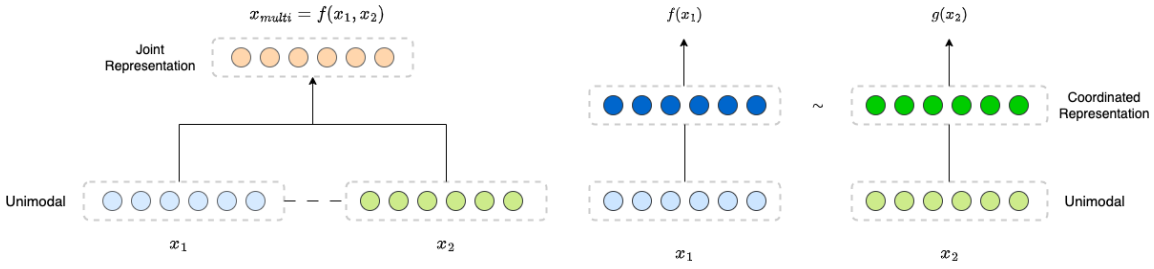

Figure 1.1: Illustration of two different types of representation for multimodal learning.

that leverage deep neural networks. Due to the black-box nature of the late, their decisions are not human-understandable. This implies a lack of transparency when used for high-stake decisions for these models. A black-box model can be a function that is difficult for humans to comprehend. There is the myth that the greater the complexity of the model, the greater the predictive performance is. However, as stated in [110], it depends on the data, if the data are structured with a good representation, then there is no significant difference in the predictive performance between models. In addition to multimodal learning, explaining decisions plays an important role. In this thesis, one of the novel applicative domains on which to focus is the use of weakly synchronized signals along with videos for the classification of risk events. With the compelling performances of the deep neural networks and a number of training parameters ranging from millions to billions it is of prime importance to understand the question *what are the features responsible for the prediction of the neural networks?*.

1.1 Problem Statement

This thesis is predominantly performing two tasks:

- Multimodal Learning for the prediction of risk events among old and frail individuals.
- Interpreting the decisions and using this interpretation for optimizing the predictive performance of the models.

Multimodal learning for the prediction of risk events encompasses learning from data obtained from multiple modalities. The general overview for the prediction of risk events is obtained by obtaining the videos using the ego-centric camera and obtaining physiological and motion data from sensors. These data are manually annotated using an annotation interface. Thus, the localization of the risk events for network training is performed manually, that is, data are segmented using the aforementioned annotation interface. Once these annotated data are obtained, we use them for the prediction of risk events. For this task, we use real-world data, and it is important to understand that these events are extremely rare in nature. The other task is the interpretation of the decisions as predicted by the network. We describe these tasks in detail in the following sections:

1.1.1 Multimodal Learning

To understand multimodal learning [11], first, the representation needs to be understood i.e, if there is a joint representation such that the data from the first

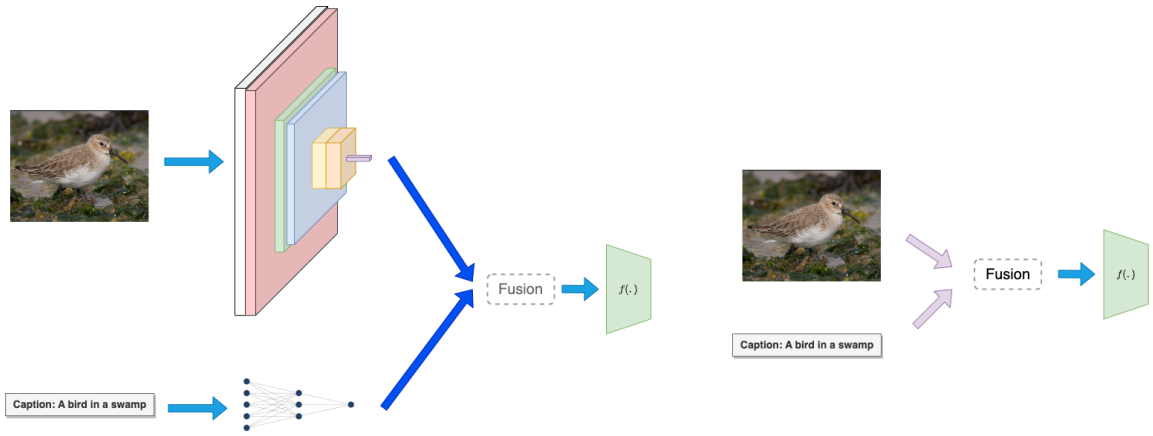

Figure 1.2: Illustration of (right) early and (left) intermediate fusion strategy.

modality is denoted by $\{x_1^1, x_2^1, x_3^1, ..., x_n^1\}$ and from the second modality is given by $\{x_1^2, x_2^2, x_3^2, ..., x_n^2\}$, then after the projection, the distribution in the joint space can be denoted as $\{\hat{x}_1, \hat{x}_2, \hat{x}_3, ..., \hat{x}_n\}$. From this joint distribution, the predictive model can be $\{y_i\} = f(\hat{x}_i; w)$ where f is a neural model and y_i is the predicted label. To obtain the distribution in the joint space, neural networks can be used on the individual modality i.e., $\{{l_i}^1\} = f_1(x_i^1; w_1)$ and $\{{l_i}^2\} = f_2(x_i^2; w_2)$ given f_1 and f_2 are neural models for the extraction of information to project in a common space $\{\hat{x}_i\} = g({l_i}^1, {l_i}^2)$ where g can be a function to combine in a common joint space.

The coordinated representation can be also termed as the multiplicative representation [49]. In this representation, the similarities between the output distributions of the individual modalities obtained from the respective neural networks are computed and it is obtained as $f_1(x_i^1; w_1) \sim f_2(x_i^2; w_2)$. The *early and intermediate fusion* as illustrated in Figure 1.2 techniques can be categorized in the joint representation space whereas the *late fusion* technique can be categorized in the coordinated representation space.

1.1.2 Interpretation of Model Decisions

Interpretability and Explainability are often used interchangeably in the literature, but the objective of both terms is to make certain properties of the model in a human-understandable form. Prior to understanding the objective of explanations, there are certain categories of explanations that need to be studied. The categories can be divided into self-interpretable models, post-hoc explanations, explanations by examples, and explanations by concepts. The objective and the task definition change depending on the form of explanations we talk about. The two most common explanation methods in the literature are post-hoc explanation methods and self-interpretable models. We will understand these two forms of explanations here and the rest are broadly discussed in Chapter 2.

- **Post-Hoc Explanations:** This type of explanation is crucial for systems where a human is not in control of the training process. The post-hoc explanations are considered mostly during the inference time; i.e., this type of explanation is used at the post-training time. To understand it's objective we take input data as $\{x_1, x_2, x_3, ..., x_n\} \in X$. Let $f(.)$ be a neural network for generalization in a supervised manner, and thus the output predicted labels can be written as $\{y_i\} = f(x_i; w)$. Thus the post-hoc explanations can either explain the model $f(;)$ at x_i i.e., $f(x_i)$, or understand how each feature is contributing or explain the whole input data i.e., $f(x_i)$ where $x_i \in X$. Thus, these two forms of explanation can be stated as local or global post hoc explanations. Global post hoc explanations aim to use a total understanding of the model parameters

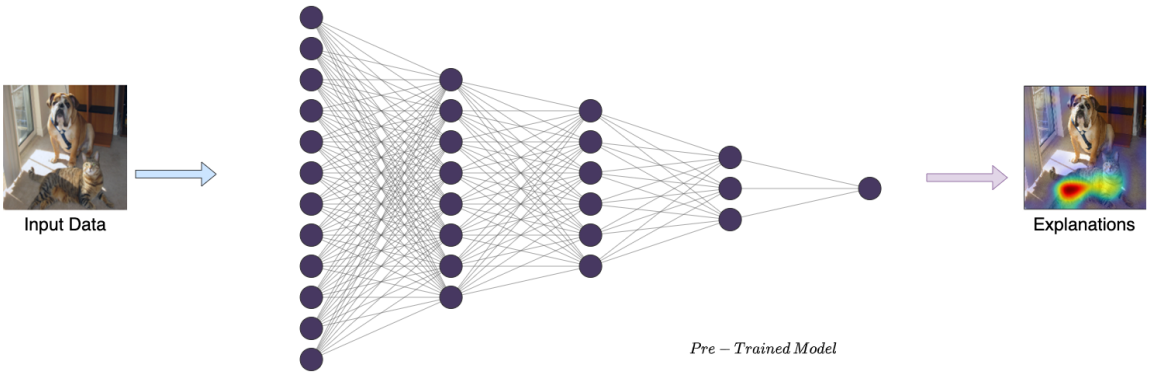

Figure 1.3: Post-Hoc interpretable method example, an input image is fed to the pre-trained network and using a post-hoc explanation method, the salient region of the image is highlighted

once the model is trained. Another form of post hoc analysis can be using a surrogate model $g(;)$ where it takes the neural network $f(;)$ and x_i as inputs and outputs and the importance vector v.

- **Self-Interpretable Models:** Most machine learning models, such as sparse linear models or decision trees, are self-interpretable models. These types of models output the predicted class and an importance vector for the explanation. Mathematically, this can be understood as $\{y_i\} = f(x_i; w)$ where $i \in \{1, 2, 3, ..., n\}$ where n is the dimension of the label space. Along with y_i, these types of models also produce an importance vector $v \in \mathrm{R}^N$ where N is the dimension of the feature space. There is a debate on the accuracy vs. interpretability trade-off while using self-interpretable models [133, 140], but it is difficult to evaluate and is not prominent in the literature. These types of models are very difficult for deep neural models and are not found in the literature to the best of our knowledge.

- The other forms of interpretable methods such as **feature-based explanations** are also quite common, especially for computer vision problems, as these highlight the features w.r.t. to the prediction. The highlighted important feature weights can be visualized in the form of a heatmap. There are two important methods for highlighting the features such as the gradient computation $\nabla f(x)$, which resembles the change of features. Perturbation-based methods can also be thought of as feature-based importance methods as we compute the feature importance by perturbing the features, e.g. for images, we can perturb pixels by gray values or some other color. Another way to compute the feature importance methods is by using a neighborhood of interest for a given feature and using a self-interpretable surrogate model such as a linear model for the explanations. The **example-importance explanations** use a single example as the prototype that can be used as representative samples (obtained from training data) during the prediction. Another family of explanations, i.e., **concept-based explanations** are based on human-based descriptions that match the training data. Concept-based explanations require human annotations that are difficult for large datasets and encompass all the concepts that are defined by humans. **Counterfactual examples** are another predominant concept to understand the underlying behavior of models. They are based on small changes in the input or features that can change the network prediction.

1.2 Applications

The advent of intelligent devices over the past few years, increased the multimodal application scenarios. One of the first multimodal applications studied is visual speech recognition. Other applications include (not exhaustive), vision and language navigation, human-computer interaction, healthcare, and surveillance.

1.2.1 Visual Speech Recognition

One of the prominent applications in current times is automatic visual speech recognition (AVSR) initially proposed in 1980s. It includes integrating two modalities, i.e., audio and visual systems to improve the performance in comparison to a single modality recognition system. In this application also correlation is important among the features between two modalities. One of the major challenges in AVSR is the noisy environment for speech recognition. Another challenge is the fusion of these two modalities as it uses different statistical patterns and properties among them. Thus multimodal learning can help in learning the joint representations among these two modalities.

1.2.2 Vision and Language Navigation

Vision and language navigation are fields that help systems interact with humans in natural language. This field requires expertise from natural language processing, computer vision, and robotics. This requires an embodied agent to navigate in a real or simulated environment and communicate with humans. tasks are important in a

way as the vision and language navigation problem has been approached in various manners such as leveraging the LSTM/recurrent networks along with attention,

1.2.3 Human-Computer Interaction

In human-computer interaction, the system tries to leverage keyboard-tipping, mouse-clicking, speech, touch, vision, and gestures. A multimodal human-computer interaction system facilitates human-like interaction. The human-like interaction between the computer and the user supported by multiple modal technologies has been applied in educational technology for a long time, especially in the field of pedagogical agents and intelligent tutoring systems. The intelligent systems can also be trained to trigger feedback from the reinforced learning system.

1.2.4 Healthcare

For healthcare, multimodal learning is very important for precision healthcare. Advances in deep learning have given the opportunity for predictions in many medical fields such as cardiovascular medicine, neurology, dermatology, ophthalmology, radiology, and other fields of medicine. It can predict the range of other health-related risks. One of the major contributing factors towards the success of deep neural networks in healthcare is the large amount of biomedical data such as Electronic Health Records (EHR). Generally, the individual data points are represented as vectors of attributes also known as features. These features are determined by the experts in the domain due to the heterogeneity of the concepts. Explainable methods are vital for healthcare applications due to their extensive potential in the detection of autism,

ADHD, developmental language disorders, etc.

1.2.5 Surveillance

For surveillance, the application of re-identification (ReID) plays an important role. With the explosion of monitoring by Close Circuit Television (CCTV) cameras across the world and the increase in surveillance, it is important for the system to first identify the person, then track the person within a single camera view, and finally use a ReID algorithm for multiple non-overlapping camera views. Another problem in ReID is the change of basic appearances, especially if the surveillance is long-term. An important problem in the past two/three years is wearing masks which makes it even more challenging. Surveillance is also important in healthcare settings such as centers for Alzheimer's patients, patients with Parkinson's, etc.

1.3 Scientific Problems and Challenges

Multimodal learning has been part of deep learning research for quite some time, but it faces several challenges in learning representations for proper semantic understanding. The problem remains challenging from representation to fusion for proper predictions.

1.3.1 Multimodal Representation Learning

For any deep neural network model, representation plays an important role because it can affect learning to some extent. What and how the representation is used can give a varied response on the same network. The representation can be used as joint or

coordinated as mentioned before. It can be based on the concatenation of the input features in the input space or projected to joint space forming joint representations that learn jointly from various modalities. The first challenge while any form of representation is the synchronization of the data across modalities. This challenge can be due to various factors such as missing data from a few of the modalities, etc. The challenge for representation also depends on the dimensionality of the data, i.e., the data can be either spatial, temporal, or spatiotemporal. It is vital to understand the task in hand for the representation, e.g., for long-term dependencies understanding the temporal modality is vital. Similarly, to understand fine-grained semantic information, spatial information is essential. The use of spatio-temporal information is also complicated, as it is important to understand the salient spatial features along with the temporal dependencies. In real-world datasets [32], such as for audio-visual learning, the presence of noise also affects the representations. It is important to extract the audio features and remove the noise from the audio signals. The removal of noise and missing of data is not limited to audio or speech processing but also other modalities.

1.3.2 Supervision

The supervision level depends on the task we are performing, as well as the amount of ground-truth labeled data available. For a fully supervised network to generalize well, there is a need for a large amount of annotated data, which in itself is a challenge, as annotation for large datasets can be an expensive and time-consuming

process. To reduce the dependency on large annotated datasets, it is essential for the neural networks to move from the supervised setting to unsupervised or semi-supervised settings. The objective in a supervised setting is to minimize the loss computed between the predicted values and ground truth. Although in my thesis, weak supervision is mostly used as we are dependent on the annotation of a single modality. In unimodal self-supervised learning, the assumption is that ground truth labels are not required, which alleviates the problem of annotating the data. Thus it is dependent on a pretext task for obtaining a pseudo-label which further can be used to supervise the neural networks. Thus, the loss is computed w.r.t. the pretext label rather than using an actual label.

Self-supervised multimodal learning is similar to self-supervised unimodal learning, instead of using ground truth labels, it uses pretext labels/pseudo labels for the minimization of the loss function. The pseudo-labels can be generated using some pretext tasks on a single modality or using joint information from multiple or a few of the modalities. There can be differences between unsupervised and self-supervised networks, such as generative models being unsupervised. One of the major challenges in the self-supervised setting is the requirement for large computational resources. To lower the dependency on large computational resources, certain efforts are put forward such as decoupled gradient accumulations [31], masked token dropping [74], parameter sharing among the modalities [119, 117, 55], sharing attention weights [13, 131] etc.

1.3.3 Imbalance of Classes

The imbalance of classes is a major challenge in real-world datasets. Generalizing a neural network with datasets having an imbalance in classes can give biased representations, thus also jeopardizing post-hoc explanations. There are certain methods in the literature to alleviate problems related to class imbalance, such as data augmentation techniques, rebalancing the classes by either removing examples from the majority class or generating more examples from the minority class, using focal loss [78], etc. Nevertheless, all these techniques have their own associated problems. Rebalancing samples may not give the true representation of the distributions as synthesizing synthetic data can be challenging if the samples for generation are not sufficient and removing samples may decrease the data for the neural networks to generalize properly. Changing loss functions such as the so-called focal loss (by using this loss we put more weight on the classes with less representation and less weight on the classes with higher representation on the dataset) can help, but the challenge is at times that it focuses a lot on the minority class overfitting the model.

1.3.4 Transferability

Transferability is a challenge across neural networks across domains, datasets, and applications. Data augmentation is an effective technique for data adaptation, while adversarial perturbations are important for improving generalization across datasets. A common scenario can be training in a dataset and inferring/predicting on another

dataset, and if the distribution gap between the training and inference data is significant, predictive performance decreases showing worse generalization. We state that the dataset trained on is the source domain and the predictive dataset for fine-tuning or inference as the target domain. Most of the finetuning requires generalizing well on the pre-trained models during training, this is due to the addition of novel classes in the target domain [7, 51]. Fine-tuning is important for the model to generalize well on the novel classes in the target domain and increment the performance of the fine-tuned model. Fine-tuned models also help in predictive performance on the novel dataset. The distribution gap can hinder generalization during the fine-tuning of the model. For multimodal datasets, it is more difficult majorly due to the cross-task distribution gap. This may be also due to the missing modalities in the inference time. In the literature, this is solved using knowledge distillation [56]. The latter consists of extracting features from a large deep-learning model and transferring the features to a smaller model.

1.3.5 Interpretability

For unimodal learning, interpretability is well studied, but the problem arises for multimodal learning. Interpretability offers various insights into the multimodal model, it is important for the model design, or debugging a dataset. Some interpretable methods can highlight the trends of explanatory features across the *whole dataset* providing global explanations. There are certain methods that focus on *individual* examples to provide deeper insight into the importance of features. Interpretability

for multimodal learning is challenging, as ideally, it should not only understand the importance of the single modality, i.e., unimodal explanatory features but also understand the relative importance of the interactions between different modalities to provide multimodal explanatory features. The literature is mostly understudied for interpretability while modeling multiple modalities. Few of the methods highlight the importance of metadata using shapely values[84], while the importance of visual modalities such as images is highlighted using attribution/saliency maps. Applications requiring multimodal interactions such as image captioning, visual question answering (VQA), visual entailment, etc, and feature alignment across different feature spaces also make it difficult to derive a multimodal explainable method.

1.4 Contributions

Our first contribution is developing a multimodal representation learning model targeted towards the application of detection of risk situations. In these contributions, we have tested the unimodal scenarios as well as furthering these models by complementing an added modality for multimodal scenarios. We have proposed multimodal architectures and algorithms based on Gated Recurrent Unit (GRUs) with attention to sequences of signal data where we have compared our results extensively with the other sequence models such as LSTMs, vanilla GRUs etc., and 3D-ConvNets for video modality. One of our main contributions has been proposing a two-stream architecture for the detection of risks. We have further extended the two-stream model to transformer-based architectures. In addition, we have also exploited the

self-attention networks for different modalities.

The work published in this thesis were published in the following venues:

- **R. Mallick**, T. Yebda, J. Benois-Pineau, A. Zemmari, M. Pech, and H. Amieva. A GRU Neural Network with Attention Mechanism for Detection of Risk Situations on Multimodal Lifelog Data. In CBMI, pages 1–6. IEEE, 2021
- **R. Mallick**, T. Yebda, J. Benois-Pineau, A. Zemmari, M. Pech, and H. Amieva. Detection of Risky Situations for Frail Adults with Hybrid Neural Networks on Multimodal Health Data. IEEE Multimedia, 29(1):7–17, 2022
- **Rupayan Mallick**, Jenny Benois-Pineau, Akka Zemmari, Marion Pech, Thinhinane Yebda, Helene Amieva and Laura Middleton. 2022 "A Hybrid Transformer Network for Detection of Risk Situations on Multimodal Life-Log Health Data". In 2022, International Conference in Multimedia Retrieval, Workshop on Intelligent Cross-Data Analysis and Retrieval
- **Rupayan Mallick**, Jenny Benois-Pineau, Akka Zemmari, Thinhinane Yebda, Marion Pech, Helene Amieva and Laura Middleton. "Pooling Transformer for Detection of Risk Events in In-The-Wild Video Ego Data". In: 2022 26th International Conference on Pattern Recognition (ICPR2022)
- **Rupayan Mallick**, Jenny Benois-Pineau, Akka Zemmari. "I SAW: A Self-Attention Weighted Method for Explanation of Visual Transformers", In: 2022 IEEE International Conference on Image Processing (ICIP) *(Oral Presentation)*
- **Rupayan Mallick**, Jenny Benois-Pineau, Akka Zemmari, Boris Mansencal,

Kamel Guerda, Helene Amieva, Laura Middleton. "Hybrid Transformer for Recognition of Risk Events in Multimodal Data with Close-Domain Transfer", In: Multimedia Tools and Applications (Under Review).

- **Rupayan Mallick**, Jenny Benois-Pineau, Akka Zemmari. "IFI: Interpreting for Improving: a Multimodal Transformer with an Interpretability Technique for Recognition of Risk Events". In: ACM Multimedia (Under Review).

In the contributions mentioned above, we have majorly focused on developing models based on the self-attention architecture due to its capability to capture long-range dependencies. In this contribution we use spatiotemporal data, and although there are a number of methodologies to localize the important spatial locations but very scarce literature on understanding the important temporal locations, especially on the self-attention-based architectures. We used a pooling approach for the computation of the importance of temporal locations. The following publication has been presented for this purpose. "Pooling Transformer for Detection of Risk Events in In-The-Wild Video Ego Data".

The interpretation of transformer-based models is still an open problem with limited works in the literature. Our next contribution is based on the localization of spatial regions responsible for the decision, thus we devised a novel approach to interpret a vision transformer [36].

Our final two contributions are to leverage the interpretability method and guide

the training based on the importance of spatial and temporal features for transformer-based models. We devised a novel method on this intuition and provided an improvement in the training algorithm.

For our contributions, we have mainly used the GPUs provided by the LaBRI. For some experiments especially for Chapter 7, we have trained our models in the Jean-Zay supercomputer, which uses the parallel distributed mechanism. For the experiments conducted in the LaBRI servers, we have used 2, 16 GB P100 GPUs and 2 46GB A40 GPUs. The Jean-Zay supercomputer provided us with several nodes of advanced GPUs such as 32 GB V100 and 40 GB A100. This helps us to use several parallel nodes helping to use large-scale datasets for our experiments.

1.5 Thesis Outline

Chapter 2, we discuss various state-of-the-art models for both learning of unimodal and multimodal representation of data, we also as well as we discuss in length different interpretation and explanation methods.

Chapter 3 presents the work initially on unimodal data comprising sensors, then extends this work to multimodal data consisting of signals from sensors and visual data. This chapter in length highlights the problems in the implementation of real-world multimodal data. In this chapter, we are concerned with the different problems that we face during the implementation of this multimodal data. We introduce the dataset Bio Immersive Risk Detection System (BIRDS) and describe the taxonomy

of the classes corresponding to the risk situations of fragile persons living at home.

Chapter 4, we devise a pooling based on the temporal dimension to obtain the important temporal locations in our temporal multimodal data. In this chapter, we use the singular modality of the videos and thus we use a fully self-attention-based transformer block. Previously in the literature, we have seen pooling helping to understand the important spatial locations such as in-class activation maps [151]. Further, this pooling approach helps to use these important temporal locations to improve the overall performance of the model.

Chapter 5 presents our novel method for the interpretation of vision transformers. This method is also applicable to other self-attention-based methods. In this work, we have introduced weighting the attention with the gradient of attention for the transformers. We evaluate our method w.r.t. human-based attention system.

A novel method for optimizing the training algorithm using feature importance by interpretability method is given in *Chapter 6*. In this, we added additional supervision using interpretable methods. We use this method for both video and signal modality. In *Chapter 7* we propose another novel method for better initialization using interpretable methods for videos and signals.

Finally, we summarise the thesis and discuss the future works in *Chapter 8*.

Chapter 2

State-of-the-Art

In this chapter, we look at the literature ranging back to the long dominance of deep neural networks for generalizing data for various tasks in different domains and modalities. First, we discuss various unimodal modeling strategies, and then we discuss multimodality. Finally, we discuss interpretability and explainability in both unimodal and multimodal scenarios. The explanation in the unimodal scenario is well-studied in the literature; for multimodality, it is very challenging and understudied.

2.1 Deep Neural Network Architectures

In many of the application domains, deep neural networks have been widely deployed for data mining and data classification that can be generalized well amongst most forms of data. However, the literature suggests using different networks for different forms of data, for example, text, images, videos, signals, etc. The nature of the data

is different, as a text paragraph needs context to make sense to humans; similarly, videos have temporal context in addition to spatial context. On the graph data, the nodes require to be learned which is represented in an adjacency matrix. Images have spatial information, which can be learned by localized filters. Signals can be considered as the point features with temporal information. But in the real world, the data consist of one or many of these modalities for humans to perceive. The predominant multimodal architectures can be seen in the application using modalities such as text and images [104], text and videos[76, 127], pose/flow with the videos[38, 90, 58], etc. Thus, these combinations of modalities can be synchronized or weakly supervised. In the following paragraphs, we will discuss different architectures for each of these modalities in detail:

The visual modality can be categorized as the image and video modality, in this section we discuss the image modality. Image modality has a huge significance, especially for applications related to computer vision. Convolution Neural Networks (CNNs) and their variants are most commonly used for image-related tasks due to their ability to capture local information, as the architectures are widely discussed in the literature. In this chapter, transformers are majorly discussed. Initially, this architecture was described for language representation tasks [128], but was adapted for images due to its strong representation ability in [36] by Dosovitskiy et al. There are other models that leverage the self-attention mechanism for images[81, 12, 122, 123]. Swin Transformer [81] is based on the shifted window approach with hierarchical architecture. Wang et al. [132] proposed the self-attention network for non-local

networks to capture long-range dependencies in images. Han et. al. [53] proposed a connection between local attention and convolution layers. Other architectures based on local attention are based on the transformer on [129, 109]. The transformers are used for other downstream tasks such as segmentation[137, 116], object detection[20, 70], pose estimation[138, 139] etc. Other vision tasks involve image generation[41, 146], inpainting[73] etc. Architectures such as 3D-CNN [124], LSTM-CNN [136], TimesFormer[16], ViViT[5], etc, extended the self-attention mechanism for videos. The videos are trained in a form that additional dimensions in addition to the spatial dimension are taken into consideration. Similarly to sequential tasks, transformers have helped in handling long-range sequences with contextual relationships of the videos.

Multimodal architectures have existed in the literature since the inception of multimodal data. Speech Recognition, Natural Language Processing, Text Generation, etc., used multimodal models to understand the underlying feature representation. One of the works [95] of audio-visual bimodal fusion uses shared hidden layer representation to understand the higher-level correlation between the audio and visual cues.

Other works such as CLIP [104] jointly train the image and the text encoder to get the correct parings in a batch to create the training examples. In the [99] work, the authors predict the temporal synchronization of audio and visual features. [126] uses a multi-modal transformer with cross attention to understand unaligned multi-modal

language sequences. These examples are numerous; however, multimodal architectures on visual information and sensor information remain rare [152]. Combining various modalities has been well studied in [80], using different fusion methods.

2.2 Interpretable Techniques

Due to the large number of parameters, it is difficult to circumvent the black-box nature of deep neural networks. The fundamental question of trust and accountability for a decision taken by a Deep Neural Network (DNN) is still being studied. There are a number of domains where reasoning is extremely important for the accountability and trust of decisions. Interpretability gives the rationale behind the decision given by deep models. Most of the explanations are qualitative in nature. The qualitative nature of explanations gives diversity to the nature of explainable methods. As given in [105], the diversity of explanations is due to what conforms to the notion of explanations in deep models.

2.2.1 Explanation Methods

The categorization of the explanation methods can be based on many taxonomies. The first of the taxonomy can be between ante-hoc methods [79] or intrinsically explainable methods and post-hoc methods. The basic principle of these methods is clearly explained in Chapter 1. The other form of categorization of explanations can be *black box* and *white box* methods as given in [48, 6].

The *black-box* methods do not consider the internal parameters and features for

computing the explanations rather it approximates a surrogate function (f_{surr}) to estimate how an input corresponds to the prediction. For white box methods, the assumption is model parameters and the DNN function is accessible thus these parameters can be used to obtain the explanations related to the DNN. One of the most common methods to access the parameters of a DNN is using the backpropagation algorithm. The other method perturbation-based method is majorly a black-box method.

Another categorization of explanation methods is divided into two methods such as *backpropagation* and *perturbation* based methods. Visualization methods highlight the characteristics of inputs or features that are responsible for making the decisions. Backpropagation methods are related to gradient accumulation during network training. In perturbation-based methods, the fundamental principle is modifying the input to study the change in the output.

