Final Report

Explainable Artificial Intelligence for Multi-Sensor Information Fusion

CM3203 One Semester Individual Project
40 credits

Author:
Jack Furby

Supervisor:
Prof Alun Preece

Moderator:
Dr Bailin Deng

Contents

Explainable Artificial Intelligence for Multi-Sensor Information Fusion

Jack Furby

1 Abstract

This project focuses on explanations on fusion tasks using a newly introduced explainability technique: VADR, for generating multimodal explanations on multimodal data. To demonstrate this, an efficient but state-of-the-art Neural Network architecture was used as part of a mid-fusion network performing action