2.2.1.1 Backpropagation Based Methods

This method is part of the ***white-box*** method as we are accessing the network parameters and their interactions during the backpropagation process. In backpropagation-based methods, we quantify how sensitive the output presented by the deep model w.r.t., the input features. The fundamental approach for the backpropagation-based methods is given by visualizing the derivative of the network to that of the input. Next, we discuss some of the backpropagation-based methods. One of the first works in this type of method is ***Activation Maximisation*** [40]. One of the advantages of this method is it is really simple and we can compute the importance of features

at any layer giving the visualization of the internal representations. In this method, the idea is to maximize the activation of a neural unit in a given layer by optimizing the input. This is performed by computing the gradient of the activation w.r.t. the input X. Optimizes X to find X^* in the direction of the gradient. Mathematically, it is given in Equation 2.1, where X is the input and θ are the parameters and $a_{i,j}$ is the obtained activation of the neuron between a neural unit i for the j^{th} layer:

$$X^* = \arg\max_X a_{i,j}(X, \theta) \tag{2.1}$$

One of the major drawbacks of this method is that it gives a global explanation i.e. w.r.t, the whole and not individual model predictions. To visualize the features from higher layers, Zeiler et. al. [145] proposed an algorithm called ***Deconvolution***. In this method, the author assumes the deep model as Convolutional Neural Networks (CNNs). The deconvolution network is devised as the inverse of the convolution networks. This is a top-down approach instead of bottom-up as presented in Convolutional Neural Network. The convolutional layers in CNNs are replaced by deconvolution layers. The kernels for deconvolutions are the transposed kernels in the convolution predictions. Unpooling layers are used instead of pooling layers in deconvolution layers. The deconvolution operation is simply the reverse of the convolution operations as given in Equation 2.2, where $\mathcal{A}^\ell$ is the output of the layer ℓ in the convolution operation, K is the learned filter of the convolution operation, b is the bias passed during the convolution operation, and s^ℓ is referred to as switches:

$$\mathcal{A}^{\ell-1} = \text{unpool}(\texttt{ReLU}(\mathcal{A}^{\ell} - b^{\ell}) * K^{\ell^T}), s^{\ell}) \tag{2.2}$$

The intuition behind this method is to see how much information is retained by the extracted features on each layer. Similar to activation maximization, this method too does not provide individual model predictions. Zhou et al. [151] provide the visualization method called *class activation maps* (CAMs). Class Activation Maps are created using global average pooling (GAP) [77] in convolutional models as presented in Figure 2.3. Lin et al. [77] proposed applying global average pooling to the layer before the fully connected layer (FC). For the classification task, the FC layer has a total of C nodes for C classes. To retrieve the activation maps, the weights between the convolution and fully connected layers are multiplied by the activation of the convolutional layer. Mathematically, it can be expressed as given in Equation 2.3, where $\mathcal{A}_k$ is the activation of the convolution layer containing the convolution filter k and $w_{k,c}$ are the weights between the convolution and the fully connected layer:

$$\text{map}_c = \sum_{k}^{K} w_{k,c}\mathcal{A}_k \tag{2.3}$$

This method is a class-dependent method, i.e. providing explanations w.r.t. the classes. In simple terms, we scale the obtained saliency/attribution map to the size of the input image. These attribution maps highlight the important regions of the input image w.r.t the classification giving a unique saliency map for each class. Another CAM-based method was proposed by Selvaraju et. al. [111] called ***Grad-CAM***.

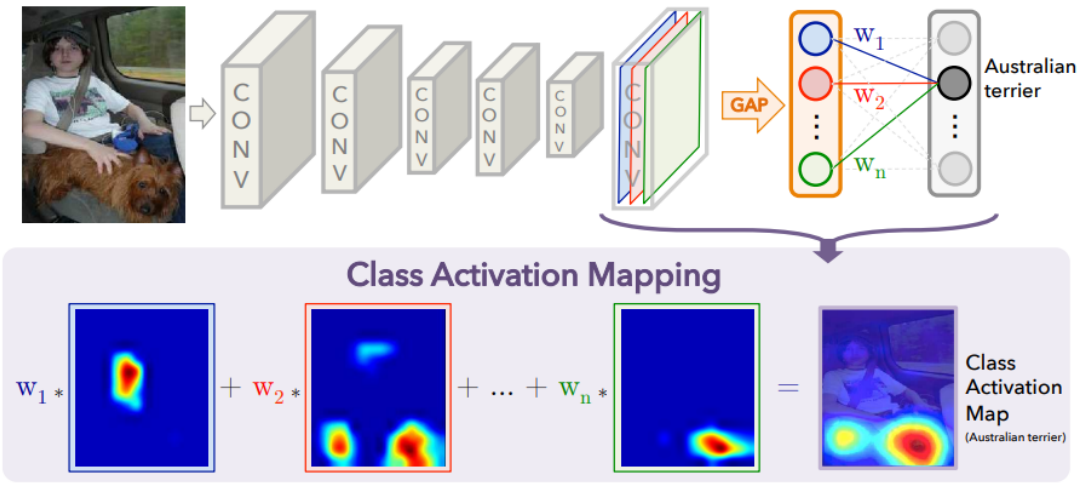

Figure 2.1: In this method the predicted class score is mapped back to the previous convolutional layer to generate the class activation maps (CAMs) to highlight the class-specific discriminative regions. [151]

This method generalizes the CAM by computing the gradient of the output of the network w.r.t., activation of the last convolution layer prior to the FC layer. Further, it is averaged across the dimension of the feature map $\mathcal{A}_k$ to obtain the importance score for the particular class c. The algorithm is expressed in the Equation 2.4 where the A_k is of size $m \times n$ and $\alpha_{k,c}$ is the importance computed.

$$\alpha_{k,c} = \frac{1}{m \cdot n} \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{\partial y_c}{\partial \mathcal{A}_{k,i,j}} \tag{2.4}$$

After obtaining the importance score, we compute the saliency map. The latter is computed similarly to that presented in Equation 2.3 but instead of multiplying the activation to that of the weights, it is weighted by the importance score and passed through the non-linear function ReLU. The Equation is given in 2.5

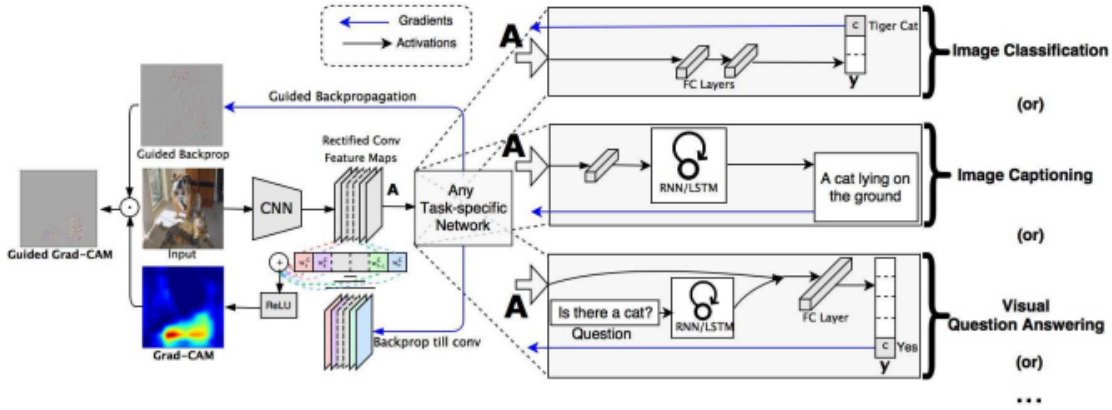

Figure 2.2: The overview of the GradCam method given a class of interest (tiger cat in this scenario). [111]

$$\mathrm{L}_{\mathrm{map}_c} = \texttt{ReLU}\Big(\sum_{k}^{K} \alpha_{k,c}\mathcal{A}_k\Big) \tag{2.5}$$

We use `ReLU` to obtain only the scores that have a positive influence w.r.t. the output class. Since the activation is weighted with the importance scores which in itself is computed by the gradient of the class scores to the activation feature map, this method is called Gradient-weighted Class Activation Map. Methods using class activation maps describe the sensitivity of the input features. The schema for the method is presented in Figure 2.2

A family of methods measures the relevance of input features to the output of the network. One of the most common relevance-based methods is ***Layer-wise Relevance Propagation***. In this method, the output of the deep neural network is decomposed into several relevance scores across the input features $x = \{x_1, x_2, x_3, ..., x_N\}$. LRP is computed using Deep Taylor Decomposition, where

the assumption is deep model $f(.)$ is differentiable and therefore can be approximated using Taylor expansion at some root value $\hat{x}$ such that $f(\hat{x}) = 0$. The Taylor expansion is given in the Equation where ϵ represents all second and higher-order terms:

$$f(x) = f(\hat{x}) + \nabla_{\hat{x}} f \cdot (x - \hat{x}) + \epsilon = \sum_{i}^{N} \frac{\partial f}{\partial x_i}(\hat{x_i}) \cdot (x_i - \hat{x}_i) + \epsilon \tag{2.6}$$

The relevance scores are the first-order partial derivative terms in the 2.6. The deep Taylor decomposition approach considers the conservation of the relevance scores across the layers, starting from the output layer through each intermediate layer and finally to the input. The relevance scores are given in the following Equation 2.7 for the layer ℓ and node i:

$$r_i^{\ell} = \sum_{j}^{M} r_{i,j}^{\ell} \tag{2.7}$$

The relevance scores are backpropagated from the last layer to the input space. Since `ReLU` is not applied in the LRP, the final heatmap shows the negative attributions. The chosen root value plays a vital role in the final visualization of the relevance maps in the image. There are two other common visualization methods such as ***DeepLIFT*** [112] and ***Integrated Gradients*** [120]. There are several other back-propagation-based visualization methods. Shrikumar et. al. [112] proposed ***DeepLIFT*** similar to LRP requires a reference image and computation of relevance and contribution scores. The core idea is to compute the contribution scores based on the difference between the input features x and reference image $\hat{x}$.

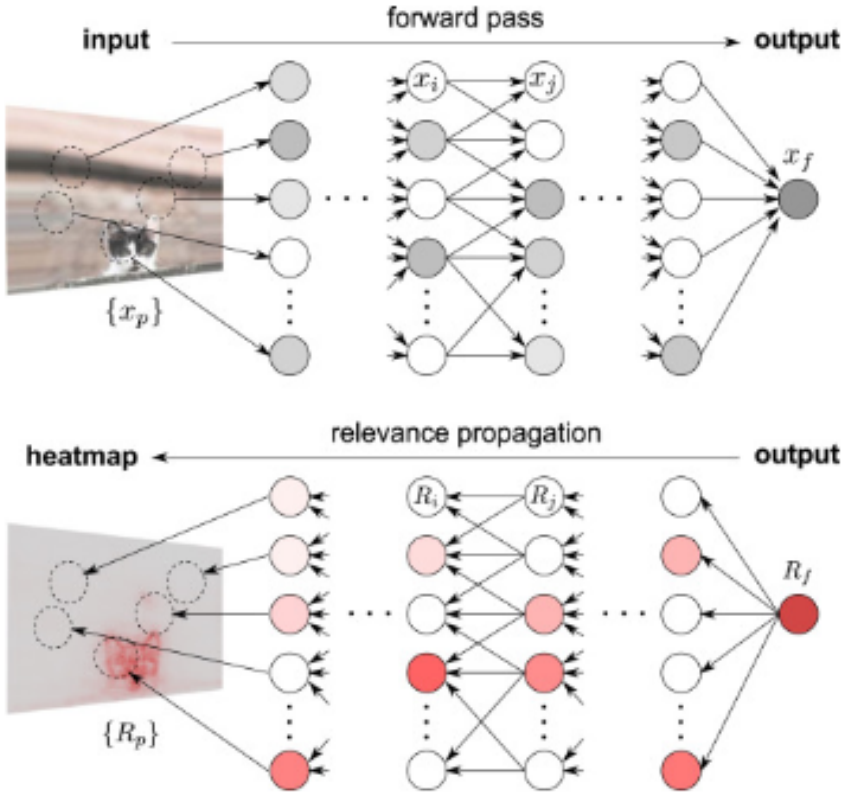

Figure 2.3: LRP [8] decomposes the prediction function as a sum of layerwise relevance values. It uses deep Taylor decomposition of the prediction. [48]

Let t be the output neuron and t^0, an output neuron for the reference point, therefore $\Delta t = t - t^0$. The contribution score $C_{\Delta x_i \Delta t}$ assigned to Δx_i such that :

$$\Delta t = \sum_{i=1}^{N} C_{\Delta x_i \Delta t} \tag{2.8}$$

The Equation (2.8) is called the summation-to-delta property, to simplify it can be thought of as the influencing factor of Δx_i on the Δt. The computation of contribution score can be computed using the Linear, Rescale, and RevealCancel Rule which is an approximation to the shapely values. We define a multiplier in

Equation (2.9) to assign the contribution of Δx with respect to Δt.

$$m_{\Delta x \Delta t} = \frac{C_{\Delta x \Delta t}}{\Delta x} \tag{2.9}$$

The multiplier is analogous to the partial derivative. We can also imply the chain rule for the multipliers as given in Equation 2.10 that is simply allowing us to compute it for all hidden layers in a layer-by-layer manner. In Equation 2.10 a_j are neurons in the hidden layer.

$$m_{\Delta x_i \Delta t} = \sum_j m_{\Delta x_i \Delta a_j} m_{\Delta a_j \Delta t} \tag{2.10}$$

Similarly to the LRP, the heatmap for the input image will be dependent on the reference image. Different reference images can produce various heatmaps for interpretation. This method produces positive and negative attributions. Sundarajan et al. [120] proposed another method that requires a reference image called ***Integrated Gradients***. This method consists of two axioms: i) sensitivity and ii) implementation invariance. The first axiom sensitivity is as: compared to a reference input $\hat{x}$ differing from the actual input x such that $f(x) \neq f(\hat{x})$ along the feature x_i, then the importance score for x_i must be non-zero. The second axiom implementation invariance is as: when the model outputs are equal for all the possible inputs available the importance score for x_i is equal for both networks say $f_1(;)$ and $f_2(;)$. Primarily, given a deep network $F : \mathrm{R}^n \rightarrow [0, 1]$, the integrated gradient for feature i is calculated as given in Equation 2.11.

$$\texttt{IntegratedGrads}_{\texttt{i}}(x) = (x_i - \hat{x}_i) \int_{\alpha=0}^{1} \frac{\partial F(\hat{x} + \alpha(x - \hat{x}))}{\partial x_i} d\alpha \tag{2.11}$$

The Equation (2.11) can be effectively approximated using the Riemann approximation as presented in Equation (2.12) where M is the number of steps of approximation.

$$\texttt{IntegratedGrads}_{\texttt{i}}(x) \simeq (x_i - \hat{x}_i) \times \sum_{k=1}^{M} \frac{\partial F(\hat{x} + \frac{k}{M}(x - \hat{x}))}{\partial x_i} \times \frac{1}{M} \tag{2.12}$$

For practicality, the reference image is taken as a black image while for text this can be taken as zero vector embedding. Overall, the intuition in this method is interpretation is cumulative sensitivity of F to the changes in the feature i on all the inputs between the straight line x and $\hat{x}$.

2.2.1.2 Perturbation Based Methods

For perturbation-based methods, the algorithm computes the interpretation by the difference of the network output when removing or occluding a part of the input features and actual input features. One of the primary perturbation-based methods is ***Sensitivity to Occlusion*** as proposed by Zeiler and Fergus et al. [145] In this method, the authors slide a gray patch across the input image to see the variation of the network output while the gray patch covers certain regions. The intuition is the relevance of information or the importance of image regions for the network output. If the network performance changes by a greater magnitude, the importance of that

particular region is higher compared to the occluding a region that has less decrease or change in magnitude of the network performance. This gives the correlation between the region of interest and the output of the network. Zhou et al. [150] introduced a similar method by forming a grid on the image and occluding it with gray squares on this grid. In these types of methods, the patch size, density, and shape of the patches may vary giving different results for different choices. One of the major drawbacks to this type of method is computational time, as the higher the resolution of the heatmap, the smaller the size of the occluding patches. As this method concentrates on a particular region of interest, another drawback is the multiple regions of interest in the input. Li et al. [71] proposed a method for natural language tasks, by removing a textual embedding or setting the dimension to zero for hidden activation. This is evaluated by using reinforcement learning to get feedback on the network decision on multiple text embedding. An advantage of this method is handling the sensitivity with occlusion to a combination of regions of interest. The use of reinforcement learning helps in finding the minimum change in the input features to alter the network decisions.

In the paper [44], the authors propose explanations as meta-predictors. For a particular class c, we define a neural network f, i.e. this network only classifies for the class c. To explain the behavior of the c classifier, the rule is given in Equation (2.13) where $\mathcal{X}_c$ is a set with all instances of class c and $\mathcal{X}_c \subset \mathcal{X}$ and $f(x) = +1$ means the presence of class c where Q_1 is the rule for the *local explanation*.

$$Q_1(x; f) = \{x \in \mathcal{X}_c \Leftrightarrow f(x) = +1\} \tag{2.13}$$

As this is a perturbation-based method, the authors use three different types of perturbation i) replacing regions with constant values, ii) replacing regions with noise, and iii) blurring the regions. This method uses a form of local explanations, as for a specific image x^0, the visualization is obtained by perturbations. The perturbations show the sensitivity of the neural network $f(x^0)$ to different regions of x^0. Since it is a form of local explanation, the perturbation of a given input instance is kept to a minimum resulting in a much more concentrated saliency map and fewer spurious locations. Another rigorous approach uses the deletion of information from the input space to measure the influence of the network output. This approach is based on the principle by [108] where they compute the marginal probability $p(c|x_{-i})$ by keeping and deleting a certain feature x_i and x_{-i} is the features except x_i.

$$p(c|x_{-i}) = \sum_{x_i} p(x_i|x_{-i})p(c|x_{-i}, x_i) \tag{2.14}$$

Using Equation (2.14), the importance/relevance score is calculated as the prediction difference as given in Equation (2.15).

$$\text{Diff}_i(c|x) = log\Big(\frac{p(c|x)}{1 - p(c|x)}\Big) - log\Big(\frac{p(c|x_i)}{1 - p(c|x_{-i})}\Big) \tag{2.15}$$

Zintgraf et al. [154] improved the prediction difference by sampling the patches instead of the pixels, patches give better spatial context compared to pixels which

in itself increases the robustness. Finally, this method helps alter the intermediate activations and evaluate the effect on downstream layers.

2.2.1.3 Local Approximations

For local approximations, a surrogate model is developed on a subset of inputs to mimic the decisions by deep neural networks. The small subset of inputs is approximated within a small neighborhood or a subspace of the input data say x_i. The data subsets are chosen with similar feature values. According to Baehrens et al., local approximations have been generated in [9] where they present a vector defined by the derivative of conditional probability. The direction and magnitude of the derivatives at x_0 along the data space define a vector field that characterizes the flow away from a corresponding class.

Ribiero et al.,[106] proposed a very popular method named **Local Interpretable Model-Agnostic Explanations**. This method is a post-hoc method, where a surrogate model is used to explain the deep model. The surrogate model used is termed the interpretable model. Let $f(;)$ be the deep model and $g(;)$, be the surrogate/interpretable model. $g \in G$, where G is a class of inherently interpretable models, such as linear regression models, decision trees, etc. For decision trees and regression models to be interpretable, it is important to realize the complexity of the interpretable model (e.g. the depth of the decision trees) which is denoted by $\Omega(G)$. Let $x \in \mathbb{R}^d$ be the original input representation to the deep model f and x' as the interpretable representation such that $x' \in \{0,1\}^{d'}$ as the domain for g is $\mathbb{R}^{d'}$. When defining the locality of x, we define the proximity parameter $\Pi_x(z)$. The proximity parameter

defines the proximity of the function as the proximity of the perturbed data points z to the original data point x. A loss term $\mathcal{L}(f, g, \Pi_x)$ defines the unfaithfulness of g in approximating f within the locality Π_x. Thus, the objective is to minimize this loss term as given in Equation 2.16:

$$\varepsilon(x) = \arg\min_{g \in G} \{\mathcal{L}(f, g, \Pi_x) + \Omega(G)\} \tag{2.16}$$

Given a perturbed sample $z \in \{0, 1\}^{d'}$ (which contains a fraction of nonzero elements of x'), we recover the sample in the original representation $z \in \mathbb{R}^{d'}$ and obtain $f(z)$, which is used as a label for the explanation model. Given this dataset $\mathcal{Z}$ of perturbed samples with associated labels, we optimize equation 2.16 to get an explanation $\varepsilon(x)$. $\mathcal{Z}$ is obtained to train the interpretable model and $\mathcal{Z} = \{z', f(z), \Pi_x(z)\}$. This method is model-agnostic and generalizes the surrogate model around the local neighborhood of the reference image. Therefore, only a single local interpretable model for a set of similar inputs is needed. This method works well with data distribution that has low variance. An example after the implementation of LIME on an image is shown in Figure 2.4. Lundberg et al. [84] demonstrated another method by computing Shapely values for the input features. A perturbed input is provided to the model.

2.2.1.4 Attention Mechanisms

Attention mechanisms can be considered an intrinsic method and can be considered one of the inherent explainability methods. The placement of the attention

(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Figure 2.4: Explanation of different predictions in an image trained on Inception network. [106]

mechanism can capture semantic information in the earlier layers and determine the fine-grained feature interaction among the pixel tokens if placed on later layers. The output and explanation can be obtained simultaneously in this method. Attention visualizations can be considered inherent visualizations. To compute the attention weights, several ways have been proposed in literature such as computing cosine similarity [130], the dot product of matrices [85], additive model structure [10], etc. The attention mechanism was earlier used for sequential tasks, multimodal fusion, etc. but lately, attention has been used for most of the applicative tasks to improve the performance of deep neural models. In simple terms, attention mechanisms provide show the weighting of the input features. Prior to the use of attention for visual data, attention was used for text processing. Natural language tasks such as language translation [128, 10, 85], sentiment analysis [33, 69], etc., use an attention mechanism for better performance of the deep models. This can be of two types such as self-attention or simply attention. For self-attention, the computation of attention is simply a dot product between inputs from the same distribution. In the attention mechanism, the important part is to use a decoder system to use the learned

weights for the input sequences for greater emphasis. The decoder can be a recurrent network for language translation or a CNN for visual tasks. Attention mechanisms emphasize important correlations in data distributions. For multimodal modeling tasks, the attention mechanism can aid in feature alignment which further aids in the fusion of multimodal data. Applications related to multimodal interaction tasks include visual question answering, image captioning, or visual entailment. Mascharka et al. [91] introduce a neural module that models the attention mechanism that decreases the gap between explainability and performance for visual reasoning tasks. The attention mechanism for multimodal modeling increases the interpretability of the models due to the complementary information on the domain, but representation and interaction can be challenging for this task.

2.2.2 Jointly-Training

Training an additional model for explainability along with the model for the primary tasks such as detection, classification, segmentation, etc is also studied in the literature. This additional model for explaining can provide the explanation in various forms, such as concept-based explanation, text explanations, the association between latent and input features, etc. The additional model for explanation is jointly trained with the actual model (model used for primary tasks). This method can be easily understood using the example of image captioning as given in Equation 2.17. In Equation 2.17, the first loss term $\mathcal{L}(y_n, y')$ corresponds to the prediction loss and $\mathcal{L}(e_n, e')$ corresponds to the loss of the explanation component.

$$\arg\min_{\theta} \frac{1}{N} \sum_{n=1}^{N} \alpha \mathcal{L}(y_n, y') + \mathcal{L}(e_n, e') \tag{2.17}$$

Some explanations from joint training are presented as text instead of statistics which is understandable to humans as it is natural language. Similarly, another explanation method is the use of the association of explanation, as mentioned earlier, uses association input and latent features. The association can be between input features and semantic concepts or between model prediction and a set of input features. The advantage of this type of method lies in the fact that semantically meaningful concepts can be represented as relational graphs, heatmaps, etc. The high-level concepts are associated with the internal model representation. The challenge with this method is the access to the internal representation of the model. Another disadvantage of this method is the high computational costs, which causes a bottleneck of this method.

2.3 Transformer Based Interpretation

The state of the art for transformer-based interpretation methods is not very widely discussed in the literature. Therefore, we have presented it in the dedicated Chapter 5 with our own proposed solution.

Chapter 3

Hybrid Deep Neural Networks for Risk Detection

3.1 Introduction

For the first time in history, most people worldwide can expect to live into their sixties and beyond. Between 2017 and 2050, the number of people aged 60 and older is expected to double, to reach 2.1 billion, representing more than 21% of the world's population, and up to 40% in some European and Asian countries. The quality of life of the elderly continues to improve through an important investment by the scientific community to cope with the aging of the world population [100], especially in developed countries. In spite of this investment, the number of so-called "frail" people continues to increase.

For this reason, promoting healthy living in place has progressively become a major challenge for all societies worldwide. In this context, "SmartHealth" technologies

are clearly a promising level of action to improve the living environments of the elderly population.

To meet the current demands of the elderly, particularly the frail elderly, for independent living, monitoring systems have to include the detection of risks and situations encountered by the frail elderly. The development of smartphones and other wearable devices such as smartwatches etc helped in obtaining the recordings from accelerometers, gyrometers, and other vital signals such as heart rate, blood pressure, etc. Several datasets are publicly available such as OPPORTUNITY[24], WISDM[66] and UCI-HAR[4]. An effective monitoring system should not be limited to a device with a limited range of daily living situations but should be extensible and include a wide range of situations that may be encountered by a person living at home. The integration of e-health sensors in portable devices has increased the potential for full-time monitoring applications.

Multimedia technologies are deployed today on the multimodal data collected by wearable sensors. The adventure of Deep Neural Networks (DNNs) as powerful classifiers and the recent multi-stream DNN architectures allow us to handle the heterogeneous data in the tasks of monitoring frail subjects [94]. Data fusion which is necessary as both the context of the person and his/her physiological and motor status have to be considered can be efficiently realized with such architectures.

However, there are several constraints associated with each sensor technology, such as missing data issues related to wireless technologies, latency in response times, etc., and difficulties in synchronization for real-time monitoring. The major challenge for

our problem is the pre-processing of the dataset. In the recording scenario defined by psychologists sixteen sensors are used, some of them are wired, others use blue tooth protocol and WiFi [141]. While recording a risky situation there can be a failure for the sensors in data transmission. Furthermore, events to detect in the monitoring process, such as risky situations are rare compared with the overall volume of the data which might be collected in a daily recording lasting for several hours. This is why data pre-processing for the DNN training becomes a complex task. Compared to another previous work, [143], we have developed new architectures for the detection of risk situations. In particular, we propose a hybrid architecture allowing to simultaneously include sensor signals and video data. This data collection protocol has been established by Dr. Thinhinane Yebda and Dr. Marion Pech

In this chapter, we report multidisciplinary research on the prevention of risk situations of frail people. Not only do we have to design an efficient risk detection system on multimodal data, but also to identify real-life situations that can be considered risky, that is, to design the risk taxonomy, and to define a data collection protocol in the most ergonomic manner for frail people. This chapter proposes an end-to-end network for the detection of risk situations from a novel health dataset comprising both visual and unsynchronized multi-variate time series data. The contributions of the chapter, are the following:

- An adapted GRU architecture with attention blocks to perform detection of risk situations in a pre-recorded in-the-wild dataset.

- A two-stream hybrid 3DCNN-GRU architecture on an unsynchronized health

monitoring data comprising visual and signal modalities.

- An adapted two-stream architecture using self-attention models as backbone instead of CNN and RNN-based architecture.

- We build on our previous work [142] to precise and develop the risk taxonomy.

- Data collection protocols for individual monitoring of frail persons are developed.

The chapter is organized as follows: Section 3.2 presents the state of the art, making the focus on the healthcare aspect of the present research, the use of IoT for the frail and elderly, and data analysis. In Section 3.3, our scenario and data collection protocols are presented and the taxonomy of risk situations is defined. Section 3.4 describes the proposed hybrid two-stream architecture of the DNN classifier for the detection of risk situations. In Section 3.5 experiments and results are reported. Finally, conclusions and future work are discussed in Section 3.6.

3.2 State-of-the-Art

In this state-of-the-art, we will initially focus on the healthcare aspect, then Internet of Things (IoT) technologies for assisted aging will be reviewed. Finally, the focus will be on sensing data analysis with Deep Neural Networks.

3.2.1 Healthcare Aspects

When developing systems for healthcare and monitoring, clinical and specific aspects have to be taken into account. These include conditions typical of health care and care of the elderly in general. The first focus category includes dementia in all its forms and stages, Alzheimer's disease is a special case of severe dementia, Parkinson's disease, "frailty and falls", or chronic diseases in general. Elder care refers to the care of elderly people who do not have a specific disease, but rather need monitoring and maintenance of an active and healthy lifestyle in old age. This can be achieved by non-intrusive technologies and is known as "ambient assisted living" (AAL). A wide range of risk situations encountered by frail people have been identified in [121], These studies were used in [142] to select a necessary set of sensors and design data collection scenarios. The most common accidents among the elderly are falls. Furthermore, the risks faced by older adults differ according to their medical history. For example, for Parkinson's patients, the most urgent risks are Parkinson's falls, for Alzheimer's patients, there are stressful situations, loss of orientation and loss of direction. For people with diabetes, hyperglycemia and hypoglycemia are dangerous. For the elderly without a particular disease, risk situations often rely on the context and their physiological conditions, e.g. a person in a hypo-tonic status could stumble in a bathroom or in a kitchen and thus has a *risk of fall*, or being in a stressed condition and cooking he/she could forget fire on the cooker, that is has a *domestic accident risk*. In the follow-up we shortly review IoT technology, which allows for human sensing in monitoring and risk prevention.

3.2.2 IoT for Elderly

In the broad range of Internet of Things (IoT) technologies wearables dominate the literature due to their growing popularity and affordability. In [115], the authors selected the most relevant devices in an AAL context, including wearables that contribute to the well-being of the elderly.

They studied different types of them such as headbands, sociometric badges, camera clips, smart watches, and sensors integrated into clothing, Biometric sensors are a special type of portable or non-portable devices that are used for continuous and on-demand measurement of physiological and medical data. In the field of healthcare, they are used e.g., for measuring body temperature, electrocardiogram (ECG), pulse oxygen saturation, blood pressure, blood sugar, etc. Smart home devices are usually ambient and unobtrusive in an AAL context. A study [134] reviews indoor positioning systems, emphasizing on human activity recognition, as well as biometric sensors (vital sign monitoring, blood pressure, and glucose).

The data recorded by these sensors represent time series and have to be analyzed either online or offline (lifelog) to detect situations of potential risk in the everyday life of elderly and frail persons. Hence rises the classification problem for risk situations detection. The most powerful classifiers/predictors nowadays are Deep Neural Networks. In the following, we will review some of them.

3.2.3 Monitoring Data Analysis with Deep Neural Networks

The ultimate goal of IoT systems for frail subjects monitoring consists of real-time decision-making for risk detection and prevention. The general trend for such a

detection consists in the development of supervised machine learning approaches particularly Deep Neural Networks (DNN)[17].

Such classifiers are heavy to train but are quite light in decision making as required matrix operations which are executed in parallel and can be implemented on mobile devices. Nevertheless, before the designed architecture may be made lighter and transferred to a wearable device, e.g., by network quantization, it is necessary to design an efficient DNN solution yielding a high accuracy in the classification of multimodal signals from the wearable. The sensor data are time series that can be efficiently processed by Recurrent Neural Networks (RNNs). RNNs have a "memory" that captures information about what has been computed so far and performs the same task for each element of a sequence. RNN can not only learn local and long-term temporal dependencies of data but can also accommodate variable-length input sequences. Although RNNs are a simple and powerful algorithm, it suffers from gradient vanishing. To combat these problems, the particular design of RNN was proposed by S. Chochreiter et al. in 1997, namely long-term and short-term memory (LSTM) architectures [57]. Currently, RNNs such as LSTM and Gated Recurrent Units (GRU) are used in healthcare research. Friedrich et al. [45] present Inertial-Measurement-Units (IMU) and LSTM to predict mobility assessment score that gives valuable information to physicians to diagnose changes in mobility and physical performance. In visual data processing, Convolutional Neural Networks (CNNs)[68] are the most popular. Many studies applied CNNs to detect risky situations especially

falls [22]. For the analysis of temporal video information, the so-called 3D ConvNets[59] are suitable as they process chunks of successive video frames to take into account temporal coherency and singularities for video classification. Some of them are designed as two-stream architectures to take into account [90] both appearance (pixel data) and motion data (optical flow).

The concept of two-stream architecture is suitable for our task of classification of risk situations, as the complex multimodal data contains both time-series signal data from sensors and video data. To present the classification problem, we first describe the risk detection scenario and taxonomy.

3.3 Risk Situations for Frail Persons

In this section, we will first describe the scenario of detection of risk situations we are working on. Then the taxonomy of the considered risk situations will be presented.

3.3.1 Scenario and Taxonomy of Semantic Risk Situations

In the monitoring scenario, a frail person remains in his/her home environment. He/she is monitored only indoors. We are interested in "semantic risk situations". It represents a combination of both: the person's motor and physiological status and his/her contextual environment. In our scenario, the context is observed with the help of the wearable camera and the state of the person - with physiological and dynamic sensors. Such a recording process has been already exhaustively studied in literature[60]. The proposed taxonomy of risk situations comprises "immediate"

risks and long-term risks.

Immediate Risks Immediate risks are those risks that can have an immediate impact on the lives of frail subjects:

- *Environmental Risk of fall*: Concerning environmental risk situations, the potential association between household hazards and older adults falls is well explored in the literature [83]. Walking on a slippery floor, climbing stairs, and stumbling on obstacles, for example, are some of the environment-related actions encountered by older adults that may increase their risk of falls.

- *Physiological Risk of fall*: Age-related physiological changes also increase the risk of falls among older adults as they correspond to potential changes in physiological symptoms such as heart rate, acceleration, meta-acceleration, etc. [92]. Sit to stand is an example of a daily living activity risk scenario in which older adults can be exposed to acute fall risk [101].

- *Risk of Domestic Accident*: Domestic accidents are defined as daily activities such as cooking, using knives, handling of "dangerous" utensils, and ironing that can potentially be associated with burns or increased risk of sustaining injuries.

- *Risk of Intrusion*: The risk is defined as a situation when the subject is near a door and another person is present.

Long-Term Risks are described as situations with less immediate impact on individuals and include:

- *Risk of Dehydration*: The individual does not drink (water, tea, or coffee) during the daily monitoring period; the detection problem consists on the contrary in drinking action detection.
- *Risk of Medication Intake*: It is also a risk when a person under medication forgets to take medicine. As in the risk of dehydration, the detection problem consists in drug intake detection.

The risk of falling resembles all risks that can arise from actions carried out by frail people on a daily basis. For the risk of domestic accidents, we consider all risks that can be linked to the activities that frail people carry out specifically in the kitchen. Therefore we assimilate the risk of domestic accidents to the possibility that a frail person is in his/her kitchen.

We note, that a frail person is in a risky situation when the "dangerous" context is combined with his/her unusual status, e.g., a frail person is not each time in a risky situation when he/she enters the kitchen or goes upstairs in the house. This is why contextual (visual) and multi-modal (physiological) sensing is necessary. Furthermore, we intentionally exclude the "fall" detection from our taxonomy as it is not a risk, but already a dangerous event and because of the large availability of fall detection products already on the market. Finally, in this paper we will present our results on four classes: a)*No Risk*, b) *Risk of falling*, c) *Risk of domestic accident*, d) *Drinking of water*. The project is continuing and other risk situations will be further

recorded. We also stress that we work under *one-subject scenario*. Risk detection has to be adapted to the environment of each frail subject.

3.3.2 Data Collection Protocol

Healthy volunteers were recording data on a wearable kit simulating risk situations. They were also asked to write a short recording dairy approximately indicating the time instant when the subject puts the devices on himself, adjusts them, records, and simulates various situations where each risk or situation from our taxonomy is reported e.g., 13:45 entering into the kitchen (*risk of domestic accident*).
The wearable kit consists in devices connected to the developed Android application.It comprises the following sensors

- A bracelet Empatica E4, which is a medically-graded wearable device. It is equipped with an accelerometer, an Electrodermal Activity (EDA) sensor, a PPG (Photoplethysmogram) sensor, an Infrared Thermopile Sensor, and a Bluetooth Low Energy transmitter.
- A chest-worn wearable device MetaMotionR. The following measures are extracted: acceleration, angular velocity, and magnetic field.
- A wearable camera which is positioned on the shoulder. The latter is directly connected to a phone considered as the main controller of the whole device. The camera records 3 seconds of video every 10 seconds with 10 fps frame rate.

The device was worn for thirty days by two healthy volunteers: One young adult twenty-six years old, wore it for 21 days and a sixty-two years old volunteer wore the

device for 9 days.

Psycho-gerontologists consider that a frail person may be in a risky situation when he/she does not e.g. simply enter the kitchen, but when he/she is disturbed, stressed, etc. Thus we asked our volunteers to simulate such situations by associating emotions (fear, stress, etc.) with their actions.

Recorded dataset BIRDS- Bio-Immersive Risk Management System - contains recordings of 30 days of the overall volume of 6316 min. The recording time per day varies from nearly two hours up to five hours. The duration of risk situations represents only a few percent of the whole. The corpus BIRDS will be made publicly available subject to fulfilling legal procedures according to GDPR (https://gdpr-info.eu/). We have simulated the recording process with healthy volunteers. For frail elderly special permission is required with insurance. This is a part of future work.

3.4 Two-Stream Neural Network for Recognition of Semantic Risk Situations

We consider sensor data and visual data as two modalities and design a two-stream architecture to solve the multi-class classification task of semantic risk detection.

3.4.1 Two-Stream Network Architecture

The proposed architecture is illustrated in Figure 3.1. The upper branch is a 3D convolutional network, built on ResNet backbone. We use this model as it proved

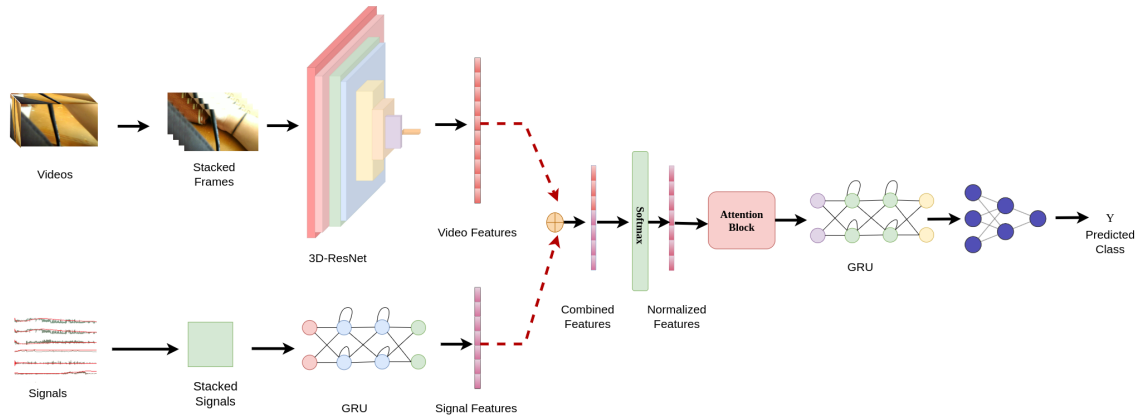

Figure 3.1: Two-Stream Network for the multimodal data constituting of the videos and multivariate time series signals.

to be efficient when recognizing actions in the video in our previous work [90] and shortly describe it in Section 3.4.3. The bottom branch is a recurrent neural network GRU which is precisely an encoder network described in Section 3.4.4. The fusion of modalities is performed by an intermediate fusion approach. Two streams of our network are trained end-to-end using synchronized training data from video and multi-modal sensors.

Figure 3.1 illustrates the concatenation of normalized features from the two branches with a red cross. The concatenated features are submitted to a decoder network, which is a GRU network with an attention block. The class assignment is realized with the usual fully connected and softmax layers.

The data is split into 80%, and 20% as training and test data. We describe data pre-processing in Section 3.4.2.

3.4.2 Data Pre-Processing

The real-world data collected in the wild require cleansing.

Visual Data: Visual data is collected from a wearable camera, see Section 3.3.2 requires data preprocessing as similar to sensor data the risk situations constitute only a few percent compared to the whole volume of visual data. Thus, scrapping of non-risky situations is a necessity for balancing the dataset to avoid the high bias of the deep network. The frame extraction is performed. The frame aggregation is then realized using a sliding window approach prior to feeding the data to the network. It facilitates the preservation of the temporality of the data. The temporal window size N_v is fixed to 10 frames according to our preliminary experience. The frame rate in recorded videos is not constant, it varies between 12 - 25 fps, due to the various connection delays on the Android platform. Hence, the window of length N_v can correspond to a different duration in time. Raw RGB pixel data are used.

Sensor Signal Data: Data pre-processing is composed of two steps, i.e data imputation and data normalization. Data is recorded with a number of sensors, which may malfunction at arbitrary time moments or loose their connection via WiFi and Bluetooth, which results in missing values from the data requiring imputation of data. We use a classical data imputation by mean value, computed on the available raw data of each sensor on the whole dataset.

In the following, we denote x_i the the vector of dimension S whose entries are the observed values of the S sensors at time i and x_{mean} the vector of means of all available values of sensors. We also denote by x the set of variables $\{x_1, x_2, \cdots, x_D\}$. This is our whole set of measures in the whole corpus. Supposing the Independence of the coordinates of x the covariance matrix Σ of x will be diagonal. The normalization

we propose consists for the whitening of the data, see Equation (3.1) in a vector form.

$$\zeta(x_i) = \frac{x_i - x_{mean}}{\sqrt{\Sigma(x)}}. \tag{3.1}$$

Then we linearly scale the whole set of whitened measures $\zeta(x_i)$ to fit the interval $[0, 1]$ for each coordinate of x_i, and obtain the normalized data x_{norm_i}.

$$x_{norm_i} = \frac{\zeta(x_i) - \min(\zeta(x_i))}{\max(\zeta(x_i)) - \min(\zeta(x_i))}. \tag{3.2}$$

Data Synchronization: Data synchronization is introduced to facilitate the end-to-end network architecture as well as encompass the challenge of variable sampling rate between the video recording device and the devices recording the multivariate time data. The visual data is sampled w.r.t multi-variate sensor data using a key i.e for this data, it is, namely, the timestamp, thus the timestamps recorded by sensors are matched with the videos and the corresponding frames are generated for the particular timestamp. So, for a particular video, the corresponding sensor data is established. The synchronization is performed on a frame level.

3.4.3 3D ResNet Encoder for Visual Data

In the intermediate fusion paradigm, the processing of visual data consists in encoding them via the "visual" network which is a 3D convolutional network, we built on ResNet-26 backbone with 3D convolutions. The use of 3D CNN helps us to capture the temporal information of the videos. Upon applying the 3D convolution, the value

at the particular position x, y, z at the j^{th} feature map in the i^{th} layer is given by Equation (3.3).

$$v_{ij}^{xyz} = \tanh\left(b_{ij} + \sum_{m}\sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1}\sum_{r=0}^{R_i-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)}\right) \tag{3.3}$$

In Equation (3.3), the 3D kernel is of size R_i along the temporal dimension and w_{ijm}^{pqr} is the $(p, q, r)^{th}$ value of kernel connected with the m^{th} feature map of the previous layer The visual encoding is performed by using the RGB videos pre-processed as mentioned in Section 3.4.2 for *visual data*. Features are extracted using the ResNet-26 architecture for feeding to the decoder network. 3D CNNs i.e. containing 3D convolution kernels for spatio-temporal data that have been used quite extensively for activity recognition tasks [59]. The features from the ResNet-26 are extracted from the penultimate layer i.e. just before the softmax layer. The 3D ResNet used here is the basic block architecture which constitutes two convolution layers followed by a batch normalization layer and ReLU for each of the individual layers mentioned. As mentioned in the architecture of ResNet the use of the bypass, we use the shortcut pass to connect the top of the block with the last ReLU layer of the block.

3.4.4 GRU Encoder-Decoder with Attention Layer for Sensor Data and Two-Stream Fusion

In this section, we present the GRU network with an attention layer. Its encoder part is used to process the multimodal time series data from sensors. The decoder part in this architecture comprises the attention layer that is used for intermediate

fusion in our two-stream network.

The encoder network maps the normalized sensor data to a sequence of representation features. The input sequence is taken as x_{norm_i} within a temporal window of the length N_s. An input sample at time t is $x_t \in \mathbb{R}^{S \times N}$. The equations describing the GRU encoder are given below:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{3.4}$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{3.5}$$

$$\tilde{h}_t = \tanh(W x_t + U(r_t \odot h_{t-1}) + b) \tag{3.6}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{3.7}$$

Here r_t ,z_t, h_t refer to reset gate, update gate, and hidden state, respectively. W_z, W_t, W, U_z, U_t, U, b_z, b_r, b are model parameters, and σ denotes the sigmoid function.

The decoder network contains the attention block followed by two stacked layers of recurrent neural networks (in our case a GRU) and finally a fully connected layer. The decoder network is initialized with the same state as that of the encoder network. The class state is predicted by the final fully connected layer. Figure 3.1 (right) gives the overview of the decoder network.

Attention Block in the Decoder: It stresses the features and dependencies leading to better predictions. We pass the concatenated features from the video stream and from the signals to the decoder network for classification. The equations below describe the attention block for the network.

The multidimensional output from the sensor stream is $h_t \in \mathbb{R}^{M \times N_s}$ with M the number of features on the hidden state in the GRU block. The output h_t of the GRU-based encoder is then fed to the attention block. For the sake of clarity, we will further denote the output of the encoder as o_t. The output features from the video stream i.e. the ResNet-26 features are denoted by $res_t \in \mathbb{R}^{C \times F}$ with C being the number of channels and F as a number of features which depend on the size of the input image which for Res-Net is 224×224. The architecture used here is the basic block and bottleneck, and with our preliminary experience, the basic block architecture is used [54]. The features obtained from both the streams are flattened and concatenated as given in Equation (3.8).

$$c_t = res_t \oplus o_t \tag{3.8}$$

$$\tilde{c}_t = \text{SoftMax}(c_t) \tag{3.9}$$

For the calculation of the attention weights the combined normalized input features are concatenated with the initial hidden state of the decoder as the features while training are updated with respect to the decoder output and its hidden state. Equations (3.10), (3.11), and (3.12) show the calculation of attention weights.

$$\acute{c} = \tilde{c}_t \oplus h_0 \tag{3.10}$$

$$a_t = \acute{c}_t.W_t^T \tag{3.11}$$

The output of the layer is passed to the softmax for normalization as in Equation (3.12)

$$\tilde{a}_t = \text{SoftMax}(a_t) \tag{3.12}$$

$$g_t = \tilde{a}_t.c_t^T, \tag{3.13}$$

To push the output c_t to the decoder we first initialize the decoder GRU with the same initial state h_0 as for the encoder GRU for the sake of reproducibility of results. Then the outputs c_t are processed and passed to the decoder, the first layer of it is the attention block, which is a linear layer, see Equation (3.11). The attention is given in Figure (3.2).

The result g_t is submitted to the GRU classifier - decoder, see Figure 3.1 lower part. We also designed this GRU classifier as a two-layered network as the addition of the second layer improved the performance accordingly to our preliminary experiences. Now let us consider the self-attention models for the purpose of the two-stream network:

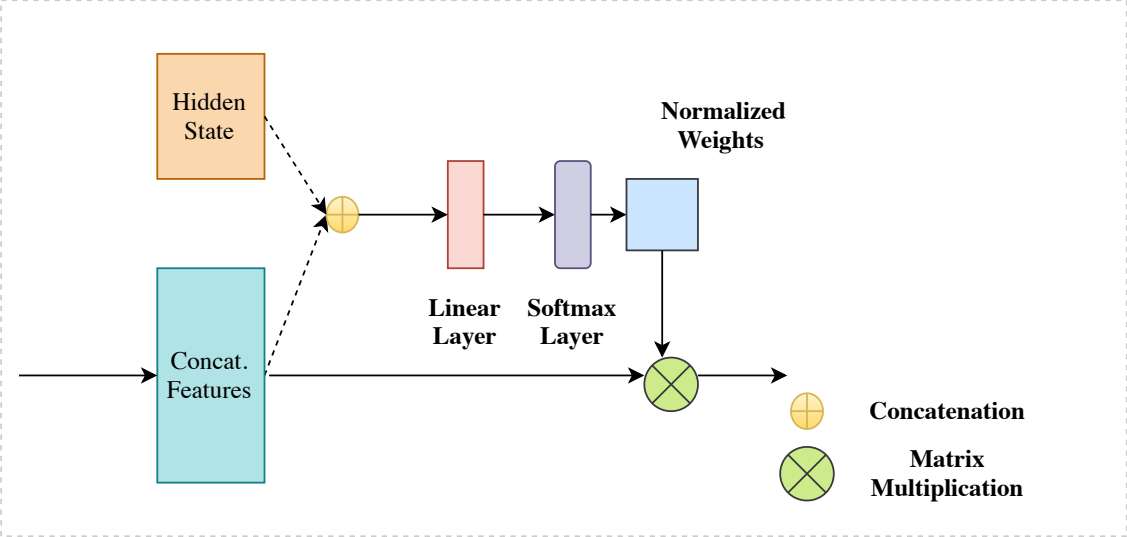

Figure 3.2: Design of the attention block.

3.4.5 3D Bottleneck Transformer for Videos

3D Bottleneck transformer has been developed for the visual data which are in the form of videos inspired by [114]. In the original work [114] the bottleneck transformer is based on ResNet-50 network [62]. The idea is to leverage the benefit of self-attention in the bottleneck blocks of the ResNet. For this purpose, we have inflated the 2D self-attention on the original network to 3D self-attention to accommodate the temporal information. The self-attention is presented in the last stack of the ResNet-50 architecture. The reason to use the self-attention in the last block is to use the lowest resolution of the feature maps. In this transformer, we have changed the 2D convolutions to 3D convolutions to process chunks of video frames instead of images. A global multihead self-attention is applicable to the 3D feature maps. We applied the multi-head self-attention on the last three layers of the ResNet-50 instead of spatial convolution. The attention of a particular i^{th} head in a transformer

is expressed by Equation (3.14):

$$head_i = \text{SoftMax}\Big[\frac{QW_i^Q(KW_i^K)^T}{\sqrt{d_k}}\Big]VW_i^V \tag{3.14}$$

Here Q, K, V are respectively query, key and value matrices and W_i^Q, W_i^K, W_i^V are their weight matrices and d_k is the dimension of the input key vectors. As Q,K,V the same feature tensor is obtained from an initial layer of ResNet-50 exactly as in [114], but computed with 3D convolutions.

The multi-head attention in the transformer is expressed by Equation 3.15

$$\text{MultiHead}(Q, K, V) = \text{Concat}(head_1, head_2, ..., head_h)W^0 \tag{3.15}$$

Here W^0 are learnable parameters multiplying the attention matrix after concatenation of attentions of each head h is the number of heads. The transformer weight matrices are trained with Adam optimizer[63] as in [114]. The feedforward network of the transformer is multi-layer perceptron with one hidden layer. The output feature dimension of the transformer for video is $d_v = 64$.

3.4.6 Linear Transformer for Sensor Data

For sequences of vectors of sensor signals we used linear transformer proposed by Wang et al. [113] for the sake of computational efficiency. Here Key (K) and Value (V) matrices are of dimension $n \times d$ as composed of n vectors of sensor signals of dimension d. They proposed a self-attention mechanism that is based on context

mapping in linear time and complexity w.r.t sequence length. The idea is to add two linear projection matrices after the computation of key and value matrices represented as E and F in Equation (3.16), thus reducing the dimension. The authors project the original $(n \times d)$ dimensional key K and value V matrices to $(k \times d)$ matrices with projection matrices $E_{k\times n}$, $F_{k\times n}$, where the target dimension k satisfies $k << n$. Thus in the computation of attention for one head, instead of using $n \times d$ matrices for a general transformer, as given in Equation (3.14), $k \times d$ dimensional matrices are used to reduce the computational cost. The final attention matrix $\overline{head_i}$ in Equation (3.16) remains of the same dimension $n \times d$. In [113] the authors report similar performances of the linformer with projection on k-dimensional space, where k is twice lower than n of the original transformer without projection. Despite our sequences being of lower temporal dimension $n = 25$ than those considered in [113] as defined by risk detection application, in view of the complexity reduction in training and the future real-time generalization performance, it is still interesting to reduce the computational complexity.

$$\overline{head_i} = \text{softmax}\Big[\frac{QW_i^Q(E_iKW_i^K)^T}{\sqrt{d_k}}\Big]F_iVW_i^V \tag{3.16}$$

The multi-head self-attention is computed as in other transformers by concatenation, see Equation 3.15.

Once Multi-Head Self Attention is obtained, we pass it through a feed-forward network comprised of multi-layer perceptron with a single hidden layer. The output

feature dimension is $d_s = 64$

3.5 Experiments and Results

3.5.1 Datasets

In this work, we have applied the proposed method to two datasets: the first one is open sensor dataset UCI-HAR[4] for action recognition. The second one is our recorded BIRDS dataset.

3.5.1.1 UCI-HAR Dataset

UCI-HAR is a dataset for Activities of Daily Life (ADL). The dataset was recorded by 30 volunteers on a Samsung Galaxy SII smartphone during two trials. In the first trial the smartphone was placed on the left side of the belt and for the second trial, after a period of 5 seconds, the smartphone was placed according to the preference of the user. A visual inference was used for the annotation of the ground truth. A total of 9 parameters were recorded at a sampling rate of 50 Hz using a tri-axial linear acceleration and angular velocity with the phone accelerometer and gyroscope. For the reduction of noise, a median filter, as well as a 3rd order low-pass butter filter, were used. Mapping of signals to the frequency domain is through *Fast Fourier Transform* and sampled by a fixed width sliding window, i.e., 128 readings/window (2.56 seconds and 50% overlap). This dataset contains a total of 10299 instances. The taxonomy comprises 6 activities: *Walking, Walking Upstairs, Walking Downstairs, Sitting, Standing, Laying Down.*

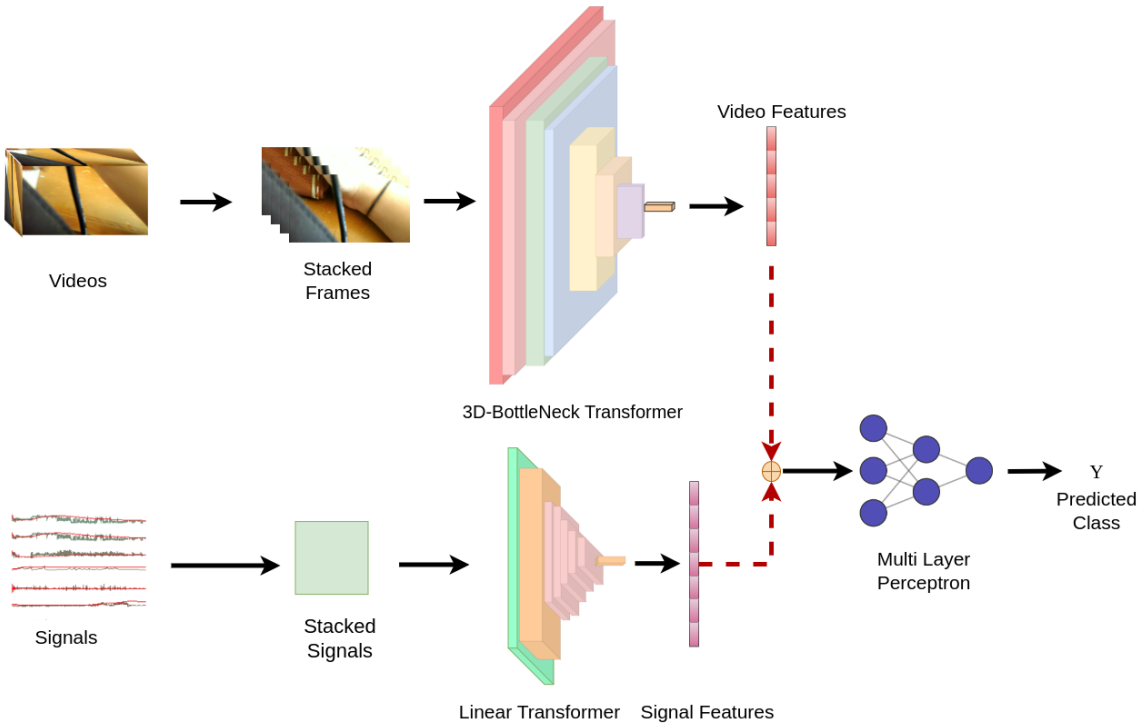

Figure 3.3: Hybrid Transformer Network for the multimodal data consisting of the visual data as videos and multivariate time series signals.

For the creation of training and test datasets, a total of 70% and 30% volunteers are used respectively. Table 3.1 shows data distribution for the training and test sets for each class. The dataset does not contain video data. The combined architecture is presented in the Figure 3.3

3.5.1.2 BIRDS Dataset

Our dataset BIRDS have been recorded according to the protocol described in Section 3.3.2. The overall number of sensors is 16, which defines the dimension of the data vector.

In the experiments, we use recordings from one frail adult volunteer of 12 days

Activities	Train Instance	Test Instance
Walking	1226	496
Walking Upstairs	1073	471
Walking Downstairs	986	420
Sitting	1286	491
Standing	1374	532
Laying Down	1407	537

Table 3.1: Distribution of Human Activity Classes on UCI-HAR Dataset. [4]

duration, because of privacy constraints on the data. Three situations appear on this span of 12 days.

They are: *Environment Risk of Falling, Physiological Risk of Falling, Risk of Domestic Accident* and an "opposite-to-risk" situation, such as *Risk Associated to Dehydration* and *Risk due to Medication Intake*. The rejection class is labeled as *No Risk* situations. Table 3.2 contains the distribution of samples for the classes of this taxonomy in the dataset. In the second column, the number of signal samples is indicated, and in the third column - the number of recorded video files.

Risk Situations	Sensor Data		Video Data	
	Samples	Percentage	Samples	Percentage
No Risk	20176233	95.03%	18207	94.45%
Environmental Risk of Fall	312425	1.47%	248	1.30%
Physiological Risk of Fall	180307	0.85%	156	0.81%
Risk of Domestic Accident	333616	1.57%	477	2.47%
Risk associated with Medication Intake	145992	0.69%	114	0.60%
Risk associated to Dehydration	81624	0.38%	62	0.32%

Table 3.2: Distribution of risk situations on the raw sensor and video data for BIRDS dataset.

As stated earlier, no risk samples constitute most of the data. Hence, the dataset is strongly imbalanced and cannot serve for DNN (Deep Neural Networks) training as it is. To balance it, we have retained only 0.85% proportion of *No risk* samples from the whole sensor dataset. Additionally to the data normalization and scaling, we, see Section 3.4.2, adopted a sliding window approach at a pre-processing step for the creation of instances for the sensor data. A fixed length window of 20 samples was considered with a sliding stride of 4 samples. Note that the time intervals between different samples are not stable because of the sensors' imprecision. Hence, the fixed 20 sample window can have a different duration in time (in the limit of a few milliseconds). For assigning a unique label to a window, we used the maximal mode of the ground truth label histogram computed on it.

We note, that the BIRDS dataset is much more challenging than the UCI-HAR dataset [4]. The latter is regularly sampled and less prone to noise.

3.5.2 Risk Classification with a Two-Stream Hybrid Neural Network

In this section, benchmarking and experiments of the proposed hybrid two-stream network are presented.

As discussed in Section 3.4 we are using two streams, one for feature extraction from videos and the second stream for feature extraction from the sensors. The sensor signals have to be processed by our GRU with an autoencoder network. Hence, we first bench-marked our approach only on the signals using the lower branch of the

two-stream architecture, see Figure 3.1. Then the best model was used in the two-stream architecture and finally, we compared our two-stream hybrid network in risk detection problem with the 3D CNN classifier applied to the video data only.

3.5.2.1 Benchmarking on Sensor signals

Our first series of experiments was conducted on the UCI-HAR dataset [4] as it is more regular and balanced than the BIRDS dataset.

We used three networks: LSTM [57], GRU[30], Autoencoder (GRU)[28], LinFormer [113] and our GRU Autoencoder with Attention. The optimization method was Adam [63] with a fixed learning rate of 0.0001 and a batch size of 128 for UCI-HAR dataset. We have varied the number of epochs for each network as presented in Table 3.3. From this Table, we conclude that the proposed GRU Autoencoder with Attention block gives the best result for a reasonable number of epochs (20).

The second series of experiments was conducted on our challenging BIRDS dataset. Here we first optimized the learning rate and batch size hyper-parameters by bisection method and got a learning rate of 0.00001 and batch size of 256 on LSTM. The same parameters were used for other networks: GRU, Autoencoder GRU, and GRU with Attention, see Table 3.4.

The training was done using the open-source PyTorch Framework. Training has been conducted in a single server with 2× Tesla P100-PCIE-16GB.

The reported test accuracy score in Table 3.4 is lower than on the UCI-HAR dataset, but still, the GRU with attention performs the best.

Algorithms	Test Acc. %	Epoch
LSTM [57]	81.38	20
LSTM [57]	92.36	200
GRU [30]	87.03	20
GRU [30]	89.65	30
Autoencoder(GRU)	85.34	20
Autoencoder(GRU)	90.46	25
Autoencoder + Attn (GRU)	**92.28**	20
LinFormer [113]	**89.71**	20

Table 3.3: Comparative Study of various Algorithms on UCI-HAR Dataset.

Algorithms	Test Acc. %	Cross-Val. ($\mu \pm \sigma$)
CNN[68]	48.43	0.728± 0.22
LSTM[57]	51.33	0.721 ± 0.43
GRU[30]	50.31	0.77 ± 0.50
Autoencoder(GRU), [28]	56.14	0.806 ± 0.59
GRU with Attention	**57.13**	0.832 ± 0.73
Linformer [113]	56.92	0.791 ± 0.67

Table 3.4: Accuracy and 10-fold average Cross-Validation Scores on the signal sensor data with balanced dataset. BIRDS dataset.

3.5.2.2 Results of Two-Stream Hybrid Network

At first, we present the results on the four classes instead of six classes as well and the data is a subset of the data comprising six classes in volume. As mentioned data acquisition is still an ongoing task, therefore the acquisition of new classes as well as data volume is progressive.

3.5.2.2.1 Ablation of the two-stream 3DCNN-GRU network: The results for this section are presented in **four classes instead of six**, *Environmental Risk of Fall,*

Algorithms	p-Values
CNN[68] vs Ours	2.6×10^{-4}
LSTM[57] vs Ours	6.8×10^{-6}
GRU[30] vs Ours	6.6×10^{-5}
Autoencoder[28] vs Ours	8.9×10^{-6}

Table 3.5: Paired t-test results (p-values) between Ours (GRU with Attention) and other methods

Risk of Domestic Accident, Risk associated to Hydration and the reject class *No Risk*

Balancing of data is an important success factor for training a well-generalizing model of a DNN. Our dataset *BIRDS* are highly imbalanced between *risk* and *no risk* situations. As we stated in Section 3.5.1.2 we retain only 0.85% of *No Risk* class for sensor signals accordingly in our preliminary experiments. For balancing video data, we have conducted a series of three experiments with different data scrapping percentages reducing the *No Risk* data sample proportion. The *No Risk* samples were randomly chosen from the original *No Risk* class population according to the fixed percentage. The percentages were chosen as 1.5% (*Experiment 1*), 0.5% (*Experiment 2*) and 0.75% (*Experiment 3*).We compare the performances of our two-stream network on balanced data.

The results are presented in Table 3.7. They are expressed as per-class Precision, Recall, and F-score. The results of *Experiment 1* show the precision of the class *Hydration By Water Intake* to be 0, which implies this particular class is not predicted. Reducing the sample size of *No Risk* situation to 0.5% of the total amount of the *No Risk* in the training dataset, which is equivalent to the total sample size of the class *Hydration by Water Intake*, the precision of *Hydration By Water Intake* becomes

almost 100% see "Experiment 2". Finally, in *Experiment 3*, we used a higher proportion 0.75% of the No Risk visual data, which provides us with the overall accuracy score of 80.21% with much less bias in the network for randomly split data. The experiments also show the importance of the *No Risk* situation while calculating the evaluation scores. Mean 10-fold cross-validation results on sensors yield us the best results as given in Table 3.4.

The cross-validation scores with respect to sensors have been obtained on the validation set. The cross-validation score provides a significant improvement in the mean accuracy scores from 60.55% to 83.20% as provided in Table 3.4 for our sensor data. This result also highlights the importance of the data when randomly split and while performing the cross-validation. Since the data has been taken on several days over a period of months thus it also signifies how features can change specifically for health data due to physiological and psychological changes.

The accuracy of the best model using the cross-validation is around **83.26%** which is an improvement from the previous model considering random splitting. It is evident from the results that there is almost an improvement of 20% in the accuracy score using cross-validation for the sensor data whereas only 3% improvement is observed while using the combination of visual and sensor data. Consequently, it is observed that visual features are dominant over the features which are obtained from the sensors.

Statistical hypothesis tests (paired t-test) were performed for the amount of data used in our experiments as presented in Table 3.5 validating our experiments. For

each comparison of our method with any method $\mathcal{M}$ listed in Table 3.5, we tested the hypothesis H_0: the accuracy of method $\mathcal{M}$ is equal to the accuracy of our method. The p-values computed between our model vs. the model $\mathcal{M}$ are much lower if we consider an α to be 0.05.

The values thus given in Table 3.5 are significantly lower than alpha thus we can reject all the hypotheses.

3.5.2.3 Results using Transformers for the Two-Stream Network

Here we apply the transformer network on both the video and sensor streams of the BIRDS dataset. The results are shown in Table 3.6 in terms of the f-scores, precision/accuracy, and recall per class. The *overall accuracy* of the model is 72.19%. The cross-validation provides a slight improvement of the overall accuracy of 71.09% (mean on 10-fold cross-validation). This is a slight improvement when compared to just the video stream giving an accuracy of 70.47% (mean on 5-fold cross-validation) or only the sensor stream which achieves an accuracy of 35.55% (with the same scheme of mean on 10-fold cross-validation).

Risk Situations	**Precision**	**Recall**	**F-Scores**
No-Risk	0.83	0.86	0.80
Environmental Risk of Fall	0.75	0.82	0.69
Physiological Risk of Fall	0.27	0.20	0.40
Risk of Domestic Accident	0.73	0.72	0.74
Risk Associated with Medication Intake	0.52	0.59	0.47
Risk Associated to Dehydration	0	0	0

Table 3.6: Evaluation Metrics in the BIRDS dataset

From Table 3.6 the metrics for ***Risk Associated to Dehydration*** **are** 0 which is a drawback for the network. It can be argued that features from 'opposite-risk' situations are very similar to *No-Risk* thus it may have been classified as *No-Risk* class.

3.5.2.4 Comparison w.r.t. Risk Detection on Video only

Hence in this experiment, we compare our hybrid two-stream network performance on the classification of risk situations on video only with 3D Res-Net-26 and 3D-BotNet. The video accuracies give 65.67% and 70.47% respectively. The accuracy of video classification is 65.67% for only video streams compared to 73.86% for our two-stream framework using the 3DCNN-GRU framework while using a self-attention-based model the combined accuracy is 72.19%.

3.6 Conclusion

In our chapter, we have proposed a novel two-stream method hybrid network 3DCNN-GRU as well as self-attention-based models, for the classification of risk situations on temporal data including sensors and video. The 3DCNN-GRU method comprises the extraction of features from the two networks, the fusion of the features followed by an introduction of an attention block, with training performed in an end-to-end manner. The "video stream" is processed by a 3D Res-Net with 3D convolutions. The other stream GRU encoder is used to encode the features from the multi-modal sensor data. For the self-attention based model the video features are extracted

using a self-attention based model named 3D BottleNeck transformer and signal features are obtained using a linear transformer. The combined features are then fused and normalized followed by an MLP. This network is also trained in an end-to-end manner.

Further, this chapter introduces a novel dataset along with the visual data with a number of challenges such as missing data and non-synchronization of data between the videos and sensors. This framework is unique as the video classification of risk situations is performed with complementary features from the multi-modal sensor data.

The hybrid architecture showed quite a large increase in performance compared to the semantic risk classification on the sensor data only. Working on the real-world "in-the-wild" dataset which is highly imbalanced, we have developed strategies and tools for handling such kind of data and will continue dataset recording and developing further, the classification network. This chapter also highlights the necessity for improving the performance for the risk-detection task in the self-attention-based model.

Table 3.7: Evaluation scores for various experimentation on BIRDS dataset. **Experiment 1**: Situation of *No Risk* is 1.5% of the total *No Risk* situations. **Experiment 2**: Situation of *No Risk* is 0.5% of the total *No Risk* situations **Experiment 3**: : Situation of *No Risk* is 0.75% of the total *No Risk* situations

Risk Situations	**Experiment 1**			**Experiment 2**			**Experiment 3**		
	F-Scores %	Precision %	Recall %	F-Scores %	Precision %	Recall %	F-Scores %	Precision %	Recall %
No Risk	86.05	81.36	91.32	23.06	30.00	18.75	48.78	58.82	41.67
Risk of Falling	48.64	99.98	32.14	77.77	70.00	87.50	83.72	85.71	81.82
Risk of Domestic Accident	54.20	49.15	60.41	89.11	85.14	93.47	87.38	79.50	97
Hydration by Water Intake	0	0	0	62.06	99.99	45	62.85	99.98	45.83

Chapter 4

Importance Based Pooling Transformer for Detection of Risk Events

In this chapter, we are focused on visual data, more specifically on videos. We adapt transformer models for the videos and try to optimize them. The target application is the detection of risks of frail persons in their home environment, but only using visual data. In contrast to the last chapter, we limit the complementary information from the signals. The proposal of Vision Transformer by Dosovitskiy et. al. [36] transformed image generalization using the transformers. The observed challenge for this method is the change of view as we are using ego-centric video data and not from a fixed singular viewpoint camera.

For interpretability, the idea is the localization of the features responsible for the model decisions. In this chapter, we seek to identify these important locations during

the training time. We are extending our work by deploying a complete transformer-based model instead of self-attention blocks on a convolution-based model. As discussed in previous chapters, transformers were initially proposed for sequential modeling and replace the traditional architectures such as RNNs and their variants such as LSTM [57] and GRU [29].

Our target application is the detection of risks of frail persons in their home environment, as was the case in our previous contributions. In Chapter 3 we have described our multimodal corpus BIRDS. Video data are most important in the recognition of the so-called "semantic risks" [144]. The latter refers to complex visual events and actions such as taking pills, drinking water, etc. Therefore, we resort to the latest models of transformers to analyze these video data. We note that recent transformer models allow for the design of systems to handle such data [94]. In this chapter, we are focused on the single modality of video, as we seek the best attainable accuracy in this modality. We remind you that our dataset BIRDS consists of challenging in-the-wild recorded data. Its annotation was fulfilled with visual inference and the diary recorded by the subjects during their monitoring. Hardware failures during recording or loss of connection yield missing data and noise. Therefore, a benchmark on publicly available "clean" datasets is necessary. We do it on the publicly available Kinetics-400 dataset [21].

To better meet the real-world accuracy requirements of automatic risk detection systems, we propose a video transformer architecture with a temporal pooling operation to handle noisy in-the-wild real-world data.

Transformer models have recently become a very popular tool for data analysis for various classification problems. Recently, transformers have been used for computer vision tasks such as video understanding [125], object detection [20], action recognition[16, 5], etc. The transformers can be seen as the integration of the self-attention module with the CNNs, as using the CNNs as a feature extractor or as a pure transformer without CNNs. Due to the lower inductive bias compared to CNNs, the use of transformers is based on a large amount of data. Video understanding and recognition have been long studied in the literature, with the use of LSTMs on top of convolutional features [72] or with the advent of 3D CNN models [124, 59]. For downstream tasks such as video classification and object detection, the use of self-attention with convolution operation has been well studied in [50].

The use of DNNs in a supervised learning paradigm for health data analysis has become state-of-the-art in the detection of critical situations and prognostics [17]. The lower the number of cases to be checked by a human operator, the more acceptable automatic decision-making systems is for in-the-wild monitoring of frail subjects. Self-attention model in DNNs has proven to be efficient in increasing the accuracy of detectors on physiological signals as was shown in [86]. Nevertheless, there still remains a place for improvement.

In the healthcare domain, transformers remain mainly designed for the mining of medical records, sometimes being employed for the joint mining of images and text [55]. Transformers architectures used for the extraction of spatiotemporal visual

data, such as video, have recently been proposed [149], [82], adding temporal attention to the spatial attention of visual transformers designed for image analysis [36]. The difference of our work from such transformers [16], consists in introducing a learnable temporal pooling operation for better frame selection from a video segment. The use of pooling in the transformer is also studied in works like [149, 42] but our core architecture as well as the pooling operation are different. In [42], the pooling operation is used to hierarchically expand the channel capacity and pool spatial resolution, similar to CNN. The [149] uses pooling to reduce the temporal resolution with the *topk_std* approach. In the attention matrix, where each row corresponds to a frame in a video clip, they retain only k rows according to their strongest standard deviation of them. Thus, the video frames with the most concentrated attention are retained as representative frames of the clip. On the contrary, in our method, we learn the temporal locations of k important frames and retain frames accordingly.

As the aging of the population becomes a massive phenomenon with the proliferation of age-related diseases, there is a growing need for monitoring technologies for assisted living. Patients with chronic and age-related diseases require surveillance under "ecological conditions" at their homes.

For the elderly without a particular disease, risk situations often rely on the context and their physiological conditions, e.g., a person being in a stressed condition and cooking could forget fire on the cooker, that is, has a domestic accident risk.

Wearable IoT devices have been successfully penetrating the practices of monitoring frail subjects in the AAL paradigm. In a recent review [115], the authors have

selected the most relevant devices for AAL, specifically wearables. When analyzing the vast literature available today on the use of IoT for elderly care, one can state that such systems perform a large number of physiological and dynamic measures such as body temperature, pulse oxygen saturation, blood pressure, acceleration, and angular velocity measured in different parts of the body.

The recorded data represent time series. The video data from the wearable cameras, the ego video data, is the unavoidable component of understanding complex risk situations related to the context in which the person evolves. We call these risk situations "semantic risks"[144].

The chapter is organized as follows: Section 4.1 describes the proposed architecture of the video transformer with pooling. In Section 4.2 our experiments and results are reported. Finally, the conclusions and future work are discussed in Section 4.3.

4.1 Pooling Video Transformer for Detection of Semantic Risk Situations

To detect risk situations from the presented taxonomy, we propose a video transformer architecture. Our transformer is based on the Visual Transformer (ViT) [36]. Contrarily to video transformers that have recently been proposed [82, 5], in our transformer we introduce a pooling operation on the input video data in the representation space to select the most important frames, thus reducing the temporal redundancy of visual information. To recover the original temporal information, we

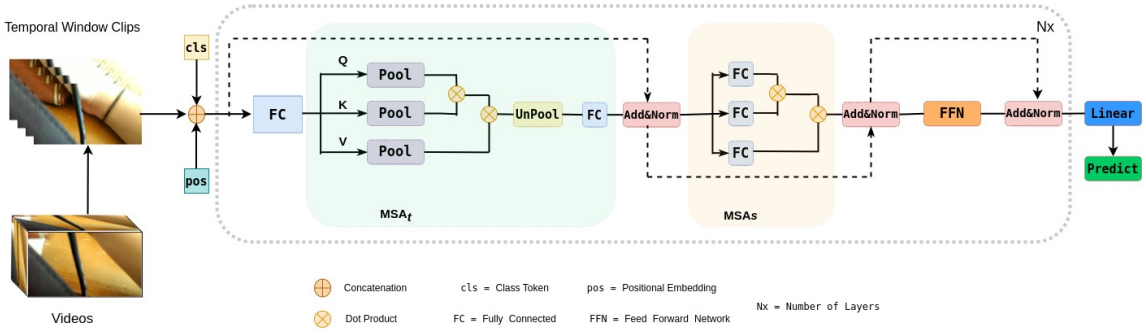

Figure 4.1: Video Transformer with separable spatial and temporal attention for the videos

also symmetrically introduce the unpooling operation. We detail our transformer in the following.

4.1.1 Visual Transformer Architecture

The video transformer architecture that we propose is adapted from Visual Transformer (ViT) [36] which is inspired by the original transformer model [128] used for natural language processing tasks. The ViT was proposed for images. Here, each image is split into N non-overlapping patches, $\mathbf{x_i} \in \mathbb{R}^{P\times P}$, $i = 1, \cdots, N$. We flatten the patch to the 1D dimension and get the vector of size P^2. Note that we dropped the channel dimension for simplicity of notation. Then, a trainable linear projection of each patch of size P^2 is performed, yielding 1D tokens $\mathbf{z_i} \in \mathbb{R}^d$ with d being the target dimension of latent vector space. The projected output $\mathbf{z_i}$ is called the "patch embedding". Therefore, we get for N patches an $N \times d$ embedding matrix. Hence, we denote $\mathbf{E}$ as an embedding operator of the $(\mathbf{x_i})_{i=1,\cdots,N}$ patches (the combination of the flattening and the projection). We further prepend the class token $\mathbf{z_{cls}}$ to the embedding of the input patch and get $[\mathbf{z_{cls}}, \mathbf{E}(\mathbf{x_1}), \cdots, \mathbf{E}(\mathbf{x_N})]$. The class token

$\mathbf{z_{cls}}$ is a learnable embedding of dimension $\mathbb{R}^d$ that is initialized with 0. Finally, $\mathbf{E}_{pos} \in \mathbb{R}^{d\times(N+1)}$, called positional embedding, is added to retain positional information. Thus the $\mathbf{Z}^\ell$ input of l^{th} transformer layer $l = 1, \cdots, L$. The Equation (4.1) gives the expression for 0^{th} layer. The output of the 0^{th} layer is used to compute the input of the next layer as presented in Equation (4.2).

$$\mathbf{Z^0} = [\mathbf{z_{cls}}, \mathbf{E}(\mathbf{x_1}), \cdots, \mathbf{E}(\mathbf{x_N})] + \mathbf{E}_{pos} \tag{4.1}$$

The L transformer layers constitute the encoder. The encoder consists of alternating layers of *multiheaded self-attention* block and *multi-layer perceptron* block. Layer Normalization (LN) is performed before each block and residual connections are added after each block. If we denote by $\hat{\mathbf{Z}}^\ell$ the output of a transformer layer, by MSA - multihead self-attention operation of the transformers [36] and by MLP multi-layered perceptron, then the equations for the alternating layers are given below, see Equations (4.2) and (4.3) for MSA and MLP layers, respectively. In Equation (4.2) and (4.3) we have $\ell = 1....L$.

$$\hat{\mathbf{Z}}^\ell = \mathrm{MSA}(\mathrm{LN}(\mathbf{Z^{\ell-1}})) + \mathbf{Z^{\ell-1}} \tag{4.2}$$

$$\mathbf{Z^{\ell+1}} = \mathrm{MLP}(\mathrm{LN}(\hat{\mathbf{Z}}^\ell)) + \hat{\mathbf{Z}}^\ell \tag{4.3}$$

4.1.2 Spatio-Temporal Transformer

We divide each video into M temporal windows $V_m \in \mathbb{R}^{T\times H\times W}, m = 1, \cdots, M$. Each window V_m is of spatial dimension $H \times W$ of video frames and contains T frames. Further, we call V_m video clips. This is illustrated in Figure (4.1) by a stack of "clips". We divide each clip into spatio-temporal cuboids, splitting each frame into patches as for ViT, see Section 4.1.1. The spatio-temporal transformer is shown in Figure (4.1) Given the patch size in each image as $P \times P$ and flattening each patch, now a clip V_m is a collection of tensors $V_{m,ii=1,\ldots,N} \in \mathbb{R}^{T\times(P^2)}$. In the following, we will drop the index m. Let us now consider a patch in an image at the time moment t: $\mathbf{x}_{(i,t)} \in \mathbb{R}^{P^2}$. Similar to the image transformer we perform a linear embedding for the individual patches as given in Equation (4.4) where $\mathbf{E}_{pos_t} \in \mathbb{R}^{d\times(N+1)}$ and transformer is using a latent vector of size d, $t = 1, ...T$:

$$\mathbf{Z_t^0} = [\mathbf{z}_{cls}, \mathbf{E}(\mathbf{x}_{(1,t)}), \cdots, \mathbf{E}(\mathbf{x}_{(N,t)})] + \mathbf{E}_{pos_t} \tag{4.4}$$

First, we apply temporal attention to the patch $\mathbf{x}_{(i,t)}$. Thus, the temporal attention for a i^{th} patch on the layer ℓ in a given clip is computed as follows. For each block at layer ℓ the query (q), key (k) and value (v) are computed as given in Equation (4.5), (4.6) and (4.7):

$$q^{\ell}_{(i,t)} = W^{\ell}_{q_t}(\mathrm{LN}(\mathbf{Z}^{\ell-1}_{(i,t)})) \tag{4.5}$$

$$k^{\ell}_{(i,t)} = W^{\ell}_{k_t}(\mathrm{LN}(\mathbf{Z}^{\ell-1}_{(i,t)})) \tag{4.6}$$

$$v^{\ell}_{(i,t)} = W^{\ell}_{v_t}(\mathrm{LN}(\mathbf{Z}^{\ell-1}_{(i,t)})) \tag{4.7}$$

Here, $W_{q_t}, W_{k_t}, W_{v_t}$ are weight matrices. The linear layer is illustrated as the FC layer in Figure 4.1 in the left part.

For generalizing on the smaller dataset and incrementing temporal (frame) importance, the pooling operation is used in the temporal self-attention.

$$\hat{q}^{\ell}_{(i,t)} = pool(q^{\ell}_{(i,t)}); \hat{k}^{\ell}_{(i,t)} = pool(k^{\ell}_{(i,t)}); \hat{v}^{\ell}_{(i,t)} = pool(v^{\ell}_{(i,t)}) \tag{4.8}$$

$$s^{\ell}_{(i,t)} = \mathrm{MSA}_t(\hat{q}^{\ell}_{(i,t)}, \hat{k}^{\ell}_{(i,t)}, \hat{v}^{\ell}_{(i,t)}) \tag{4.9}$$

Here *pool* denotes the pooling operation, which consists of dropping frames from a video clip, which we will explicit in the subsection 4.1.3 below. Multi-head self-attention operator (MSA_t) is described in Equation (4.10) (illustrated in Figure (4.1 as $\mathrm{MSA_t}$) using the dot product between the query $\hat{q}^{\ell}_{(i,t)}$ and the key $\hat{k}^{\ell}_{(i,t)}$:

$$s^{\ell}_{(i,t)} = \mathrm{SoftMax}\Big(\frac{(\hat{q}^{\ell}_{(i,t)}) \cdot (\hat{k}^{\ell}_{(i,t)})^T)}{\sqrt{d}}\Big) \cdot \hat{v}^{\ell}_{(i,t)} \tag{4.10}$$

Once we obtain the $s^{\ell}_{(i,t)}$ we unpool or upsample it to the original actual temporal dimension T as in Equation (4.11).

$$\hat{q}^{\ell}_{s(i,t)} = unpool(s^{\ell}_{(i,t)}); \hat{k}^{\ell}_{s(i,t)} = unpool(s^{\ell}_{(i,t)}); \hat{v}^{\ell}_{s(i,t)} = unpool(s^{\ell}_{(i,t)}) \tag{4.11}$$

Here *unpool* denotes unpooling the operation which is inverse of *pool* and is also introduced below. Now, similarly for spatial attention, the query, the key and, the value are computed as given in Equation (4.12), (4.13) and (4.14) :

$$q^{\ell}_{s(i,t)} = W^{\ell}_{q_s}(\text{LN}(\hat{q}^{\ell}_{s(i,t)})) \tag{4.12}$$

$$k^{\ell}_{s(i,t)} = W^{\ell}_{k_s}(\text{LN}(\hat{k}^{\ell}_{s(i,t)}) \tag{4.13}$$

$$v^{\ell}_{s(i,t)} = W^{\ell}_{v_s}(\text{LN}(\hat{v}^{\ell}_{s(i,t)}) \tag{4.14}$$

Here, $W_{q_s}, W_{k_s}, W_{v_s}$ are weight matrices.

Following self-attention with respect to the temporal dimension, spatial attention is performed as given in Equation (4.15) which is described in Equation (4.16) (illustrated in Figure (4.1) in the block MSA_s).

$$s^{\ell}_{s(i,t)} = \text{MSA}_s(q^{\ell}_{s(i,t)}, k^{\ell}_{s(i,t)}, v^{\ell}_{s(i,t)}) \tag{4.15}$$

$$s^{\ell}_{s(i,t)} = \text{Softmax}\Big(\frac{(q^{\ell}_{s(i,t)}) \cdot (k^{\ell}_{s(i,t)})^T)}{\sqrt{d}}\Big) \cdot v^{\ell}_{s(i,t)} \tag{4.16}$$

Finally, as shown in Figure (4.1), $s^{\ell}_{s(i,t)}$ is passed to the multilayer perceptron as given in Equation (4.17).

$$\mathbf{S}^{\ell} = \text{MLP}(\text{LN}(s^{\ell}_{s(i,t)}) + s^{\ell}_{s(i,t)} \tag{4.17}$$

4.1.3 Temporal Pooling and Unpooling

The temporal pooling operation is performed to reduce temporal redundancy, as in egocentric videos the camera could point on the spatially close locations. Therefore, non-important frames can be removed.

The temporal *pooling* and *unpooling* of video have been inspired by [61]. The authors perform temporal downsampling in their 3D convolutional network. Their pooling is realized in the input space. In our work, we propose pooling after tokens have been obtained in a transformer, that is in the feature space. We retain embedded video frames at learnable grid locations on a non-uniform temporal grid. The number of frames to retain is regulated by a target temporal reduction ratio $\alpha < 1$. Thus, the fixed number $n_f = \alpha \times T$ of the most important frames must be retained, with T being the initial number of frames in the clip. We express the importance of frames by $prob_j$ where $j = 1, ...\alpha T$ of their temporal location. To compute this probability, we follow the method proposed in [61]. There, $prob_j$ is obtained by the projection of spatio-temporal features into the space of dimension $\alpha \times T$. In our case, we compute this probability in the q, k, v spaces for each coordinate separately thus getting the probability vector $\mathbf{prob_j} = (prob_j(q), prob_j(k), prob_j(v))^T$. To sample at

a higher rate with higher importance, the cumulative distribution function (**cdf**) of $1 - prob_j(i), i \in (q, k, v)$ is taken for each marginal distribution. Thus, the grid-location at time j can be given as $(loc_j(i) = T \cdot \mathbf{cdf}(1 - prob_j(i)))$. Finally, the input token in q, k or v is interpolated with respect to the learned grid location. This is the *pool* operation as in Equation (4.8). The pooling is carried out in the three spaces q, k, and v, but the temporal locations are different and correspond to the most important temporal features in each of the three spaces. The difference between the computation of self-attention as given in Equation (3.14) is basically putting the self-attention in a ResNet block. The self-attention is computed on the features inside the ResNet block. The difference here is using the self-attention layer after obtaining the features from MSA_t

Prior to the computation of spatial attention, to attain the original temporal resolution, upsampling of the tokens is performed using the inverse mapping of the cumulative distribution functions. The unpooling operation is performed to regain the temporal dimension. The retention of temporal dimension is required for computation of MSA_s, as the temporal dimension needs to be constant across the layers. This is the *unpool* operation in Equation (4.11).

4.2 Experiments and Results

Experiments have been performed on two datasets. For benchmarking, we use the publicly available Kinetics-400 dataset [21]. For risk detection, we use the dataset BIRDS specifically recorded for our project. In the Kinetics-400 dataset, we only used

Figure 4.2: A:(top) A clip of 8 frames of Kinetics-400 dataset for the class *'Bowling'* B:(bottom) A clip of dataset BIRDS for the class *'Environmental Risk of Fall'*

RGB stream without flow stream, as in the reference transformers only this stream is used. Figure 4.2 shows the clips of Kinetics-400 and BIRDS for a single class. The difference in view between the ego video and the general action classification videos is clear.

4.2.1 Benchmarking on Kinetics-400

Kinetics-400 [21] is the dataset for human action classification. The videos are taken from YouTube with real-world scenarios and are of around 10s duration. The dataset contains about 306,000 video clips, about 240,000 as train, 20,000 as validation, and 40,000 as test set. It comprises 400 classes of actions. For our experiments, we have sampled video clips of temporal dimension 8 from the original videos with a sampling rate of 32. Thus, each video clip is a tensor in $RGB - T$ space with dimensions $(3 \times 8 \times 224 \times 224)$. Here $C = 3$ is the number of color channels, $T = 8$ is the temporal length, $H = W = 224$ is the spatial dimension of the cropped video frames.

The results are reported and compared with various models in Table 4.1. They are obtained on the RGB stream. The flow stream is not considered for the computation of the accuracy of models such as [21, 43, 132]. All transformer-based models, e.g.,

Algorithms	Test Acc. %	Pre-Trained
3D-ConvNet [124]	56.1	✓
I3D [21]	71.1	✓
I3D NL [132]	77.7	✓
X3D-M [43]	76.0	✓
TimesFormer [16]	75.1	✓
Video Swin Transformer (Swin-T) [82]	78.8	✓
Video Transformer (with pooling)	78.3	✓

Table 4.1: Test accuracy scores (top1) of various Algorithms on Kinetics-400 on the RGB stream. [16, 82] are compared for the same configuration of input parameters, such as spatial and temporal resolution and frame number we gave above. For our model, we are using the pretrained weights on ImageNet1K followed by the training on Kinetics-400 by 'divided-space-time' configuration for the [16] model.

As can be seen from Table 4.1, the best model in Kinetics-400 is the Video Swin Transformer [82]. Our model with pooling and unpooling performs closely to it. If we compare our model with its baseline from [16], it gives an accuracy increase of 3%.

4.2.2 Risk Category Classification with Video Transformer Model on BIRDS dataset

The recording of the BIRDS dataset has been presented in Chapter 3. The BIRDS dataset is made up of a total of 19,500 videos. We classify 6 situations: 5 risks and *No-Risk* as mentioned in Chapter 3. The class *No Risk* constitutes most of the data, see data distribution between classes in Table 4.2. To balance the dataset and reduce bias in the models, the *No Risk* class was reduced. Only 5% of *No Risk* videos were

randomly selected. This gave a total number of videos of 1800 comprising samples of all classes of our taxonomy.

In the case of the BIRDS dataset, we obtain video clips of 8 frames using a sliding window approach with a stride of 4 frames between two consecutive clips. Indeed, our problem is to detect a risk situation in which temporal borders do not correspond to the borders of the video.

Each clip is a tensor of size $(3 \times 8 \times 224 \times 224)$, as was for the Kinetics-400 dataset, see Section 4.2.1. We split the dataset with a ratio of 75% for the train and 25% for the test data.

Risk Situations	**Video Data**	
	Samples	Percentage
No Risk	18207	94.45%
Environmental Risk of Fall	248	1.30%
Physiological Risk of Fall	156	0.81%
Risk of Domestic Accident	477	2.47%
Risk associated with Medication Intake	114	0.60%
Risk associated to Dehydration	62	0.32%

Table 4.2: Distribution of video data for risk situations for the BIRDS dataset.

All models mentioned in Table 4.3 used are pre-trained with ImageNet1K. BotNet3D mentioned in Table 4.3 is adapted from the BotNet [114] model for the images. We used 3D convolution instead of 2D convolution, and, for self-attention, inflated the 2D features to the 3D features.

For experiments on the BIRDS dataset, we applied stochastic gradient descent (SGD) with a momentum of 0.9, with a constant learning rate of 0.0001, and used a

Algorithms	Test Acc. %	Pretrained
3D-ConvNet [124]	65.32	✓
BotNet3D	70.47	✓
Video Swin Transformer [82]	78.11	✓
TimesFormer [16]	83.29	✓
Video Transformer(with pooling)	**86.77**	✓

Table 4.3: Test Accuracy Scores(top1) of various Algorithms compared to proposed Video pooling Transformer on BIRDS dataset.

batch size of 4. The experiments have been run for a total of 30 epochs on the pre-trained ImageNet1K dataset, as mentioned in Table 4.3. The experiments have been conducted on GPUs NVIDIA P100, V100. As given in Table 4.3, *Video Transformer with Pooling* achieves the best accuracy compared to other video transformers [16, 82]. Compared to the Video Swin Transformer which was the best on Kinetics-400, it gives 8.67% of increased accuracy and 3.5% of increased accuracy with respect to TimesFormer [16].

In this chapter, a factorized form of video transformer was used with separable spatial and temporal attention. The pooling method was proposed for the temporal attention block. To validate this temporal pooling we have compared it with other pooling methods such as *Min Temporal Pooling* and *Max Temporal Pooling* preserving the temporal location to help us for inverse mapping during the unpooling operation. We achieved lower accuracy scores of 83.24% and 82.33%, respectively. The superior performance of the Grid Pooling over average and max pooling can be due to learning the importance of temporal locations to extract the most contributing frames.

4.3 Conclusion

In this chapter, we have proposed a new model of video transformers with pooling and unpooling operations in query, key, and value space. A factorized form of video transformer was used with separable spatial and temporal attention. The pooling method was proposed for the temporal attention block. To validate this temporal pooling we have compared it with other pooling methods such as *Min Temporal Pooling* and *Max Temporal Pooling*. We achieved lower accuracy scores of 83.24% and 82.33%, respectively. The superior performance of the Grid Pooling over min and max pooling can be due to learning the importance of temporal locations to extract the most contributing frames. These operations allow for the use of video frame importance in the decision process.

In the future, we aim to add time series signals obtained from various sensors worn by subjects in monitoring. The added modalities may improve overall performance, as the dynamic and psychological sensors can help reduce false positives for the classes where the inter-class variance in the RGB data is low. Furthermore, interpretability for the multimodal data needs to be considered in order to better understand the features that contribute the most to the results. The following chapter presents the interpretability of transformers. It shows a novel method to construct the saliency maps for them.

Chapter 5

A Self-Attention Weighted Method for Explanation of Visual Transformers

5.1 Introduction

For image classification tasks, humans make an informed decision using the visual cortex that filters the regions relevant to decision-making [67]. DNNs which are bio-inspired models imitate human decision processes, but they are often considered black boxes by human decision makers. This is why explainable AI research has become very intensive [15]. The goal here is to explain the elements and patterns in the input data that have influenced the decision the most. Such an explanation makes human users trust AI tools and is particularly needed in critical application domains such as e.g. medicine. For parameter-heavy models such as *transformers*, it is not evident which features influence the decision. Hence, their explanations are needed to understand the features and thus input importance for a particular

decision. This also helps to provide feedback on the network to optimize it. In the explanation of transformers, the know-how on explainable artificial intelligence (XAI) applied to DNNs can be used and further developed.

Recently, in transformers, self-attention which is the basic block, has been used for the interpretation of decisions as proposed in [2]. Abnar et. al. [2] developed two methods for combining the attention scores across layers, i.e., attention rollout and attention flow. They represent a transformer as an attention graph, where attention from different layers can be backpropagated to previous layers until the input, thus explaining the classification result. In the first method *attention rollout*, the input token identities are assumed to be combined linearly based on attention weights. The latter are trained during the transformer training. Attention weights are then adjusted by rolling them out to capture the propagation of information from input tokens to intermediate embeddings. Rolling out means recursive multiplication of raw attention matrices $A_k, k = i, ...j; i > j$ of transformer layers, thus propagating attention from layer i to layer j. The second method *attention flow* formulates attention propagation as the max flow problem on a graph since it considers the attention graph as a flow network. This method is not class-specific.

Chefer et. al [25] proposed a transformer explanation inspired by the LRP [8] method, which was developed for explaining the decisions of CNNs. On the contrary, their class-specific method, as [25] integrates the relevance scores with the gradient of attention w.r.t class score. The computed relevance score is an updated version from [8] as the author uses both positive and negative attributions, which are simply

the relevance values (computed using Deep Taylor Decomposition (DTD)). For the non-parametric layers (add layer), a normalizing term is added by the authors of [25].

In [26], the authors of [25] extended their work to co-attention methods performing on multisource input (images and text) as well as encoder-decoder attention.

The contributions of our work are in proposing a self-attention-based explanation method for vision transformers. This method is class-specific, but model-agnostic compared to popular relevance propagation-based methods. We compare our method with the state-of-the-art explainers, a recently adapted version of *Relevance Propagation* for the transformers [25] being amongst them.

When comparing methods, we are based on the hypothesis that a good explanation of a network or a transformer has to correlate with human attention deployed in visual recognition tasks. Thus we compare our explanation maps with Gaze Fixation Density Maps (GFDMs) obtained from psycho-visual experiments when humans observe visual scenes in a specific visual recognition task [97, 39]. Hence, the quality of explanations can be measured by usual metrics such as Pearson Correlation Coefficient (PCC) and Similarity(SIM) with Gaze Fixation Density Maps as was proposed in [3] and further developed in [18, 153].

5.2 Proposed Method

In this section, we describe our proposed Self-Attention Weighted (SAW) method for the vision transformers. This is a class-specific and model-agnostic approach. This

is a "sensitivity" based method and not a relevancy-based method as the change in input changes the interpretation.

5.2.1 Computation of Attention

The computation of self-attention in vision transformers is based on the so-called self-attention mechanism [128]. First of all, we briefly recall a typical vision transformer [36], which we have already introduced in Chapter 4. In a vision transformer, an image is divided into N non-overlapping patches $\mathbf{x_i} \in \mathbb{R}^{P \times P}$ sampled on the regular grid. Each patch is flattened to get a vector of size P^2. Then these vectors are the inputs into the transformer network. In the following, we drop the channel dimension for the simplicity of notation. 1D tokens $\mathbf{z_i} \in \mathbb{R}^d$ are obtained from the patches by trainable linear projections where d is the target dimension of the latent space vector. $\mathbf{z_i}$ is called patch embedding. Thus, the embedding matrix is of dimension $\mathbb{R}^{N \times d}$. To aid in classification, the learnable class token $\mathbf{z_{cls}}$ is prepended on the embedding matrix. Finally, the positional embedding $\mathbf{E}_{pos}$ of dimension $\mathbb{R}^{d \times (N+1)}$ is added to retain the positional information of the patches. Thus, the input to the first layer of the transformer is given in Equation (5.1). The computation of attention is different as computed in Chapter 3, and Chapter 4. In the other two chapters, the computation of the self-attention is for videos and signals. In Chapter 3, self-attention is implemented inside a ResNet block whereas in Chapter 4, there is use of temporal and spatial self-attention separately. In this chapter, we are simply computing the self-attention on the image patches. The Equation (5.1) gives the

embedding vector for the 0^{th} layer. This equation is equivalent to the Equation (4.1)

$$\mathbf{Z^0} = [\mathbf{z_{cls}}, \mathbf{E}(\mathbf{x_1}), \cdots, \mathbf{E}(\mathbf{x_N})] + \mathbf{E}_{pos} \tag{5.1}$$

If we consider a transformer of L layers, then at each layer ℓ, the query (q), key (k) and value (v) are computed accordingly to the Equations (5.2, 5.3, 5.4). The W_q^ℓ, W_k^ℓ, and W_v^ℓ represent the weight matrices and LN denotes layer normalization.

$$q^\ell = W_q^\ell(\mathrm{LN}(\mathbf{Z}^{\ell-1})) \tag{5.2}$$

$$k^\ell = W_k^\ell(\mathrm{LN}(\mathbf{Z}^{\ell-1})) \tag{5.3}$$

$$v^\ell = W_v^\ell(\mathrm{LN}(\mathbf{Z}^{\ell-1})) \tag{5.4}$$

The self-attention is computed by Equation (5.5, 5.6).

$$\mathbf{A}^\ell = \mathrm{SoftMax}\Big(\frac{q^\ell \cdot k^{\ell T}}{\sqrt{d}}\Big) \tag{5.5}$$

$$\hat{\mathbf{A}}^\ell = \mathbf{A}^\ell \cdot v^\ell \tag{5.6}$$

Equations (5.5, 5.6) express the attention weight matrices for only one head of a multi-head self-attention block. To obtain the final attention weight matrices, the attention matrices are concatenated.

The architecture consists of encoders, where each encoder comprises alternating layers of the *multiheaded self-attention* block and *multi-layer perceptron* block. The skip or residual connections are also introduced from the input tokens of each encoder to the output of the self-attention block. The skip connection can be seen in the right of Figure 5.1.

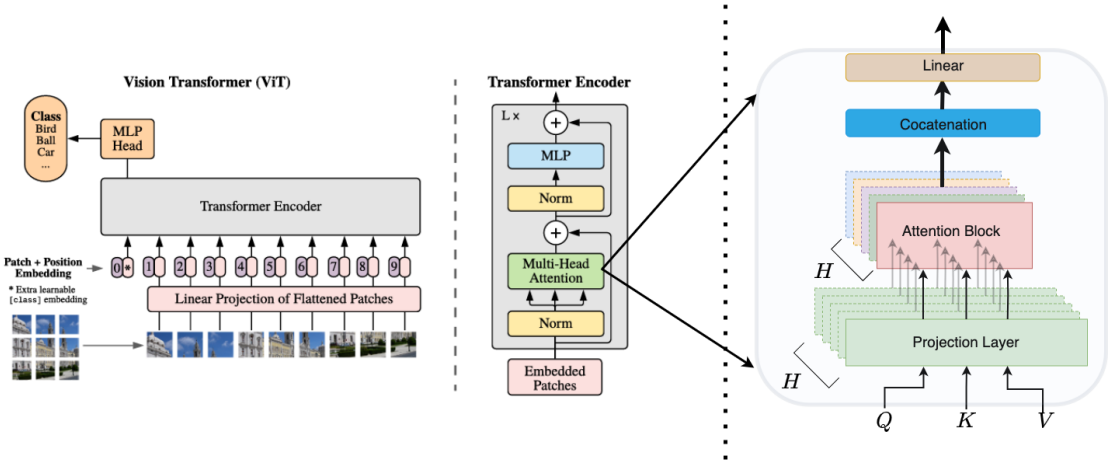

Figure 5.1: Illustration of the Vision Transformer (ViT). (Right): This is the illustration of *Multi-Head Attention* showing different heads. *Projection Layer* constitutes the Linear Layer. (Left): This diagram is taken from [36]

5.2.2 Self-Attention Weighted Method

The idea is to propagate the attention and the gradient of the attention from the last layer to the input patch $\mathbf{x_i}$. The computation of the weighted attention map is inspired from [2] and [25]. The rollout method is class-agnostic as it simply depends on the aggregation of the attention weights as obtained from Equation (5.5). Equations (5.7, 5.8) state the rollout method where I is the identity matrix that accounts for the residual or skip connection and h is the index of the H attention heads.

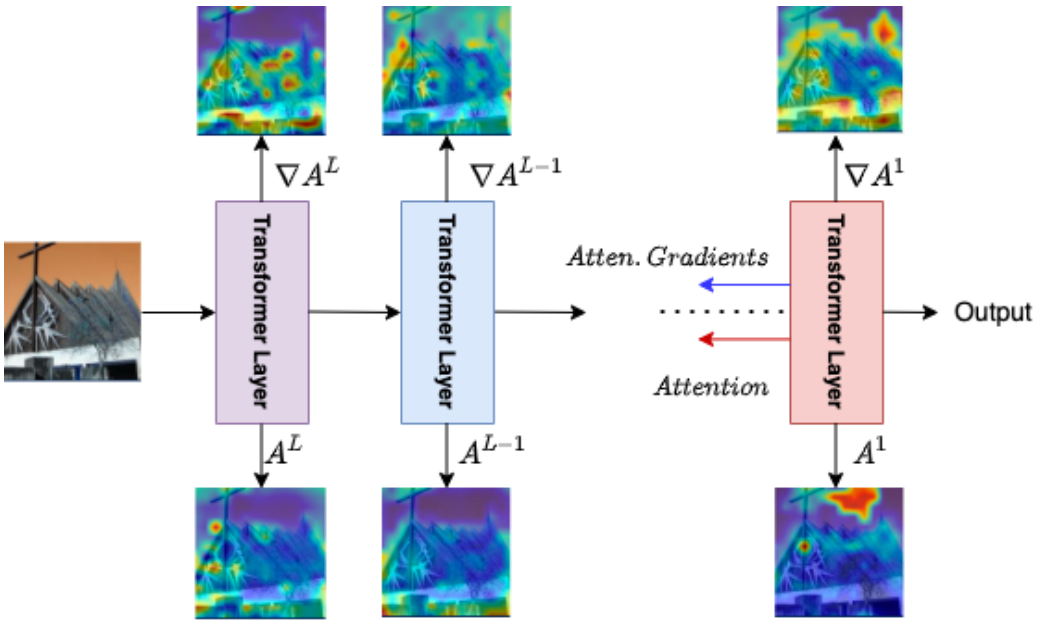

Figure 5.2: Illustration of the attention and attention gradient across the transformer layers.

$$\mathbf{A}'^{\ell} = I + \frac{1}{H}\sum_{h}\mathbf{A}_h^{\ell} \tag{5.7}$$

$$\mathbb{A}'_{rollout} = \prod_{\ell=1}^{L}\mathbf{A}'^{\ell} \tag{5.8}$$

We wish our method to be class-specific, i.e., for each given class from the final classification decision at the generalization step, to explain what were the features in the input token and thus the pixels in the input patch that have contributed to the classification of the patch to the given class.

Therefore, in our *Self-Attention Weighted Method (SAW)*, attention weights obtained from Equation (5.5) are taken and are element-wise multiplied by the gradient of the attention with respect to a specific class. This is expressed by Equations (5.9)

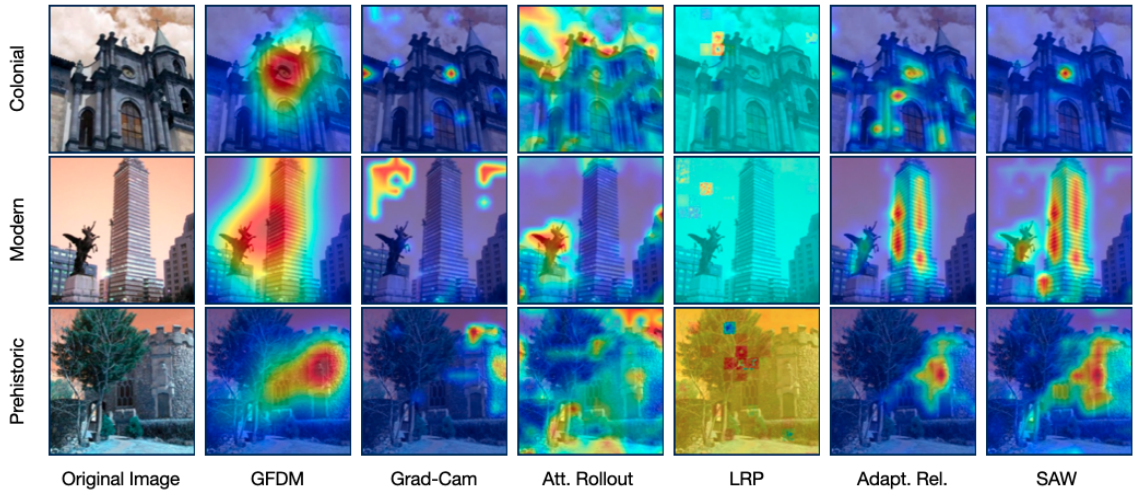

Figure 5.3: Comparison of various explanations (baselines and our proposed method SAW) w.r.t Gaze Fixation Density maps.

and (5.10) below:

$$\mathbf{A'}^{\ell}_{saw} = I + \frac{1}{H}\sum_{h}(\nabla \mathbf{A_h}^{\ell} * \mathbf{A_h}^{\ell}) \tag{5.9}$$

$$\mathbb{A'}_{saw} = \prod_{\ell=1}^{L} \mathbf{A'}^{\ell}_{saw} \tag{5.10}$$

Similar to Equation (5.7), in Equation (5.9) we add I i.e., the identity matrix to account for the residual or skip connections. To extend, the I matrix avoids the self-inhibition of each token. In Equation (5.9), ∇ denotes the gradient and $*$ denotes element-wise multiplication. Figure 5.2 illustrates the attention computation for each layer. The attention matrices are visualized as heat maps in the bottom of the figure and the gradient matrices are depicted in the upper part with a heatmap. Thus the modified attention weights $\mathbb{A'}_{saw}$ are computed from the attention weights of the

trained transformer model. The matrix $\mathbb{A}'_{saw}$ is then interpolated to the resolution of the input image and normalized using min-max to fit the interval $[0, 1]$. In Figure 5.3 $\mathbb{A}'_{saw}$ is visualized as a heat map on the input image together with other heatmaps of reference methods.

Algorithm 1 Algorithm to obtain the Explanation Map

Input: $\text{model}_{trained}, \mathcal{X}$
$\mathbf{A}, \text{y}_\text{c} \leftarrow$ evaluate $(\text{model}_{trained}, \mathcal{X})$
$\nabla \mathbf{A^c} = \frac{\partial \text{y}_\text{c}}{\partial \mathbf{A}}$
$\mathbf{A'}^{\ell}_{saw} = I + \frac{1}{H} \sum_h (\nabla \mathbf{A}^{\mathbf{c}\ell} * \mathbf{A}^{\ell})$
$\mathbb{A}'_{saw} = \prod_{\ell=1}^{L} \mathbf{A'}^{\ell}_{saw}$
$map_c \leftarrow$ Interpolate $\mathbb{A}'$
Output: map_c

5.3 Experiments and Results

In this section, we present the evaluation approach we have followed and benchmark our SAW method against other explainers.

5.3.1 Dataset

We evaluate our SAW method on MexCulture- a public dataset containing images together with gaze fixations [96].

MexCulture: Mex-Culture dataset [93] contains a total of 20000 images of architectural structures for classification of cultural heritage buildings. The taxonomy ranges to four different classes i.e., three classes of architectural structures *Colonial*, *Prehispanic*, *Modern*, and the structures which cannot be classified as in any of the mentioned three classes are classified as *Other*. The 20000 images are divided into

12000 training images, 4000 validation images, and 4000 test images. In this dataset, 284 images are complemented by Gaze Fixation Density Maps (GFDMs), which are computed using the gaze fixations and available at [1]. Gaze fixations were recorded in a psycho-visual experiment where subjects performed a visual task of recognition of an architectural style of historical buildings [93]. Visual saliency-based attention mechanisms seek to bring the external knowledge of high saliency regions in images that have been built upon recorded gaze fixations or predicted by a powerful visual attention model, into a DNN. It gives them priority over other regions/pixels in images

5.3.2 Evaluation Scheme and Results

In evaluating the quality of our explanations, we follow the principle that explanation maps are good if they are similar to human observations of the images in a visual recognition task. This approach has become popular now and was, namely, proposed in [18]. Human observations are expressed as Gaze Fixation Density Maps (GFDMs) built upon gaze fixations. We refer the reader to [96] for a detailed explanation of the computation of GFDMs. Heat maps built upon explanation maps are obtained in a number of state-of-the-art methods such as GradCam [111], adapted relevance propagation [25], rollout method [2], and our self-attention weighted method (SAW). All explanations are obtained on Visual Transformer architecture [36] with pre-trained weights on the ImageNet base configuration and fine-tuned on our dataset for the four-class classification problem. In our experiments, images of size 224×224 are

[1] https://www.nakala.fr/data/11280/5712e468

used, the number of layers in the transformer is $L = 12$, the size of image patches is $P = 16$, and the number of heads in the transformer is 12. Training is performed for 20 epochs and the obtained test accuracy is of 89.71%.

We first *qualitatively* evaluate different methods visually comparing them with GFDMs on the same images. An illustration is given in Figure 5.3. It shows a sample from three classes of buildings: Colonial, Modern, and Pre-Hispanic, the images are selected randomly from the three classes. The reference GFDMs are depicted in the second column. In all three classes, our proposed method is closer to visual attention than the maps of four state-of-the-art methods as we can see from Figure 5.3. When classifying images, the network not only focuses on the object but also on the other parts, similar to visual attention, which uses outside knowledge in the prediction. Most of the available literature performs the quantitative evaluation against the bounding boxes and segmentation maps. This may not be always correct. An object part, background, texture, shape, and other forms of semantics can also contribute towards the recognition of the object. Humans have prior knowledge as well as the surrounding knowledge to make decisions.

For *quantitative* comparison, we use two metrics to compare the saliency maps as in [3]. These two metrics are *Similarity* and *Pearson Correlation Coefficient (PCC)*. In the following, a GFDM is denoted as S_1, and pixel importance/explanation maps obtained by an explanation method are denoted S_2. The maps are normalised to sum to one, that is $\sum_i S_{1_i} = \sum_i S_{2_i} = 1$. The first metric, similarity (SIM) considers the

two maps as experimental probability laws. And expresses the intersection of experimental laws. Thus, if the two importance maps completely overlap, then a maximum similarity of 1 is achieved; otherwise, for no overlapping maps, the similarity is of 0 value. The *Similarity* is given in Equation (5.11).

$$SIM(S_1, S_2) = \sum_i min(S_{1_i}, S_{2_i}) \tag{5.11}$$

For the Pearson Correlation coefficient (PCC), if the two maps are perfectly correlated then PCC is close to 1 else if they are absolutely not correlated then PCC is 0. The *Pearson Correlation Coefficient (PCC)* is computed for two saliency/importance maps as given in Equation (5.12) where $\text{cov}(S_1, S_2)$ is the covariance between the maps and $\sigma(S_1)$ and $\sigma(S_2)$ is the standard deviation of S_1 and S_2.

$$PCC(S_1, S_2) = \frac{\text{cov}(S_1, S_2)}{\sigma(S_1) \times \sigma(S_2)} \tag{5.12}$$

Table 5.1 gives the evaluation metrics values between various methods and gaze fixation density maps over 284 images of Mexculture dataset.

Methods	**Similarity** ($\mu \pm \sigma$)	**PCC** ($\mu \pm \sigma$)
Gaze Den. Map v/s LRP [8]	0.4963 ±0.0283	0.042 ±0.1800
Gaze Den. Map v/s GradCam [111]	0.5516 ±0.0059	0.352 ±0.0230
Gaze Den. Map v/s Attn. Rollout [2]	0.6247 ±0.0031	0.355 ±0.0300
Gaze Den. Map v/s Adapt. Rel [25]	0.6444 ±0.0049	0.456 ±0.0215
Gaze Den. Map v/s SAW (ours)	**0.6682 ±0.0040**	**0.477 ±0.0228**

Table 5.1: Comparison of the metric scores for various baseline methods and to our *Self-Attention Weighted Method*

The scores in Table 5.1 give the quantitative evaluation of the importance/saliency maps explaining transformer decisions compared to the corresponding gaze fixation density maps which are used as ground truth. Upon comparing our method SAW with respect to the baseline methods available in the literature, we note that SAW has the highest similarity of 67% as well as the PCC score of 48% with the gaze fixation density maps.

5.4 Conclusion

In this chapter, we have proposed a novel method *Self-Attention Weighted Method* (SAW), to interpret the decisions by visual transformer networks. This method is class-specific and model-agnostic. It is solely dependent on self-attention for the interpretations of visual transformer decisions. The proposed method has an improved similarity of 2.5% and 2% of PCC improvement with *gaze fixation density maps*, compared to the previous state-of-the-art methods. This method is closest to human visual attention, thus showing that self-attention in visual transformers can be used for the explanation of their decisions. In the next chapter, we have extended the method SAW to spatio-temporal models for the improvement in the training methodology

Chapter 6

Training using Interpretable Deep Learning

Methods of Explainable AI (XAI) are popular for understanding the features and decisions of neural networks. On the other hand, transformers used for single modalities such as videos, texts, or signals as well as multi-modal data can be considered as a state-of-the-art model for various tasks such as classification, detection, segmentation, etc. They generalize better than conventional CNNs as presented in the previous chapters. The use of feature selection methods while training using interpretability techniques can be exciting to train the transformer models. Thus, in this chapter we propose the use of an interpretability method based on attention gradients to highlight important attention weights along the training iterations. This guides the transformer parameters to evolve in the more optimal direction and thus generalize better. This work considers a multimodal transformer on multimodal data: video and signals from sensors. First studied in the video part of our multimodal

data, this strategy is applied to the sensor data in our multimodal transformer architecture before fusion. We show that the late fusion via a combined loss from both modalities outperforms single-modality results. The target application of this approach is Multimedia in Health for the detection of risk situations of lonely persons i.e. frail adults in the @home environment from the wearable video and sensor data (BIRDS dataset). We also benchmark our approach on the publicly available single-video Kinetics-400 dataset to assess the performance, which is indeed better than the state-of-the-art.

The increasing number of parameters in deep models is very evident with the increasing amount of data. With the advent of transformers and parameters ranging from millions to billions, a larger amount of data is needed to generalize. This also explains the fact that the interpretability is lower in these kinds of models due to the higher complexity, which in itself is an inherent challenge. Now, the idea is to see whether these interpretability methods help in getting better models. Various explanations are available for vision as well as language models for a better understanding of the models as presented in Chapter 2. Before transformer-based models CNNs dominated the literature for vision, thus a number of explanation and interpretable models were proposed [111, 151, 148, 75].

In this chapter, we are using BIRDS encompassing two modalities, a) an egocentric video modality and b) a combination of physiological and motion sensors. The target application as presented in Chapter 3 is the detection of risk events amongst old and

frail individuals.

Most of the literature focuses on interpreting the models and using them in a human-understandable manner as is well presented in Chapter 2. In this chapter, the intuition is to focus on the parts of attention that are relevant to the particular class/label, thus we are weighing the gradient of attention to the attention during the training time. Our video models are based on the fine-tuning of models trained on a larger dataset. During the fine-tuning procedure, our model employs two loss terms, the first one is to compute the classification accuracy, and the second one encourages more focus on the relevant attention weights. To better understand the risk events, an additional modality comprising signals from sensors is added to the visual modality. To the best of our knowledge, this is a novel work using interpretability to train a multimodal network.

In this chapter, the contributions are as follows.

- We propose a gradient-based interpretable method that can be used at training time to improve classification scores at the generalization step in both video and signal transformers;
- We are using a multimodal architecture, compounding sensor data (signals) to the visual data (videos) for a better understanding of the context in the application of recognition of risks.

The chapter is organized as follows. In Section 6.1 we present our interpretability technique applicable to both video and signal data and the multimodal architecture that we propose for the recognition of risk events from wearable video and signal

data. In Section 6.2, we present the descriptions of datasets, training using a single modality and multiple modalities, multimodal data organization, etc. For the final Section 6.3, we have the conclusion and discuss our future works.

6.1 Methodology

Our objective is to identify attention weights in a transformer responsible for classifying the data point to a particular class, therefore, added supervision is provided. This added supervision is based on an interpretable method which in itself is a gradient-based method. To cater to the application of recognizing risk events, we apply the same methodology on the video transformer and on the sensor signal transformer.

6.1.1 Training Transformers with Interpretable Methods

The interpretable method for a ViT was proposed in [87]. It uses the gradient of attention. We are similarly using the same gradient of attention to train our video transformer. Our video transformer is based on the Swin-3D model [82]. The intuition to use the gradient of attention is to direct the computation of the attention weights toward the obtained class score as the gradient of attention is computed with respect to the class score.

A vanilla image transformer (ViT) uses the decomposition of the input image into rectangular patches and considers each patch as a vector [36]. The Query (Q), Key(K), and Value (V) matrices are an embedding in the $\mathbb{R}^{(P^2 \times d)}$ space where (P, P) is the patch size and d is the dimension for the linear projection of patches-vectors

into the representation space. Recall that self-attention is computed as the inner product of the *Query* and the *Key* as given in Equation 6.1. The final attention is given in Equation 6.2

$$\mathrm{A} = Q.K^T \tag{6.1}$$

Thus, the attention as proposed in [128].

$$\mathrm{Attention}(Q, K, V) = \mathrm{Softmax}(\frac{QK^T}{\sqrt{d}})V \tag{6.2}$$

If we now consider temporal data such as a bunch of video frames or a bunch of sequential signal recordings, then the computation of attention can be performed as in the Video Swin Transformer [82]. The reason to use the Swin Transformer is due to the lower computational complexity compared to [36]. In [36], the computational complexity is quadratic with respect to the dimension of the input image while the computational complexity for the [81] is linear with respect to the input image. They consider blocks of video frames as a 3D structure with time as the 3rd axis. Instead of taking patches in images, a 3D block is taken in the video as a data point. Each of these data points is embedded to Q, K, and V with the dimension of $(MP^2 \times d)$ where M is the number of temporal windows. It is given by introducing a 3D relative position bias and shifted windows. The 3D relative position bias B gives the geometric relationship between the visual tokens that encodes the relative configurations in the spatial or temporal dimension, that is, encoding the distance between the two tokens. The distance between $[-M+1, M-1]$ for the temporal

axis and $[-P+1, P-1]$ for the spatial axis, i.e., height and width. The attention for the swin-transformer [81] is given in the Equation 6.3 with $Q, K, V \in \mathbb{R}^{(MP^2 \times d)}$ and $B \in \mathrm{R}^{M^2 \times P^2 \times P^2}$. The input video clip, which is our token, is a tensor of $T \times H \times W$ dimension. We divide it into $\lceil \frac{T}{M} \rceil, \lceil \frac{H}{P} \rceil, \lceil \frac{W}{P} \rceil$ non-overlapping windows with a window size of $M \times P \times P$ to get our data points.

$$\text{Attention}(Q, K, V) = \text{SoftMax}\Big(\frac{QK^T}{\sqrt{d}} + B\Big)V \tag{6.3}$$

For our work, we are using interpretable methods to train models in an optimal direction. The core idea of using the gradient of attention ∇A helps in the understanding of change in the attention weights w.r.t classes. Thus, we are multiplying the attention weights A_n at each n^{th} epoch with the gradient of attention w.r.t. the class as given in Equation 6.4, ∇A^c_{n-1} is the gradient of attention w.r.t class c and $A_n \in \mathbb{R}^{MP^2 \times MP^2}$.

$$A_{n_{interpret}} = A_n \cdot \nabla A^c_{n-1} \tag{6.4}$$

We use this to compute the loss for interpretability, as given in Equation 6.5, the objective is to use the interpretable features which highlight the important features from the attention weights and reduce the cost function between the weighted attention weights and the obtained attention weights:

$$\mathcal{L}_{interpret} = \text{CrossEntropy}(A_n, A_{n_{interpret}}) \tag{6.5}$$

The final loss of the training algorithm is given by the weighted sum of the classification loss $\mathcal{L}_{class}$ and the interpretability loss $\mathcal{L}_{interpret}$. The final loss function is in Equation 6.6 where α and β are hyperparameters with $\beta + \alpha = 1$ and $\alpha \geq 0, \beta \geq 0$.

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{class} + \beta\mathcal{L}_{interpret} \tag{6.6}$$

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{class} + (1-\alpha) * [-\sum A_{n_{interpret}} log(A_n)] \tag{6.7}$$

The algorithm is given in Algorithm 2. Training schemes for single modalities (video and signal) are illustrated in Figures 6.1 and 6.3, respectively.

Algorithm 2 Training Algorithm for the Transformer model

if *epoch* is 0 **then**
  $A_n, model_0 \leftarrow$ train $(model_{pt})$ ▷ $model_{pt}$ is the pre-trained model
  $\nabla A_0^c \leftarrow$ evaluate $(model_0)$ (where $c \in 1, ...C$)
else if $1 \leq epoch \leq N$ **then** (where $epoch \in 1, ...N$)
  if $c == Label$: **then**
    $A_n, index, model_n \leftarrow$ train $(model_{n-1}, \nabla A_{n-1}^c)$
    $\nabla A_n^c \leftarrow$ evaluate $(model_n)$
    $\mathcal{L} = \alpha\mathcal{L}_{class} + \beta\mathcal{L}_{interpret}$
    $\arg\min_\theta \mathcal{L} \leftarrow \arg\min_\theta [\alpha\mathcal{L}_{class} + \beta\mathcal{L}_{interpret}]$
  else if $c \neq Label$ **then**
    $A_n, index, model_n \leftarrow$ train $(model_{n-1})$
  end if
end if

6.1.2 Multimodal Architecture

Our multimodal architecture is designed with two streams: a) the video stream which uses Swin-3D [82] architecture, and b) the sensor transformer which uses a LinFormer [113] as the architecture to generalize on the sensor signal data. We have

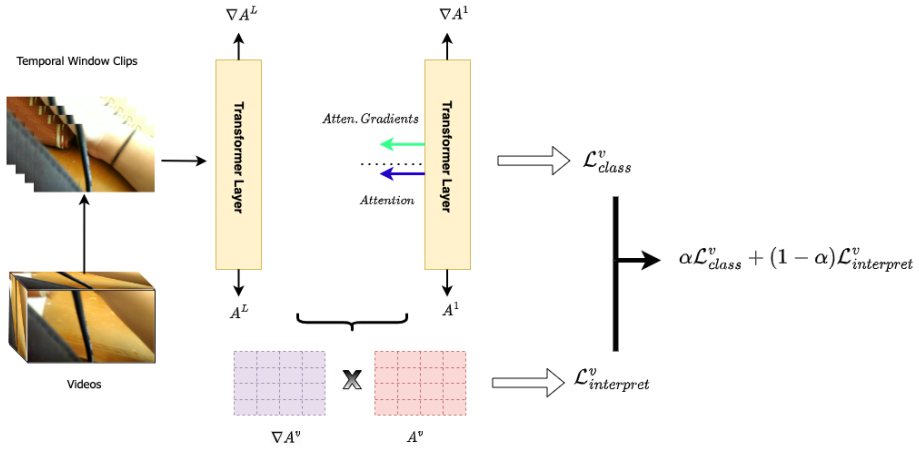

Figure 6.1: The training scheme for the video modality, using the gradient of the attention for additional supervision. A denotes the attention and ∇A denotes the gradient of attention. $\mathcal{L}_{class}$ is the classification loss, $\mathcal{L}_{interpret}$ interpretation loss.

used a late fusion technique via a combined loss on signal and video modalities. Thus, optimization for the combination of the losses is given by:

$$\arg\min_{\theta_v,\theta_s} \Big[\lambda_v \mathbb{E}_{(x_v,y_v)\sim\mathbb{V}} \left[L(\theta_v, x_v, y_v) \right] + \lambda_s \mathbb{E}_{(x_s,y_s\sim\mathbb{S})} \left[L(\theta_s, x_s, y_s) \right] \Big] \tag{6.8}$$

The λ_v and λ_s are hyperparameters to weight the loss functions ($\lambda_s + \lambda_v = 1$ where $\lambda_v \geq 0, \lambda_s \geq 0$). The pair (x_v, y_v) belongs to the distribution $\mathbb{V}$ which is for the video data, where x_v is the input, y_v is the ground-truth labels. Similarly, (x_s, y_s) is the input and ground-truth label pair of the distribution $\mathbb{S}$, that is, for the sensor signal data.

The combination of the two different modalities is illustrated in Figure 6.2.

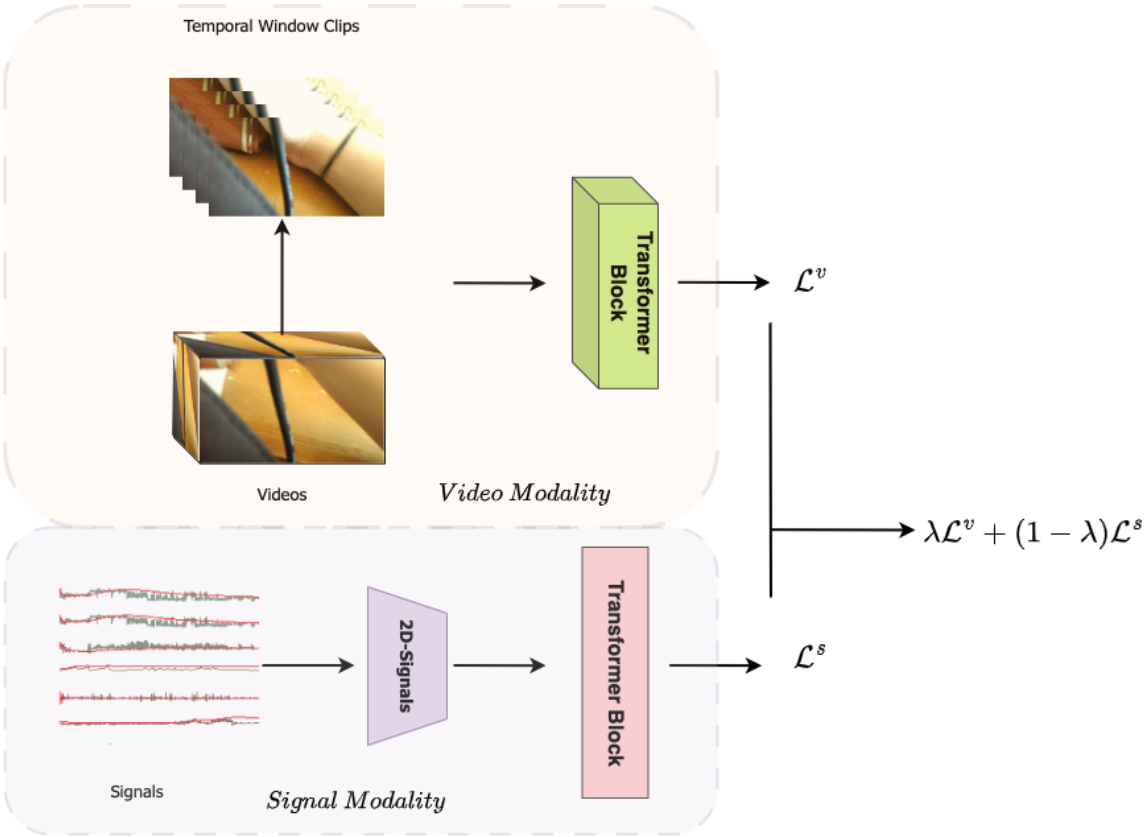

Figure 6.2: The combination of the two modalities and training using the combined loss as given in 6.8.

6.2 Experiments and Results

In this section, we will describe the experiments conducted. We first describe the datasets used and the organization of the data.

6.2.1 Datasets Description

The datasets used are the Kinetics-400 [21] and BIRDS. Kinetics-400 is used for benchmarking on videos as in Chapter 4.

6.2.1.1 Kinetics Dataset

Video (Visual Information) is the most important component in the detection of semantic events. Therefore, we first report on our experiments with only a video transformer and introduce the dataset used.

Kinetics-400 [21] is the dataset for human action classification. The videos are taken from YouTube with real-world scenarios and are about 10 seconds long. The dataset contains around 306,000 videos, about 240,000 videos constitute a training set, 20,000 are selected for validation, and 40,000 for the test. It comprises 400 classes of actions. For our experiments, we have sampled video clips of temporal dimension $T = 8$ from the original videos reduced by temporal sampling as in [21]. Thus, each video clip is a tensor in $RGB-T$ space with dimensions $(\zeta \times T \times H \times W)$. Here $\zeta = 3$ is the number of color channels, $T = 8$ is the temporal length, and $H = W = 224$ is the spatial dimension of the cropped video frames.

6.2.1.2 Multimodal Dataset

For this work, we used the multimodal risk detection dataset named BIRDS (Bio-Immersive Risks Detection System) [1].

A taxonomy of five classes has been defined [143] on immediate and long-term risks as well as presented in Chapter 3. The dataset and the definition of the taxonomy are well presented in Chapter 3, 4. The data distribution and the organization of the dataset is presented below.

This gave a total number of clips as 9600 for all classes of the taxonomy. The

[1] BIRDS will be publicly available upon GDPR clearance

training, validation, and test contain 6713, 923, and 1924 clips, respectively.

6.2.2 Multimodal Data Organization

The multimodal data are organized in the following way: Videos are split into temporal clips of duration τ. Note that this parameter depends on the video frame rate and has to be set experimentally. In our case, we propose $\tau = 8$ for our wearable video setting and variable frame rate in the BIRDS corpus, as well as for the Kinetics-400 dataset, optimized due to the sparse grid search with respect to the global accuracy objective. These clips are sampled with a sliding window approach with a covering of Δ_τ frames. We have set this parameter $\Delta_\tau = 4$ frames on BIRDS and $\Delta_\tau = 5$ on Kinetics-400 datasets, respectively.

The signals and videos are referenced using timestamps. Similarly, we take the corresponding windows from the signals using the timestamp of the videos. Using this timestamp, the corresponding signals are taken into account, where a fixed length window of $\sigma = 20$ with a sliding stride of $\Delta_\sigma = 4$ samples. The organization of signal data is illustrated in Figure 6.3. The temporal windows of σ samples taken from the group of s sensors represent a tensor of dimension $\sigma \times s$. Note that the time intervals between different samples are not stable because of the sensors' imprecision. If there is imprecision on the labels of these windows in the sensors, then we take the statistical mode of the labels in the particular window.

6.2.3 Training of 3D-Swin with Interpretability Techniques on Video

6.2.3.1 Kinetics-400

To train on the Kinetics-400 dataset, we are using the pre-trained weights on the ImageNet-1K dataset for the 3D-Swin transformer. For the Kinetics-400 dataset, in our experiments, we are using the 8-frame clip with a resolution of 224×224. Our method outperforms all the baseline methods; see Table 6.1. In [87], the authors use the class-specific form of interpretation by elementwise multiplying the gradient of attention along with the attention. In our transformer, we use the gradient of attention, see Section 6.1, to weight the importance of attention with respect to the ground truth labels for videos.

Algorithms	**Top-1 Accuracy**	**Pre-Train**
3D-ConvNet [124]	56.1%	✓
I3D [21]	71.1%	✓
I3D NL [132]	77.7%	✓
X3D-M [43]	76.0%	✓
TimesFormer [16]	75.1%	✓
Video Swin Transformer [82]	78.8%	✓
Video Swin T-In (VS-T-In, ours)	**79.1**%	✓

Table 6.1: Test accuracy scores (top-1 accuracy) on the Kinetics-400 Dataset

6.2.3.2 BIRDS Dataset

For training on the BIRDS dataset, we are using the pre-trained model on the Kinetics-400 and fine-tuning our training scheme with these pre-trained weights. Our training scheme generalizes better than the state-of-the-art on the video part of BIRDS corpus. The improvement in the average top-1 accuracy compared to the

vanilla training scheme (without additional supervision) of the Video Swin Transformer [82] is $\sim$ 2.98% as depicted in the last column of Table 6.2 validating our proposed method of additional supervision. 3D-BotNet in the second column of the Table 6.2 is a model for videos adapted from [114] which was devised for images. The nature of the videos for the BIRDS is different from Kinetics-400 as BIRDS are ego-centric videos.

Algorithms	**Top-1 Acc.**	**Pre-Train**
3D-ConvNet [124]	69.27%	✓
3D-BotNet	70.61%	✓
TimesFormer [16]	74.11%	✓
Video Swin Transformer (Swin-T) [82]	73.39%	✓
Video Transformer (with pooling)[88]	75.19%	✓
Video Swin T-In (VS-T-In, ours)	**76.37**%	✓

Table 6.2: Test accuracy scores (top-1) on the BIRDS dataset for the video modality.

As illustrated in Figure 6.4, the per-class accuracy for the video modality (orange bars) and video modality with interpretability (violet bars) differs. In four classes out of six, the accuracy of the transformer with supervision using interpretability is higher. In two classes: *Environmental Risk of Fall* and *Dehydration* (which means detection of drinking action), the baseline Video Swin Transformer performs better. We can explain this by the discrepancy of the data in these two classes corresponding to a very different viewpoint on the environment (class 2) and on the difference of close views (bottles, mugs, and glasses). However, the overall accuracy with the additional supervision by the attention gradient is higher $\sim$ 3% (average top-1 accuracy improvement).

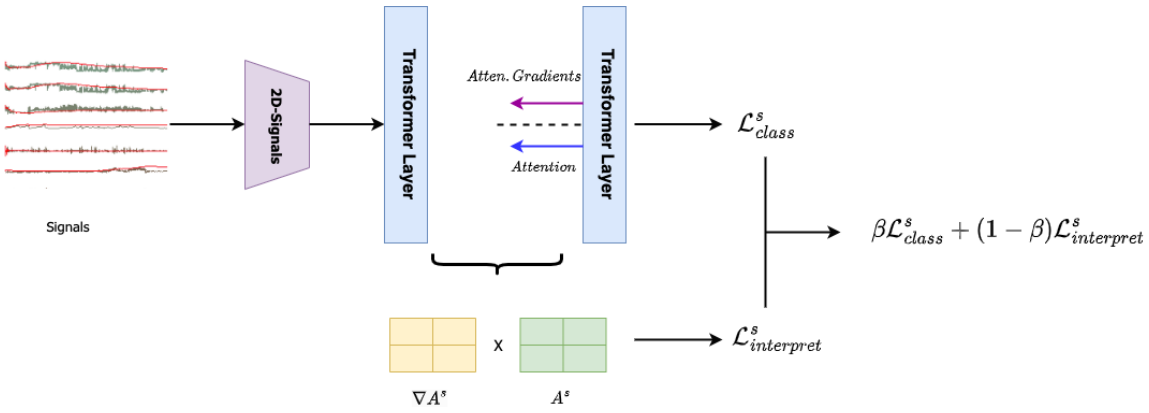

Figure 6.3: The training scheme for generalization in the signal part in the multimodal transformer.

6.2.4 Training of Signal Transformer

The training scheme of the signal transformer is similar to that of the video transformer. The challenge with signal data is non-recording of approximately 95% of the data across various features due to various failures. The data is imputed by replacing the missing values with random values sampled from the normal distribution computed on the particular signal (feature) then the data are linearly scaled along that particular signal. In signal transformers, additional supervision using interpretation gives an improvement of $\sim 4.7\%$. We are not pre-training the signal transformer due to the unavailability of similar datasets which would comprise both physiological and motion sensors. The training scheme is presented in Figure 6.3 and the comparative results are given in Table 6.3.

6.2.5 Multimodal Transformer with Interpretability Results

The accuracy score using the multimodal approach i.e., using the videos and signals together, is **78.26%** in this BIRDS dataset which is an improvement of $\sim 1.9\%$ while

Algorithms	Top-1 Acc	Pre-Train
Vanilla Transformer[128]	32.68	×
Vanilla Transformer-In(ours)	34.07	×
LinFormer[113]	35.55	×
LinFormer-In (ours)	**40.26**	×

Table 6.3: Test accuracy (top-1 accuracy) on the signal sensor data with balanced BIRDS dataset.

using the singular video modality of the network. The overall accuracy score using both modalities together **without** interpretability assisted additional supervision is 76.51%. The improvement of $\sim 1.8\%$ of our method also validates our training scheme. Figure 6.5 shows the accuracy scores of various architecture models on single sensor modality, single video modality, and combined video and signal modalities. It illustrates the better performance of multimodal models and shows that our model with interpretability is the best ($\sim 1.8\%$)

6.2.6 About Ablation

The ablation in a multimodal architecture means the use of only one modality, such as video or signals in our case. We have implicitly done this in the previous section when we were talking about training videos and signal transformers. Therefore, here we briefly discuss and compare these results in a single modality and multimodal setting.

6.2.7 Training Specifications

For the video transformers, all models use 224 resolution frames, with a patch size of 16×16 for transformers. For all the video transformers, we have used tiny models

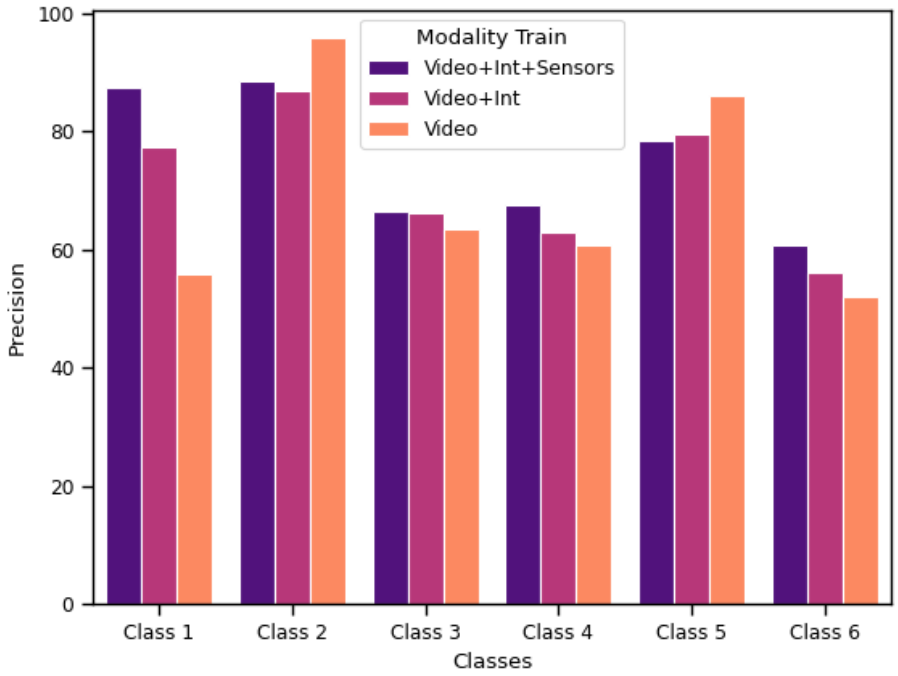

Figure 6.4: Precision scores for each class: 1-No Risk, 2-Environmental Risk of Fall, 3-Physiological Risk of Fall, 4-Risk of Domestic Accident, 5-Risk Associated with Dehydration, 6-Risk Associated with Medication Intake

for the fine-tuning, due to resource constraints. Pre-trained weights are taken from [135]. For the training, we have used a single Tesla A40 GPU. For the BIRDS dataset, the fine-tuning process is for a period of 25 epochs with a batch size of 16. A grid search is used to obtain the learning rate between 1e-3 and 1e-5. Small changes in learning rate do not have a strong impact on the model's classification accuracy. For our approach, we have used an SGD optimizer with Nesterov momentum and weight decay. To weigh the loss function of various modalities, the weighting constants are also obtained using a grid search approach; see Section 6.1. The value of $\alpha = 0.7$, $\beta = 0.65$ and $\lambda = 0.8$ for our experiments.

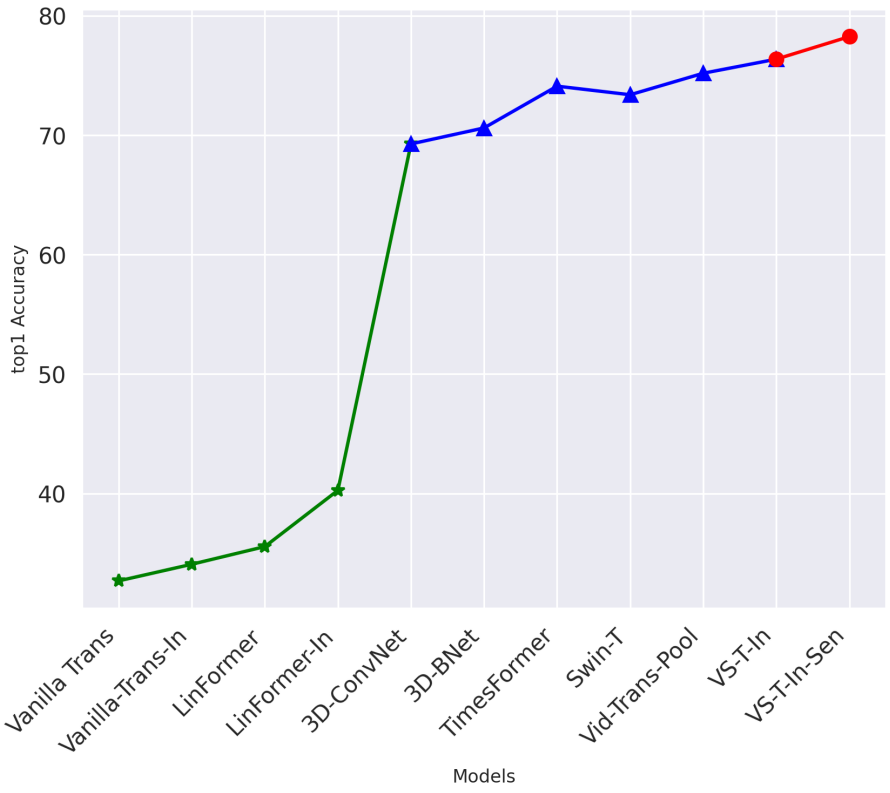

Figure 6.5: Top-1 accuracy scores. In Green: signal modality only. In blue: video modality only. In red: video and sensor modalities together. For BIRDS corpus

6.3 Conclusion

The key element of our work is a scheme to use additional interpretability-based supervision which improves the overall accuracy of any self-attention-based network. To validate our approach, we took a multimodal dataset comprising videos and signals from different sensors. First, our approach was validated on a single modality data i.e., video or signals. Our approach gives a tangible improvement in every single modality tested. In signals, it outperforms the best baseline of $\sim$ 4.8%, in video $\sim$ 1.2%. In the video modality of BIRDS, the proposed method outperforms

the vanilla version of the Video Swin Transformer by $\sim 3\%$. For the publicly available Kinetics-400 dataset, we also achieved better results on video than the best-performing baseline Video Swin Transformer. Computing multimodal architecture on the multimodal BIRDS corpus, we achieved 1.8% top-1 accuracy improvement using interpretation-supervised training. Therefore, we can conclude, that our proposed interpretability technique for training of transformers helps in both single-modality and multimodal model training.

In the next chapter, we focus on the initialization and domain adaptability using interpretable methods.

Chapter 7

Domain Adaptation Using Interpretability

Pre-trained models and transfer learning have played an important role in computer vision and natural language modeling tasks [102, 52]. Therefore, such kind of techniques are naturally transferable to multimodal/multimedia data. Recent models, such as transformers, require a large amount of data for training, as transformer models lack inductive bias. Models such as Vision Transformer (ViT) [36], [103] and BERT [35] show quite an improvement while pre-training them on publicly available datasets, e.g. ImageNet [34]. Pre-training the network and fine-tuning the last layers, preserving the learned features, provides impressive results on downstream tasks. This preserves the information of the prior task while adapting to that of the new tasks.

In the present chapter, we tackle multimodal data, consisting of time-series in the form of signals from and short sequences of egocentric videos from the BIRDS project. We work in a real-world scenario where data recording is challenging. To

efficiently train the proposed models, an adequate transfer learning design is therefore needed. The hypothesis in our work consists in assuming a large domain gap between the source and the target data distributions. The hypothesis originates from the fact that the classification tasks have to be performed on a totally new dataset. The analogous datasets on which the model could have been pre-trained by a conventional fine-tuning approach do not exist. Nevertheless, datasets with a piece of partial information are available. Hence, a domain adaptation is needed.

The differences in the complex multi-modal dataset we have and, available datasets are twofold: i) the data classification tasks are not the same, that is the target taxonomies are different, and ii) the dimension and the nature of the data also differ. We tackle transfer learning from partial datasets to our multimodal dataset with a particular taxonomy of clases for risk detection problem.We have presented the protocol for recording such datasets in a "one-person scenario" and the taxonomy of risk situations to detect in [89] as well as in Chapter 3, Section 3.3. In the present chapter, we propose an efficient transfer learning scheme from partial datasets on multimodal data. Its main components are:

- data dimension adaptation;
- use of interpretable techniques for reinforcement of features in the source domain for better initialization and efficient transfer.

The remainder of the chapter is organized as follows. In Section 7.1, we describe the architecture of our hybrid transformer model for the multimodal data that we

have. Section 7.2 is specifically devoted to our strategy of model training with transfer learning. The results and a discussion are presented in Section 7.3. Conclustion is given in section Section 7.4.

Pre-Training: Transfer learning is one of the key methods for learning pre-trained weights, with the idea of fine-tuning the layers to adapt to the target distribution. Regularization is one of the methods proposed to preserve the information during the pre-training [65]. Regularizing the network by freezing some layers, that is, using parameters trained in the source domain as proposed by Goodfellow and Bengio [47], reduces overfitting during the fine-tuning process. Language processing and image understanding models have shown to be effective when pre-trained on large source domain datasets and then fine-tuned on smaller datasets in the original form or using network surgery [98]. For instance, large language transformer networks such as BERT [35] and GPT-3 [19] with billions of parameters have well exploited the use of large datasets to pre-train the model to obtain stronger performances with downstream tasks on smaller models with surgery[98]. For the image domain, large datasets such as ImageNet21K [107], JFT-300M [118], and JFT-3B have played a pivotal role in the training of large vision models. Kolesnikov et al.[64] used Deep Residual Networks (ResNets) to understand the difference in scaling with pre-trained models on the ImageNet-1K, ImageNet-21K, and JFT-300M. Indeed, as various research has shown [46], for the target domain, a stabilized vs. random initialization is more efficient in terms of attained accuracy.

The majority of transfer learning schemes operate in an inter-domain manner.

This means that CNNs or Transformers (ViT) are pre-trained on a large open publicly available dataset. Then the transfer is performed into the target domain even if the target domain data does not have the same characteristics, e.g. pre-training on LeNet and transfer with fine-tuning to medical image classification tasks [1]. However, the authors of [1] show that intra-domain transfer is more efficient in terms of accuracy. Intra-Domain means that the data have been collected from similar distributions, e.g. transfer between different image modalities, such as in [1]. Furthermore, in [12], the authors propose a self-supervised pre-training for fine-tuning in downstream tasks such as classification, segmentation, etc. The authors in [12] propose that each image is divided into patches and visual tokens, where some of the patches are corrupted using masks. Pre-training predicts the visual token of the original image based on the encoding vector of the corrupted image. The pre-training parameters are first initialized randomly, and then for a given layer, the output matrices of self-attention and feed-forward network are re-scaled.

While it is tedious to find a large amount of data from the same domain to perform the intra-domain transfer, the usage of data from similar, close domains can be possible due to the existence of open corpora. Hence, inter-domain transfer from close domains seems to us a good way to proceed. Nevertheless, the pre-trained model has to be of sufficient quality and stability in order to initialize the target domain model.

Thus, in our approach, we seek two goals:

- close-domain transfer;

- stabilization of the source domain model.

Recently, interpretability techniques have been shown to be efficient in explaining the decisions of deep neural networks. They highlight input data that influence the most decision/classification output [6]. Therefore, it is seducing to use them, in the manner of self-attention models in Neural Networks for filtering out weakly relevant input data in training. Thus, the source model can be stabilized for efficient transfer to the target one. As the goal is to increase the success rate in the target domain, such a stabilization transfer strategy can be designed for a better initialization of the target domain model. This is the core of our approach. Therefore, an appropriate interpretability technique must be chosen to be included in the global process. In the follow-up of this section, we review such techniques and justify our choice.

Interpretability Techniques: These methods for interpretation of decisions of CNNs and transformers have been largely presented in the chapter 2. In the previous chapter 5 we have introduced our interpretability technique for a vision transformer. We will apply it for efficient training in the target domain in this chapter.

In the follow-up sections, we explain our method in detail, starting from the overview of a hybrid architecture for the classification of multimodal data we have designed.

7.1 Hybrid-Model Architecture

We design our transfer learning technique for the hybrid model architecture we have introduced in chapter 3. We will shortly remind it here.

The proposed hybrid transformer comprises two branches; see Figure 7.1. The first branch is a Vision Transformer(ViT) that we have developed for the mining of video data. The second branch is the Signal Transformer (ST). It encodes the sensor signals. According to our previous studies, see chapter 3 video data analysis in a hybrid architecture allows for obtaining quite high accuracies, and signal data can be considered as complementary data, increasing the accuracy via the fusion mechanism. Therefore, in our architecture, we have focused on the design of ViT. For the video data, we use the factorized form of the spatio-temporal attention model that we developed in Chapter 4 on the basis of [16]. We illustrate it in the Figure 7.3 on the frames of an excerpt of the video part of the dataset BIRDS.

The architecture of the signal transformer uses the transformer encoder from [128]. It comprises both the encoder and the decoder modules.

7.2 Model Training

7.2.1 Domain Transfer with Data Dimension Adaptation

The use of pre-trained models on physiological signals is rare in the literature. One cannot find a dataset that contains exactly the same type and quantity of sensor signals as required for the classification problem in the target domain. It is possible to find a dataset with only dynamic signals, such as tri-axial angular velocity and accelerations, e.g. a popular UCI-HAR [4] dataset. Therefore, a direct transfer between the model trained on the source domain dataset for the initialization of the

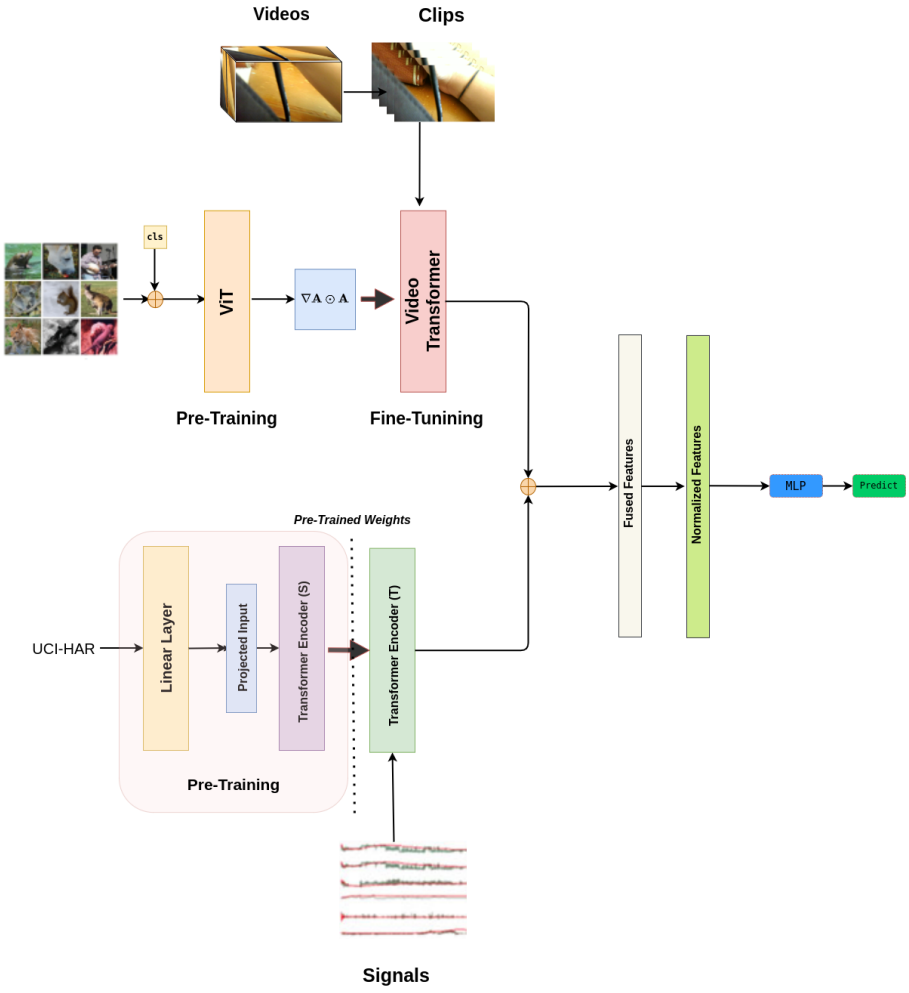

Figure 7.1: The overall scheme constituting both the modalities i.e. for the signals and the videos

target domain model would not be possible. This would result in a huge distribution shift as a new form of features is introduced after the pre-training. The challenge here is to adapt the features of the source domain to the target domain. Hence, we need to embed the source data to achieve the dimension of the target domain data.

The second question to address is the difference in class taxonomies in the source

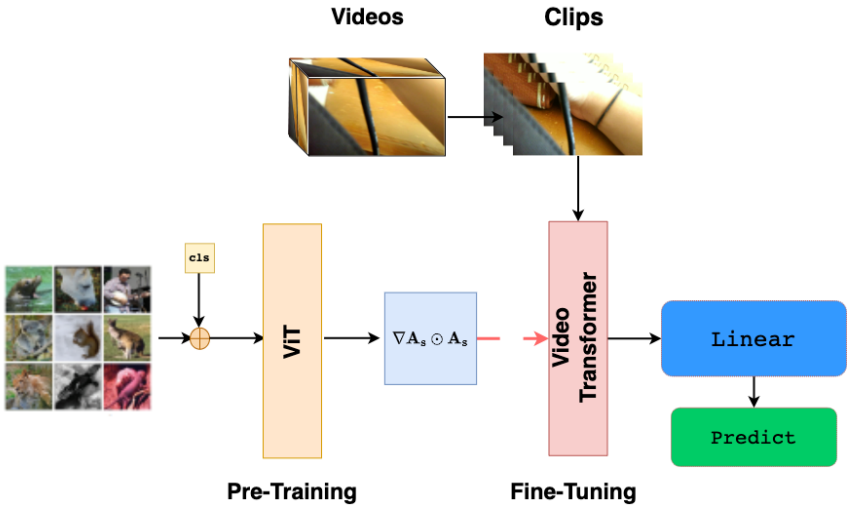

Figure 7.2: The video transformer training scheme. $\nabla \mathbf{A}$ denotes the attention gradient in the trained transformer on the source domain (ImageNet1K), while $\mathbf{A}$ denotes the attention prior to initializing the model trained on the target domain

and target domains. Due to this difference, the shapes of the loss functions will differ, which makes the initialization of an optimizer tricky.

When transferring from the video source domain to the video target domain, the question of dimension adaptation is not raised, as the nature and dimension of source-domain and target-domain data are the same. For the second point, the difference in taxonomies, the problem remains the same, be it a video, a signal, or a hybrid scheme.

For the dimension adaptation of source and target domain data in the signal transformer, we propose a linear transformation to embed and match the features to the target domain.

The domain transfer approach with adaptation to the dimensions of the data is illustrated in Figure 7.4.

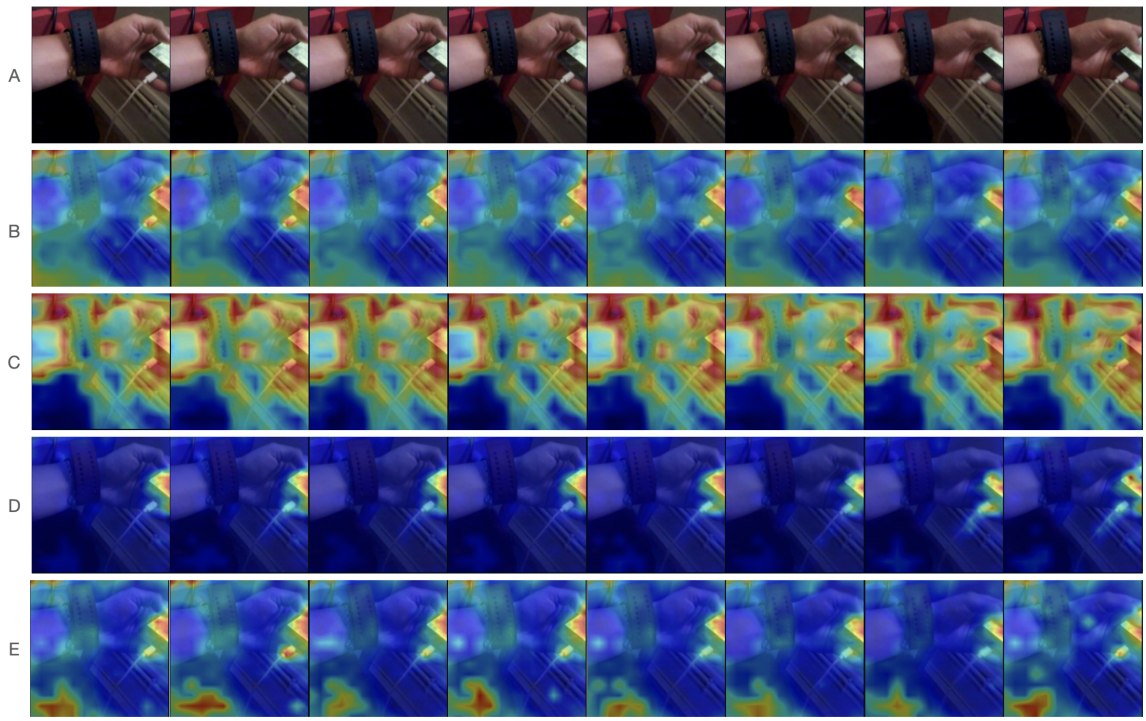

Figure 7.3: For a video clip of 8 frames, A) Actual Frames, B) Attention on the frames, C) Gradient of Attention, D) a Combination of Gradient of Attention and Attention as given in Equation (7.3) and E) a Combination of Gradient of Attention and Attention as given in Equation (7.4) that use SoftMax for the normalization when the gradient is added to $\mathbb{1}$ matrix. The attention and gradient of the attention are computed using the pre-trained weights on the source domain (i.e. ImageNet1K)

The first three blocks perform source domain data encoding and adaptation. The first block in Figure 7.4 is the linear projection of the features. Let us consider our source domain data sample $\mathbf{X}_s \in \mathbb{R}^{S_s \times T_s}$ with S_s the number of sensors and T_s the length of the temporal window. The dimension of the data in our target domain is $S_t \times T_t$ with S_t - the number of target domain sensors and T_t the length of the target temporal window. Thus, the method to adapt the dimension of the source data to the target model is as follows:

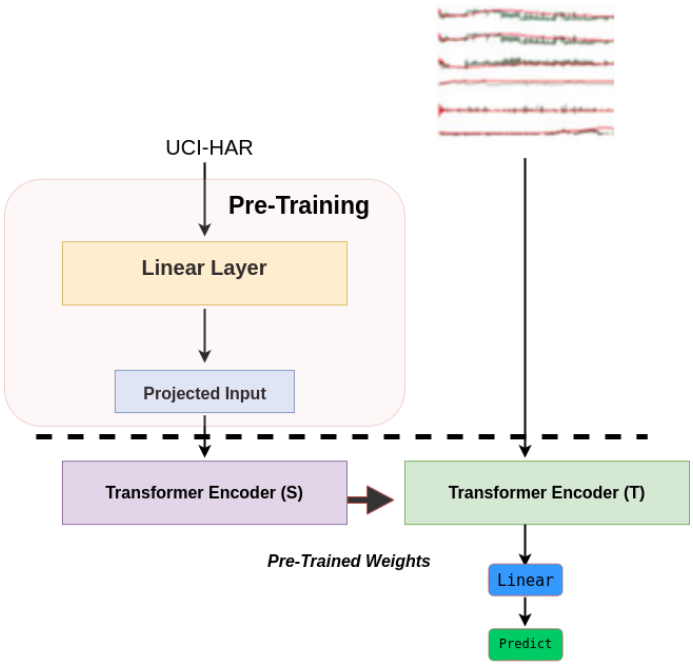

Figure 7.4: The signal transformer training scheme. UCI-HAR dataset has 9 input features while BIRDS has 16 input features, thus a Linear Layer is used as a projection and then trained on Transformer Encoder Layer (Transformer Encoder(S)). For the target domain (Transformer Encoder (T)), the BIRDS are used as the input dataset.

- We perform linear projection of our source input space $\mathbb{R}^{S_s \times N_s}$ to the target space $\mathbb{R}^{S_t \times T_t}$ using flattened version $\mathbf{U'_s}$ of our input data $\mathbf{U'_s} \in \mathbb{R}^{S_s \times T_s}$. This projection is realized accordingly to Equation (7.1).

$$\mathbf{U'}_I = \mathbf{W}_{st} \times \mathbf{U'}_s + \mathbf{B} \tag{7.1}$$

  Here, $\mathbf{W}_{st}$ is the projection weight matrix, $\mathbf{W}_{st} \in \mathbb{R}^{(S_t * T_t) \times (S_s * T_s)}$, $\mathbf{B} \in \mathbb{R}^{S_t * T_t}$ is the bias vector. Thus, we obtain the vector $\mathbf{U'}_I \in \mathbb{R}^{(S_t * T_t)}$

- The vector $\mathbf{U'}_I$ is unflattened and becomes our input data sample $\mathbf{U}_I \in \mathbb{R}^{(S_t \times T_t)}$

to train the transformer of the source domain.

Training in the source domain is carried out as schematized by the block *Transformer Encoder (S)* in Figure 7.4. Thus, we obtain the trained weights $\mathbf{M}_s$ for all layers of the transformer. They are then used for the initialization of the model, depicted as *Transformer Encoder (T)* in Figure 7.4, which is the transformer for our target domain. The weight initialization is realized as given in Equation (7.2).

$$\mathbf{M}_t^0 = \mathbf{M}_s \tag{7.2}$$

Here, $\mathbf{M}_t^0$ is the entire set of initial weights of the target transformer.

Training of the source domain is done end-to-end, where we have the linear layer for the projection of the input dimension to a different feature dimension i.e. the target input dimension. The target domain model has exactly the same transformer architecture as the source domain to facilitate fine-tuning on the target domain.

Since to adapt the weights to the target domain, we perform network surgery and delete the linear layer as shown above the dotted lines in Figure 7.4 (left side) and use the weights of the transformer layer of the source domain to initialize the transformer of the target domain.

7.2.2 Transfer with Interpretability Techniques

Interpretability techniques highlight the features responsible for a particular decision. Our intuition in this work is to use the weights that are used to learn these important features as an initialization for the target domain. In transformers, the basic building

block is self-attention, and the interpretability for transformers in Chapter 5 provides us with a better understanding of the attention weights in the network. Thus, we are using interpretability to focus on the important transformer weights learned in a source domain.

For the video transformer, we are using interpretability techniques, as we have proposed in Chapter 5. We are fusing the gradient of attention with attention. The training scheme is illustrated in Figure 7.2. In Figure 7.3, some examples of video frames, their attention maps, and their gradient maps based on the pre-trained model in the source domain are given. The interpretability technique was used only in the spatial domain in this example. The attention of the source domain $\mathbf{A_s}$ is computed using the inner product of keys and queries from the model trained on the source domain, in the same way as in Equation (6.1) from Chapter 6. To obtain the gradient of attention ∇A_s for the source domain, examples from all classes of the source domain are taken and pushed through the transformer, and gradients of attention for all the examples are then averaged.

The first way to fuse the attention gradient with the attention in the transformer is to element-wise multiply the attention by its gradient as we did in Chapter 5, 6; see Equation (7.3).

$$\mathbf{A_t^0} = \mathbf{A_s} \odot \nabla \mathbf{A_s} \tag{7.3}$$

Here $\odot$ denotes the element-wise multiplication. Strengthening the attention by multiplication with the gradient would enforce the changes, but the multiplication operator has the absorption property. Therefore, another way to do it consists of

keeping the attention in low-gradient areas and reinforcing it when the gradient is strong. This is waht we propose now. Equation (7.4) expresses this approach with $\mathbb{1}$, a matrix composed of 1's.

$$\mathbf{A_t^0} = \mathbf{A_s} \odot \text{SoftMax}(\mathbb{1} + \nabla \mathbf{A_s}) \tag{7.4}$$

In Equation (7.3) and Equation (7.4), $\mathbf{A_s}$ denotes the initial attention obtained from the pre-trained model whereas the $\nabla \mathbf{A_s}$ denotes the gradient of the attention for the pre-trained model, $\mathbf{A_t^0}$ denotes initial attention of the target model i.e the model that will be fine-tuned. Exactly the same approach is applied to the Signal Transformer. The experiments and results are reported in Section 7.3 for signal and video transformers.

7.2.3 Datasets for source and target domains

The datasets used as source domains are:

- ImageNet1K [34] for video data.
- UCIHAR [4] for signal data. It required dimension adaptation proposed in section 7.2.1

The datasets used for the target domains:

- Kinetics-400 [21] and video data of BIRDS.
- The signal data for the target domain is the signal BIRDS.

For the hybrid data, BIRDS dataset was used. Data organisation with temporal windowing has been already presented in 6.

7.3 Experiments and Results

7.3.1 Experiments on Video Modality Transformers

The experiments for the video transformers we proposed have been initially conducted on the open-source dataset, Kinetics-400, see Section 6.2.1.1, for the sake of comparison with State-of-the-Art models. The network used the pre-trained weights on the ImageNet1k dataset trained on ViT [36]. The results are presented in terms of top1 accuracy in Table 7.1. Here, our models are in bold. G depicts the model with attention computed according to the multiplication by its gradient, Equation (7.3) and SG depicts the usage of the model from Equation (7.4).

Algorithms	**Top-1 Accuracy**	**Pre-Train**
3D-ConvNet [124]	56.1%	✓
I3D [21]	71.1%	✓
I3D NL [132]	77.7%	✓
X3D-M [43]	76.0%	✓
TimesFormer [16]	75.1%	✓
Video Swin Transformer (Swin-T) [82]	78.8%	✓
Video Transformer (with pooling)[88]	78.3%	✓
Video Transformer ([G]) Eq: 7.3	77.24%	✓
Video Transformer ([SG]) Eq: 7.4	77.31%	✓

Table 7.1: Test accuracy scores (top-1) on the Kinetics Dataset [21]

Generally, our results on Kinetics-400 are similar to the SOTA. 7.1. We note

that the video Swin Transformer [82] slightly outperforms all models. It gives approximately 1% better accuracy compared to our models. This can be due to the hierarchical embedding of the input image proposed in [81]. Our model with interpretability techniques gives an increase of accuracy compared to our baseline TimesFormer [16] by 2.1% using Equation (7.3) and 2.2% increment using Equation (7.4). We also note that the model of Equation (7.4) slightly outperforms, by 0.7%, the model of Equation (7.3). Thus, we use it in our further experiments.

7.3.2 Experiments on Signal Modality Transformers

To conduct the baseline experiments for the signal modality transformer, we have used an open-source dataset, UCI-HAR [4], see Chapter 6. The same dataset was used then for training the source model. The total number of training instances is 7352, while the test instances are 2947. The evaluation scores are computed for the test set. The accuracy of our model from Equation (7.3) is 91.26% on the UCI-HAR dataset. With the model from Equation (7.4) we obtain very similar but better scores: 91.48%. Compared to the baseline without the gradient of attention (90.6% of accuracy), we state that for the signal transformer, the use of interpretability techniques allows for better scores.

7.3.3 Experiments on Multimodal Data with the Hybrid Transformer

For multi-modality, we use the video data and signals that are weakly synchronized in the BIRDS dataset; see Section 6.2.1.2. A complete synchronization is described in [89]. The data is synchronized using the nearest time stamp of the recording for

the two different modalities. From each stream, we obtain the features as illustrated in Figure 7.1. These features are fused by concatenation to form a shared representation. They are then normalized before they are submitted to the classifier. The two branches of the hybrid transformer are both initialized by the weights trained for the same architecture in the source domain. For the video transformer as a source domain, we use ImageNet1K putting the duration of video clip T = 1 (see section 7.2). For the signal transformer as a source domain, we take the UCI-HAR dataset and perform dimension adaptation for training as described in Section 7.2.1. For the layers that ingest fused features, see the illustration in Figure 7.1, the weights are initialized randomly with a flat Gaussian distribution.

The accuracy for this two-stream architecture with domain adaptation and interpretability technique in both video and signal transformers is 73.61%.

The baseline of the hybrid transformer consists in training the target domain model without any sort of pre-training and without attention models. In this case, the accuracy score is 54.39%.

7.3.4 Ablation Studies

As ablation studies, we understand performing the same classification task on only one modality: signals from sensors or video. As monitoring frail people for privacy reasons is preferred by gerontologists without video data, we first report on the results obtained with a Signal transformer only.

7.3.4.1 BIRDS Sensors

For sensor data, we first trained the dataset using the Transformer Encoder Layer[128] without any pre-training, which gives us the top-1 accuracy of 34.41%, see the first line of Table 7.2. Further, we are pre-training it with the UCI-HAR dataset [4] and dimension adaptation, according to our method see section 7.2.1. Here, the source domain (UCI-HAR) dimension is of 9 sensors and the target domain dimension (BIRDS) has 16 sensors. The use of the gradient of attention according to the model from Equation (7.3) (*Transformer[G]* in the table) is slightly lower (37.29%) than with the model from Equation (7.4) (*Transfromer [SG]* in the table). Finally, without using the gradient of attention at all, only with dimension adaptation, we obtain the highest accuracy of 40.87%. Despite interpretability techniques that have shown their performance on a "clean" dataset UCI-HAR, in the case of the BIRDS dataset quite a lot of signal values are missing and are imputed as in [89]. Therefore, in a real-world scenario, the nature of the data has an impact on the performance.

Algorithms	**Top-1 Accuracy**	**Pre-Train**
Transformer from scratch [128]	34.41%	×
Transformer[G]	37.29%	✓
Transformer[SG]	37.52%	✓
Transformer (Dimen. Adapt)	40.87%	✓

Table 7.2: Test accuracy scores (top-1) on the BIRDS dataset for the sensor modality.

7.3.4.2 BIRDS Video Data

Table 7.3 shows the accuracy scores for the BIRDS video data recorded by one subject, as presented in Section 6.2.1.2. The video model was trained - without pre-training and with pretraining on ImageNet1K datasets as for hybrid transformer. The first conclusion is obvious - the pre-training with transfer from ImageNet1K gives an accuracy increase for all models. To compare our proposed interpretability techniques with the SOTA we used TimesFormer. From our experiments, we conclude that the models of Equation (7.4) and of Equation (7.3) yield very similar results(75.32% compared to 75.55%). We also notice that these results are very close to the results from our previous work with Pooling Transformer [88], where the pooling was performed accordingly to the importance of video frames. Hence adding the "importance" of attention by fusing with the gradient or selecting important frames in video chunks goes in the sense of our intuition - in transformers as in CNNs the feature selection methods based on feature importance have to be included to boost performances.

Having performed this ablation study, we notice that the hybrid transformer performances compared to the best signal transformer performances are strongly increased (from 40.87% to 73.61%). However, the ablation of the signal branch increases the accuracy of the system when performing on video only (from 73.61% to 75.55%). We can explain this by the nature of signal data, which in our real-world scenario had to be strongly imputed. Nevertheless, we think that such a decrease does not remove the added value of a hybrid scheme and that the interpretability

techniques have proven to be interesting for source model training.

Algorithms	Top-1 Accuracy	Pre-Train
3D-ConvNet [124]	64.86%	×
3D-ConvNet [124]	73.27%	✓
TimesFormer [16]	57.29%	×
TimesFormer [16]	74.11%	✓
Video Swin Transformer (Swin-T) [82]	61.74%	×
Video Swin Transformer (Swin-T) [82]	73.39%	✓
Video Transformer (with pooling)[88]	75.21%	✓
Video Transformer ([G]) Eq: 7.3	**75.55%**	✓
Video Transformer ([SG]) Eq: 7.4	75.32%	✓

Table 7.3: Test accuracy scores (top-1) on the BIRDS dataset for the video modality.

7.4 Conclusion

In this chapter, we proposed a transfer learning model for the hybrid transformer for the classification of multimodal data. The core of the method is the use of interpretability techniques in both modalities: signals and video to train the source domain model. We should stress that we worked in the close-domain transfer framework: for video, the source domain was an image dataset, and for signals, the source domain was a signal dataset coming from a similar scenario. Our experiments have shown better performance of target domain models when initialized with trained source domain ones with interpretability techniques. We proposed strengthening attention in the transformers with its gradient by two models- element-wise multiplication and attention conservation and reinforcement. The usefulness of both models

was experimentally shown both on open source datasets Kinetics-400 and UCI-HAR and on a complex multimodal dataset BIRDS for risk detection. We also used a simple dimension adaptation scheme for signals which is quite generic and can be used for any data. These contributions are the last in our research. In the following, we conclude our work and outline its perspectives.

Chapter 8

Conclusion and Perspectives

In this thesis, we presented different solutions for the recognition of concepts in multimodal data in the framework of Deep Learning Paradigm. Our main objective was the methodological advancement of multimodal systems and the interpretation of decisions of various systems in a human-understandable manner.

The target application for most of the thesis was to take up a real-world scenario. For this purpose, we focused on the detection of risk situations of frail people living in home environments from multimodal data recorded with a Bio-Immersive Risk Detection System(BIRDS) developed in our research group together with gerontologists and an SME. In such-in-the-wild data, there are a number of challenges. They were the synchronization of sensors, video, and signal ones, due to the difference in sampling rates between the modalities and the missingness of the data. Another uniqueness of our problem is the rarity of the risks compared to the "*No-Risk*" situations. Indeed, risks can be considered as an anomaly in a person's day-to-day

activity. The classes of risk and no-risk data were therefore imbalanced. In addition to the mentioned challenges, the nature of the video also posed a challenge due to the difference of viewpoint from the general video datasets available with fixed viewpoints. In our research we have proposed several hybrid models starting from a combination of a temporal network, such as GRU autoencoder with a 3D convolution network ResNet, upt to introducing novel hybrid transformer architectures.

In Chapter 3, we devised our first two-stream architecture for the generalization of the multimodal data. Signal modalities obtained from physiological and motion sensors were modeled on a sequence model such as a GRU autoencoder with attention and benchmarked with other temporal models such as LSTM and GRU. For baseline experiments on the signal modality, we tested our algorithm on the UCI-HAR dataset, a human activity recognition dataset. The autoencoder GRU with attention showed performances comparable with LSTM, but with much lower training cost, dividing the number of training epochs by ten. The final accuracy was higher than 90%. On the application concerned dataset, i.e. BIRDS, we obtained state-of-the-art performance, this time with GRU without an autoencoder layer but with attention. Taking into account the semantic nature of our risks, which were context-dependent in our scenario an addition of visual modality was necessary. Therefore, we developed a two-stream architecture in which we weakly synchronized the two modalities due to the difference in sampling rates between the camera and various sensors. The video modality was analyzed using a 3D-ResNet. We used the intermediate fusion technique to combine the two modalities in the latent space. As self-attention

models proved to be efficient even on our complex real-world data, we used self-attention-based models in both modalities to extract the features. To quantify the contribution of self-attention modules, we performed ablation of self-attention models on a singular modality. For videos, the accuracy scores do increase with attention when compared to the ConvNet models. However, in signals, the accuracy decreases when compared to models without attention. This decrease can be attributed to the decrease of inductive biases in the models and the lack of model pre-training. Due to the class imbalance problem, we scraped a lot of data from the "No-Risk" class, as it could overfit the model. Another major problem with the data was the massive imputation of missing data, which may also cause overfitting of the model in the sensors. Our final accuracy with these hybrid networks reached 73.26% in the challenging BIRDS dataset.

Taking into account a better performance of transformer models, reported in the literature, our second hybrid model was designed as a combination of a linear transformer for signals and the BotNet transformer for video. We have inflated BotNet to the third temporal dimension of the video, compared to the reference model, which was proposed only for images. Nevertheless, the transformer model on signal modality did not bring improvement compared to the GRU-with-attention due to the data noisiness we mentioned above.

The hybrid architecture based on transformers gives an accuracy score of 72.19%. In this, we did not use a pure transformer-based/self-attention-based architecture. We used self-attention blocks instead of convolutional blocks in the last ResNet block

as was the case in BotNet.

In Chapter 4, we focus on fully self-attention transformers and propose a method to improve the accuracy of the video part of our data. For this purpose, we used a factorized form of video transformer model with separate temporal and spatial self-attentions, namely TimesFormer. We completed it by computing the importance of the temporal locations using a pooling-based approach in the latent space. This method is not a post hoc method, but rather the temporal locations are learned during the training of the model. As this model was proposed for the video modality, we benchmarked it on the publicly available Kinetics-400 dataset. The method did perform better than other factorized video transformer models but we obtained similar results to the Swin Transformer model which uses hierarchical-based shifted window self-attention blocks. For the BIRDS dataset, we outperform all the models, even the Swin Transformer model for the video. We got an overall increment of the accuracy of 3.5% with regard to the best baseline.

One of the major developments in the last years has been the introduction of transformer models, but the interpretability of these models is still understudied. In Chapter 5, we proposed a novel method for the interpretation of decisions of transformers. This method is class-specific, i.e. it is class discriminative which implies localizing different regions on the same image given different classes. However, this is a model-agnostic method, which implies that any self-attention-based models can use this post-hoc method. We call our method I-SAW: Image Self Attention Weighted

method. Allows for interpretation of the transformer layers and selection of the most important for the decision input for images. In order to evaluate this approach, we applied a methodology developed by our team which consists of the comparison of explanation maps with human perception of visual content, expressed by Gaze Fixation Density Maps (GFDMs) [18]. To benchmark our approach, we used an image dataset MexCulture [93] which provides GFDMs for evaluation purposes. We evaluated our method both qualitatively and quantitatively. The qualitative evaluation consisted of a simple visual comparison of our explanation maps with other state-of-the-art methods. Quantitative comparison consisted in computation of metrics for comparison of saliency maps, such as *similarity* and *Pearson Correlation Coefficient* score. Both comparisons gave the superior performance of our explanation method on transformers. In quantitative comparisons, the method has an improved similarity of 2.5% and 2% of PCC improvement compared to previous state-of-the-art methods. This method is closest to human visual attention, thus showing that self-attention in visual transformers can be used for the explanation of their decisions.

One question that arises is whether explanation methods can help improve the training of models. Interpretable methods point out important areas of an image by identifying the region that led to the model's decision. In Chapter 6, we proposed to use the localization effect of interpretability techniques at the feature and input levels to improve the generalization capabilities of transformers. We use interpretation techniques for both modalities during the training process. An additional supervision was proposed via a specific term in the loss function pushing the attention to

approach the gradient-weighted attention. The proposed method gave an improvement of accuracy by $\sim 5\%$ on signal for a tested Linformer[113]. The improvement has been also observed on the hybrid transformer despite it being lower $\sim 1.8\%$.

Our proposed interpretability technique was applied, in Chapter 7, for domain adaptation. We trained a video transformer on ImageNet1K and applied SAW on the pre-trained weights excluding the last layer. The gain of accuracy was obtained on the video part of BIRDS of $\sim 1\%$.

For the signal transformer on the BIRDS dataset, the gain was $\sim 3\%$. In this chapter, we have also proposed a domain adaptation technique with the adaptation of dimensionality of data from the source domain to the target domain and have applied it to the signal dataset BIRDS as a target domain using UCI-HAR data as a source domain. The dimension adaptation was done by a trained linear layer. Dimension adaptation brought $\sim 3\%$ of improvement compared to the interpretability technique. These results show that the interpretability technique has to be studied better in future works.

Furthermore, data acquisition and studies are required to create a real-world multimodal dataset with less noisy signal data. Similarly, this interpretable technique has been tested to adjust to the domain gap. In this technique, attention is given during initialization, which also improves the transformers' generalization capability.

Hence in our work, we have proposed several solutions for the analysis of multimodal data in order to recognize specific events. Our contributions consisted of the architecture design of DNNs and transformers, the introduction of new interpretability techniques for the explanation of transformer decisions, and the use of them to improve transformer training. The perspectives of this work are numerous. we can summarize them in the following:

- From an information fusion point of view - we need to more exhaustively explore different fusion strategies. In our work, we have used intermediate fusion in the latent space and late fusion by loss combination. It might be interesting to explore early fusion techniques in the input space which have shown to be performant in the e.g. visual attention tasks [23]. Before we can do it, a signal-video synchronization question will have to be further studied. The late fusion has not been extensively studied either. Here it would be interesting to apply methods of symbolic AI considering each network as an agent and to develop fusion rules or use MLPs in the decision space.

- Considering explainability, there can be numerous perspectives. We think that pooling approaches can be used to identify the most important attention heads or layers to get a better explanation. The goal of a saliency-based explanation method is to highlight or localize the important locations, and adaptive pooling approaches can suffice this requirement. There needs to be other methods for evaluating the explanation methods, this can be performed by obtaining the saliency maps and using it as a pseudo-image for a classification task. This is

not the only perspective as, the evaluation of explanation methods is an open and intensively researched question.

- In data representation, the perspectives are also open. In our work, we have used the sensor signals in the temporal domain. In hybrid architectures, audio signals are used at present in the spectral domain, e.g. via spectrograms. This is also one of the possibilities to explore.

- Transformers require a large amount of data to be correctly trained. We have seen the drawback of them when training on a limited amount of positive examples of risk situations in the BIRDS dataset. Thus the perspective of incremental learning with transformers seems to us a promising way to solve the problem of heavy training. Furthermore, incremental learning in real-world applications such as monitoring for risk prevention to adapt to changing living conditions is mandatory. This is the future of our work.

Bibliography

[1] Improving alzheimer's stage categorization with convolutional neural network using transfer learning and different magnetic resonance imaging modalities. *Heliyon*, 6(12):e05652, 2020. ISSN 2405-8440. doi: https://doi.org/10.1016/j.heliyon.2020.e05652.

[2] S. Abnar and W. Zuidema. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.385. URL https://aclanthology.org/2020.acl-main.385.

[3] K. Ahmed Asif Fuad, P.-E. Martin, R. Giot, R. Bourqui, J. Benois-Pineau, and A. Zemmari. Features understanding in 3d cnns for actions recognition in video. In *2020 Tenth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6, 2020. doi: 10.1109/IPTA50016.2020.9286629.

[4] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *ESANN*, 2013.

[5] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6836–6846, October 2021.

[6] M. P. Ayyar, J. Benois-Pineau, and A. Zemmari. Review of white box methods for explanations of convolutional neural networks in image classification tasks. *J. Electronic Imaging*, 30(5), 2021.

[7] H. Azizpour, A. Razavian, J. Sullivan, A. Maki, and S. Carlsson. Factors of transferability for a generic convnet representation. *IEEE Transactions on Pattern Analysis amp; Machine Intelligence*, 38(09):1790–1802, sep 2016. ISSN 1939-3539. doi: 10.1109/TPAMI.2015.2500224.

[8] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10(7):e0130140, 07 2015. doi: 10.1371/journal.pone.0130140. URL http://dx.doi.org/10.1371%2Fjournal.pone.0130140.

[9] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller. How to explain individual classification decisions. *J. Mach. Learn. Res.*, 11:1803–1831, aug 2010. ISSN 1532-4435.

[10] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv*, 2014.

[11] T. Baltrušaitis, C. Ahuja, and L.-P. Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443, 2019. doi: 10.1109/TPAMI.2018.2798607.

[12] H. Bao, L. Dong, S. Piao, and F. Wei. BEit: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=p-BhZSz59o4.

[13] H. Bao, W. Wang, L. Dong, Q. Liu, O. K. Mohammed, K. Aggarwal, S. Som, S. Piao, and F. Wei. VLMo: Unified vision-language pre-training with mixture-of-modality-experts. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=bydKs84JEyw.

[14] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013. doi: 10.1109/TPAMI.2013.50.

[15] J. Benois-Pineau and P. L. Callet. *Visual Content Indexing and Retrieval with Psycho-Visual Models*. Springer Publishing Company, Incorporated, 1st edition, 2017. ISBN 3319576860.

[16] G. Bertasius, H. Wang, and L. Torresani. Is space-time attention all you need

for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, July 2021.

[17] S. Boll, J. S. Lee, J. Meyer, N. Nag, and N. E. O'Connor. Healthmedia'19: 4th international workshop on multimedia for personal health and health care. In *ACM Multimedia*, pages 2720–2721. ACM, 2019.

[18] L. Bourroux, J. Benois-Pineau, R. Bourqui, and R. Giot. Multi layered feature explanation method for convolutional neural networks. In *ICPRAI (1)*, volume 13363 of *Lecture Notes in Computer Science*, pages 603–614. Springer, 2022.

[19] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[20] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In A. Vedaldi, H. Bischof,

T. Brox, and J.-M. Frahm, editors, *Computer Vision – ECCV 2020*, pages 213–229, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58452-8.

[21] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[22] E. Casilari-Pérez, R. Lora-Rivera, and F. García-Lagos. A study on the application of convolutional neural networks to fall detection evaluated with multiple public datasets. *Sensors (Basel, Switzerland)*, 20, 2020.

[23] S. Chaabouni, J. Benois-Pineau, and C. B. Amar. Chabonet : Design of a deep CNN for prediction of visual saliency in natural video. *J. Vis. Commun. Image Represent.*, 60:79–93, 2019.

[24] R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. del R. Millán, and D. Roggen. The opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognit. Lett.*, 34(15): 2033–2042, 2013. doi: 10.1016/j.patrec.2012.12.014. URL https://doi.org/10.1016/j.patrec.2012.12.014.

[25] H. Chefer, S. Gur, and L. Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 782–791, June 2021.

[26] H. Chefer, S. Gur, and L. Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the*

IEEE/CVF International Conference on Computer Vision (ICCV), pages 397–406, October 2021.

[27] S. Chen, P. Sun, E. Xie, C. Ge, J. Wu, L. Ma, J. Shen, and P. Luo. Watch only once: An end-to-end video action detection framework. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8158–8167, 2021. doi: 10.1109/ICCV48922.2021.00807.

[28] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. In *(SSST-8), 2014*, 2014.

[29] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, pages 1724–1734. ACL, 2014.

[30] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[31] Q. Cui, B. Zhou, Y. Guo, W. Yin, H. Wu, O. Yoshie, and Y. Chen. Contrastive vision-language pre-training with limited resources. In S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, editors, *Computer Vision – ECCV 2022*, pages 236–253, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-20059-5.

[32] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018.

[33] G. M. Demirci, S. R. Keskin, and G. Dogan. Sentiment analysis in turkish with deep learning. *2019 IEEE International Conference on Big Data (Big Data)*, pages 2215–2221, 2019.

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[35] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

[36] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and

N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[37] K. Duarte, Y. Rawat, and M. Shah. Videocapsulenet: A simplified network for action detection. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/73f104c9fba50050eea11d9d075247cc-Paper.pdf.

[38] M. Einfalt, K. Ludwig, and R. Lienhart. Uplift and upsample: Efficient 3d human pose estimation with uplifting transformers. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2023.

[39] U. Engelke, H. Liu, J. Wang, P. Le Callet, I. Heynderickx, H.-J. Zepernick, and A. Maeder. Comparative study of fixation density maps. *IEEE Transactions on Image Processing*, 22(3):1121–1133, 2013. doi: 10.1109/TIP.2012.2227767.

[40] D. Erhan, Y. Bengio, A. C. Courville, and P. Vincent. Visualizing higher-layer features of a deep network. 2009.

[41] P. Esser, R. Rombach, and B. Ommer. Taming transformers for high-resolution image synthesis, 2020.

[42] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer. Multiscale vision transformers. *CoRR*, abs/2104.11227, 2021. URL https://arxiv.org/abs/2104.11227.

[43] C. Feichtenhofer. X3d: Expanding architectures for efficient video recognition. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 200–210, 2020.

[44] R. C. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3449–3457, Los Alamitos, CA, USA, oct 2017. IEEE Computer Society. doi: 10.1109/ICCV.2017.371. URL https://doi.ieeecomputersociety.org/10.1109/ICCV.2017.371.

[45] B. Friedrich, S. Lau, L. Elgert, J. M. Bauer, and A. Hein. A deep learning approach for tug and sppb score prediction of (pre-) frail older adults on real-life imu data. In *Healthcare*, volume 9, page 149. Multidisciplinary Digital Publishing Institute, 2021.

[46] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, 2010. URL https://api.semanticscholar.org/CorpusID:5575601.

[47] I. J. Goodfellow, Y. Bengio, and A. C. Courville. *Deep Learning*. Adaptive computation and machine learning. MIT Press, 2016.

[48] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5), aug 2018. ISSN 0360-0300. doi: 10.1145/3236009. URL https://doi.org/10.1145/3236009.

[49] W. Guo, J. Wang, and S. Wang. Deep multimodal representation learning: A survey. *IEEE Access*, 7:63373–63394, 2019. doi: 10.1109/ACCESS.2019.2916887.

[50] X. Guo, X. Guo, and Y. Lu. Ssan: Separable self-attention network for video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12618–12627, June 2021.

[51] Y. Guo, H. Shi, A. Kumar, K. Grauman, T. Rosing, and R. Feris. Spottune: Transfer learning through adaptive fine-tuning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4800–4809, Los Alamitos, CA, USA, jun 2019. IEEE Computer Society. doi: 10.1109/CVPR.2019.00494. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2019.00494.

[52] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao. A survey on vision transformer. *IEEE Transactions on Pattern Analysis amp; Machine Intelligence*, 45(01):87–110, jan 2023. ISSN 1939-3539. doi: 10.1109/TPAMI.2022.3152247.

[53] Q. Han, Z. Fan, Q. Dai, L. Sun, M.-M. Cheng, J. Liu, and J. Wang. On the

connection between local attention and dynamic depth-wise convolution. In *International Conference on Learning Representations*, 2022.

[54] K. Hara, H. Kataoka, and Y. Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? 06 2018. doi: 10.1109/CVPR.2018.00685.

[55] L. A. Hendricks, J. Mellor, R. Schneider, J.-B. Alayrac, and A. Nematzadeh. Decoupling the Role of Data, Attention, and Losses in Multimodal Transformers. *Transactions of the Association for Computational Linguistics*, 9: 570–585, 07 2021. ISSN 2307-387X. doi: 10.1162/tacl_a_00385. URL https://doi.org/10.1162/tacl_a_00385.

[56] G. E. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015. URL https://api.semanticscholar.org/CorpusID:7200347.

[57] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[58] Z. Huang, X. Shi, C. Zhang, Q. Wang, K. C. Cheung, H. Qin, J. Dai, and H. Li. FlowFormer: A transformer architecture for optical flow. *ECCV*, 2022.

[59] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2013. doi: 10.1109/TPAMI.2012.59.

[60] C. F. C. Junior, V. Buso, K. Avgerinakis, G. Meditskos, A. Briassouli, J. Benois-Pineau, I. Kompatsiaris, and F. Brémond. Semantic event fusion

of different visual modality concepts for activity recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(8):1598–1611, 2016.

[61] K. Kahatapitiya and M. S. Ryoo. Coarse-fine networks for temporal activity detection in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8385–8394, June 2021.

[62] X. Z. S. R. Kaiming, H. and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE Computer Society, 2016.

[63] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015. URL http://arxiv.org/abs/1412.6980.

[64] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby. Big transfer (bit): General visual representation learning. In *ECCV (5)*, volume 12350 of *Lecture Notes in Computer Science*, pages 491–507. Springer, 2020.

[65] J. Kukačka, V. Golkov, and D. Cremers. Regularization for deep learning: A taxonomy, 2018. URL https://openreview.net/forum?id=SkHkeixAW.

[66] J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.*, 12(2):74–82, Mar. 2011. ISSN 1931-0145. doi: 10.1145/1964897.1964918. URL https://doi.org/10.1145/1964897.1964918.

[67] O. Le Meur and T. Baccino. Methods for comparing scanpaths and saliency

maps: strengths and weaknesses. *Behavior Research Methods*, pages 1–16, July 2012. doi: 10.3758/s13428-012-0226-9.

[68] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 396–404. Morgan-Kaufmann, 1990. URL http://papers.nips.cc/paper/293-handwritten-digit-recognition-with-a-back-propagation-network.pdf.

[69] G. Letarte, F. Paradis, P. Giguère, and F. Laviolette. Importance of self-attention for sentiment analysis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 267–275, Brussels, Belgium, Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5429. URL https://aclanthology.org/W18-5429.

[70] F. Li, A. Zeng, S. Liu, H. Zhang, H. Li, L. Zhang, and L. M. Ni. Lite detr: An interleaved multi-scale encoder for efficient detr. *arXiv preprint arXiv:2303.07335*, 2023.

[71] J. Li, W. Monroe, and D. Jurafsky. Understanding neural networks through representation erasure. *ArXiv*, abs/1612.08220, 2016.

[72] Q. Li, Z. Qiu, T. Yao, T. Mei, Y. Rui, and J. Luo. Action recognition by

learning deep multi-granular spatio-temporal video representation. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, ICMR '16, page 159–166, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450343596. doi: 10.1145/2911996.2912001. URL https://doi.org/10.1145/2911996.2912001.

[73] W. Li, Z. Lin, K. Zhou, L. Qi, Y. Wang, and J. Jia. Mat: Mask-aware transformer for large hole image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[74] Y. Li, H. Fan, R. Hu, C. Feichtenhofer, and K. He. Scaling language-image pre-training via masking. In *CVPR*, 2023.

[75] H. Liang, Z. Ouyang, Y. Zeng, H. Su, Z. He, S.-T. Xia, J. Zhu, and B. Zhang. Training interpretable convolutional neural networks by differentiating class-specific filters. In *European Conference on Computer Vision*, pages 622–638. Springer, 2020.

[76] K. Lin, L. Li, C.-C. Lin, F. Ahmed, Z. Gan, Z. Liu, Y. Lu, and L. Wang. Swinbert: End-to-end transformers with sparse attention for video captioning. In *CVPR*, 2022.

[77] M. Lin, Q. Chen, and S. Yan. Network in network, 2013. URL http://arxiv.org/abs/1312.4400. cite arxiv:1312.4400Comment: 10 pages, 4 figures, for iclr2014.

[78] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense

object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2020. doi: 10.1109/TPAMI.2018.2858826.

[79] Z. C. Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3): 31–57, jun 2018. ISSN 1542-7730. doi: 10.1145/3236386.3241340. URL https://doi.org/10.1145/3236386.3241340.

[80] K. Liu, Y. Li, N. Xu, and P. Natarajan. Learn to combine modalities in multimodal deep learning. *ArXiv*, abs/1805.11730, 2018.

[81] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, October 2021.

[82] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021.

[83] S. R. Lord, H. B. Menz, and C. Sherrington. Home environment risk factors for falls in older people and the efficacy of home modifications. *Age and ageing*, 35(suppl_2):ii55–ii59, 2006.

[84] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[85] T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1166. URL https://aclanthology.org/D15-1166.

[86] R. Mallick, T. Yebda, J. Benois-Pineau, A. Zemmari, M. Pech, and H. Amieva. A GRU neural network with attention mechanism for detection of risk situations on multimodal lifelog data. In *CBMI*, pages 1–6. IEEE, 2021.

[87] R. Mallick, J. Benois-Pineau, and A. Zemmari. I saw: A self-attention weighted method for explanation of visual transformers. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3271–3275, 2022. doi: 10.1109/ICIP46576.2022.9897347.

[88] R. Mallick, J. Benois-Pineau, A. Zemmari, T. Yebda, M. Pech, H. Amieva, and L. Middleton. Pooling transformer for detection of risk events in in-the-wild video ego data. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 2778–2784, 2022. doi: 10.1109/ICPR56361.2022.9956675.

[89] R. Mallick, T. Yebda, J. Benois-Pineau, A. Zemmari, M. Pech, and H. Amieva. Detection of risky situations for frail adults with hybrid neural networks on multimodal health data. *IEEE Multim.*, 29(1):7–17, 2022.

[90] P. Martin, J. Benois-Pineau, R. Péteri, and J. Morlier. Fine grained sport

action recognition with twin spatio-temporal convolutional neural networks. *Multim. Tools Appl.*, 79(27-28):20429–20447, 2020.

[91] D. Mascharka, P. Tran, R. Soklaski, and A. Majumdar. Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[92] S. Mayor. Nice issues guideline to prevent falls in elderly people. *Bmj*, 329 (7477):1258, 2004.

[93] A. Montoya Obeso, J. Benois-Pineau, M. S. García Vázquez, and A. Ramírez Acosta. Forward-backward visual saliency propagation in deep nns vs internal attentional mechanisms. In *2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6, 2019. doi: 10.1109/IPTA.2019.8936125.

[94] T. M. N. N. K. M. S. Nahiduzzaman, Md and M. Mahmud. Machine learning based early fall detection for elderly people with neurological disorder using multimodal data fusion. In *International Conference on Brain Informatics*, pages 204–214. Springer, 2020.

[95] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 689–696, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

[96] A. M. Obeso, J. Benois-Pineau, M. S. García Vázquez, and A. Álvaro Ramírez Acosta. Visual vs internal attention mechanisms in deep neural networks for image classification and object detection. *Pattern Recognition*, 123:108411, 2022. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2021.108411. URL https://www.sciencedirect.com/science/article/pii/S0031320321005872.

[97] A. M. Obesoa, J. Benois-Pineau, M. S. G. Vazqueza, and A. A. R. Acosta. Attention models in deep neural networks. are deep neural networks as attentive as humans? *To be published*, 12 2020. doi: Pre-Print.

[98] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, pages 1717–1724. IEEE Computer Society, 2014.

[99] A. Owens and A. A. Efros. Audio-visual scene analysis with self-supervised multisensory features. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part VI*, page 639–658, Berlin, Heidelberg, 2018. Springer-Verlag. ISBN 978-3-030-01230-4. doi: 10.1007/978-3-030-01231-1_39. URL https://doi.org/10.1007/978-3-030-01231-1_39.

[100] K. Pérès, A. Edjolo, J.-F. Dartigues, and P. Barberger-Gateau. Recent trends in disability-free life expectancy in the french elderly. *Annual Review of Gerontology and Geriatrics, Volume 33, 2013: Healthy Longevity*, 33:293, 2013.

[101] T. Pozaic, U. Lindemann, A.-K. Grebe, and W. Stork. Sit-to-stand transition reveals acute fall risk in activities of daily living. *IEEE journal of translational engineering in health and medicine*, 4:1–11, 2016.

[102] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang. Pre-trained models for natural language processing: A survey. *CoRR*, abs/2003.08271, 2020. URL https://arxiv.org/abs/2003.08271.

[103] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training.

[104] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021. URL https://arxiv.org/abs/2103.00020.

[105] G. Ras, N. Xie, M. van Gerven, and D. Doran. Explainable deep learning: A field guide for the uninitiated. *J. Artif. Int. Res.*, 73, may 2022. ISSN 1076-9757. doi: 10.1613/jair.1.13200. URL https://doi.org/10.1613/jair.1.13200.

[106] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *KDD*, pages 1135–1144. ACM, 2016.

[107] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor. Imagenet-21k pretraining for the masses. In *Thirty-fifth Conference on Neural Information*

Processing Systems Datasets and Benchmarks Track (Round 1), 2021. URL https://openreview.net/forum?id=Zkj_VcZ6ol.

[108] M. Robnik-Šikonja and I. Kononenko. Explaining classifications for individual instances. *IEEE Transactions on Knowledge and Data Engineering*, 20(5):589–600, 2008. doi: 10.1109/TKDE.2007.190734.

[109] C. Roig, D. Varas, I. Masuda, J. C. Riveiro, and E. Bou-Balust. Generalized local attention pooling for deep metric learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 9951–9958, 2021. doi: 10.1109/ICPR48806.2021.9412479.

[110] C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, 2019.

[111] 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. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[112] A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3145–3153. JMLR.org, 2017.

[113] M. K. H. F. H. M. Sinong Wang, Belinda Z. Li. Linformer: Self-attention with

linear complexity. *CoRR*, abs/2006.04768, 2020. URL https://arxiv.org/abs/2006.04768.

[114] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani. Bottleneck transformers for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16519–16529, June 2021.

[115] T. G. Stavropoulos, A. Papastergiou, L. Mpaltadoros, S. Nikolopoulos, and I. Kompatsiaris. Iot wearable sensors and devices in elderly care: a literature review. *Sensors*, 20(10):2826, 2020.

[116] R. Strudel, R. Garcia, I. Laptev, and C. Schmid. Segmenter: Transformer for semantic segmentation. *arXiv preprint arXiv:2105.05633*, 2021.

[117] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SygXPaEYvH.

[118] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[119] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the*

IEEE/CVF International Conference on Computer Vision (ICCV), October 2019.

[120] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org, 2017.

[121] B. Tavernier, C. Helmer, F. Portet, H. Amieva, I. CarriÃšre, J.-F. Dartigues, J. A. Ãvila Funes, K. Ritchie, L. M. GutiÃ©rrez-Robledo, M. L. Goff, and P. Barberger-Gateau. Frailty Among Community-Dwelling Elderly People in France: The Three-City Study. *The Journals of Gerontology: Series A*, 63(10): 1089–1096, 10 2008. ISSN 1079-5006. doi: 10.1093/gerona/63.10.1089.

[122] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou. Training data-efficient image transformers amp; distillation through attention. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10347–10357. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/touvron21a.html.

[123] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jegou. Going deeper with image transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 32–42, Los Alamitos, CA, USA, oct 2021. IEEE Computer Society. doi: 10.1109/ICCV48922.2021.00010. URL https://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.00010.

[124] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. pages 4489–4497, 12 2015. doi: 10.1109/ICCV.2015.510.

[125] D. Tran, H. Wang, L. Torresani, and M. Feiszli. Video classification with channel-separated convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[126] Y.-H. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L.-P. Morency, and R. Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6558–6569, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1656. URL https://aclanthology.org/P19-1656.

[127] A. Urooj, A. Mazaheri, N. Da vitoria lobo, and M. Shah. MMFT-BERT: Multimodal Fusion Transformer with BERT Encodings for Visual Question Answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4648–4660, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.417. URL https://aclanthology.org/2020.findings-emnlp.417.

[128] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need.

In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[129] A. Vaswani, P. Ramachandran, A. Srinivas, N. Parmar, B. Hechtman, and J. Shlens. Scaling local self-attention for parameter efficient visual backbones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12894–12904, June 2021.

[130] B. Wang, K. Liu, and J. Zhao. Inner attention based recurrent neural networks for answer selection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1288–1297, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1122. URL https://aclanthology.org/P16-1122.

[131] W. Wang, H. Bao, L. Dong, J. Bjorck, Z. Peng, Q. Liu, K. Aggarwal, O. K. Mohammed, S. Singhal, S. Som, and F. Wei. Image as a foreign language: BEiT pretraining for vision and vision-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[132] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[133] Y. Wang and X. Wang. Self-interpretable model with transformation equivariant interpretation. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W.

Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=YlM3tey8Z5I.

[134] Z. Wang, Z. Yang, and T. Dong. A review of wearable technologies for elderly care that can accurately track indoor position, recognize physical activities and monitor vital signs in real time. *Sensors*, 17(2):341, 2017.

[135] R. Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

[136] K. Xia, J. Huang, and H. Wang. Lstm-cnn architecture for human activity recognition. *IEEE Access*, 8:56855–56866, 2020. doi: 10.1109/ACCESS.2020.2982225.

[137] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Neural Information Processing Systems (NeurIPS)*, 2021.

[138] Y. Xu, J. Zhang, Q. Zhang, and D. Tao. ViTPose: Simple vision transformer baselines for human pose estimation. In *Advances in Neural Information Processing Systems*, 2022.

[139] Y. Xu, J. Zhang, Q. Zhang, and D. Tao. Vitpose+: Vision transformer foundation model for generic body pose estimation. *arXiv preprint arXiv:2212.04246*, 2022.

[140] R. Xu-Darme, G. Quénot, Z. Chihani, and M.-C. Rousset. Particul: Part

identification with confidence measure using unsupervised learning. In *Pattern Recognition, Computer Vision, and Image Processing. ICPR 2022 International Workshops and Challenges: Montreal, QC, Canada, August 21–25, 2022, Proceedings, Part III*, page 173–187, Berlin, Heidelberg, 2023. Springer-Verlag. ISBN 978-3-031-37730-3. doi: 10.1007/978-3-031-37731-0_14. URL https://doi.org/10.1007/978-3-031-37731-0_14.

[141] T. Yebda, J. Benois-Pineau, H. Amieva, and B. Frolicher. Multi-sensing of fragile persons for risk situation detection: devices, methods, challenges. In C. Gurrin, B. . Jónsson, R. Péteri, S. Rudinac, S. Marchand-Maillet, G. Quénot, K. McGuinness, G. . Gumundsson, S. Little, M. Katsurai, and G. Healy, editors, *2019 International Conference on Content-Based Multimedia Indexing, CBMI 2019, Dublin, Ireland, September 4-6, 2019*, pages 1–6. IEEE, 2019. doi: 10.1109/CBMI.2019.8877476. URL https://doi.org/10.1109/CBMI.2019.8877476.

[142] T. Yebda, J. Benois-Pineau, H. Amieva, and B. Frolicher. Multi-sensing of fragile persons for risk situation detection: devices, methods, challenges. In *2019 International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–6. IEEE, 2019.

[143] T. Yebda, J. Benois-Pineau, M. Pech, H. Amieva, L. Middleton, and M. Bergelt. Multimodal sensor data analysis for detection of risk situations of fragile people in @home environments. In *MMM (2)*, volume 12573 of *Lecture Notes in Computer Science*, pages 342–353. Springer, 2021.

[144] T. Yebda, J. Benois-Pineau, M. Pech, H. Amieva, L. Middleton, and M. Bergelt. Multimodal sensor data analysis for detection of risk situations of fragile people in @home environments. In *MultiMedia Modeling - 27th International Conference, MMM 2021, Prague, Czech Republic, June 22-24, 2021, Proceedings, Part II*, volume 12573 of *Lecture Notes in Computer Science*, pages 342–353. Springer, 2021.

[145] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10590-1.

[146] B. Zhang, S. Gu, B. Zhang, J. Bao, D. Chen, F. Wen, Y. Wang, and B. Guo. Styleswin: Transformer-based gan for high-resolution image generation, 2021.

[147] L. Zhang, S. Wang, and B. Liu. Deep learning for sentiment analysis: A survey. *WIREs Data Mining and Knowledge Discovery*, 8(4):e1253, 2018. doi: https://doi.org/10.1002/widm.1253. URL https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1253.

[148] Q. Zhang, Y. N. Wu, and S.-C. Zhu. Interpretable convolutional neural networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8827–8836, 2018. doi: 10.1109/CVPR.2018.00920.

[149] Y. Zhang, X. Li, C. Liu, B. Shuai, Y. Zhu, B. Brattoli, H. Chen, I. Marsic, and J. Tighe. Vidtr: Video transformer without convolutions. In *Proceedings of*

the IEEE/CVF International Conference on Computer Vision (ICCV), pages 13577–13587, October 2021.

[150] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. In *International Conference on Learning Representations (ICLR)*, 2015.

[151] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, 2016. doi: 10.1109/CVPR.2016.319.

[152] L. Zhou and C. Gurrin. Multimodal embedding for lifelog retrieval. In *MMM (1)*, volume 13141 of *Lecture Notes in Computer Science*, pages 416–427. Springer, 2022.

[153] A. Zhukov, J. Benois-Pineau, and R. Giot. Evaluation of explanation methods of AI - cnns in image classification tasks with reference-based and no-reference metrics. *Adv. Artif. Intell. Mach. Learn.*, 3(1):620–646, 2023.

[154] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling. Visualizing deep neural network decisions: Prediction difference analysis. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=BJ5UeU9xx.

Apprentissage profond explicable application aux données multimodales

Résumé : Les progrès réalisés par les réseaux neuronaux profonds au cours de la dernière décennie pour diverses tâches de classification ont suscité des inquiétudes quant à la nature "boîte noire" de ces modèles. La fiabilité des décisions des modèles d'IA et la compréhension par l'humain de ces décisions est un problème ouvert. Récemment, avec l'avènement de modèles à base de réseaux neuronaux profonds tels que les transformers, la complexité croissante et le nombre de leurs paramètres rendent l'explicabilité « simple » pour l'humain plus importante. Le travail présenté dans cette thèse peut être divisé en deux parties. La première concerne le développement d'un réseau multimodal destiné la détection des risques pour des personnes fragiles dans un contexte de maintien à domicile. Dans la deuxième partie de la thèse, nous développons des méthodes d'explicitation pour les transformers, plus particulièrement les transformers visuels. Ensuite, nous tirons parti de notre méthode d'explicabilité proposée et l'utilisons pour une meilleure généralisation du transformer multimodal proposé. En effet, l'utilisation de techniques d'explicabilité dans les transformers multimodaux permet d'augmenter la précision de ces classifieurs sur des données complexes du monde réel et ouvre des perspectives intéressantes pour les études sur l'éparcité et la robustesse de ces approches.
Mots-clés : Explainability, Deep Neural Networks, Multimodal Learning, Information Fusion

---

Explainable Deep Learning with Application to Multimodal Data

Abstract: The progress made by deep neural networks over the last decade for various classification tasks in all domains has raised concerns about the "black box" nature of these models. The reliability of decisions from deep neural networks in a human-understandable way is an open problem. Recently, with the advent of deep neural models such as transformers, the increasing complexity and number of parameters make explanations in a human-understandable way more important. The work presented in this thesis can be divided into two parts. The first part concerns the development of a multimodal network for the application of risk detection for frail people in the home environment. In the second part of the thesis, we develop explanation methods for transformers, more specifically visual transformers. Finally, we take advantage of our proposed explainability method and use it for a better generalization of the proposed multimodal transformer. Indeed, the use of explainability techniques in multimodal transformers increases the accuracy of these classifiers on complex real-world data and opens up interesting perspectives for studies on the sparsity and robustness of these approaches.
Keywords: Explainability, Deep Neural Networks, Multimodal Learning, Information Fusion

---

Laboratoire Bordelais de Recherche en Informatique
Université de Bordeaux, 33405 Bordeaux, France.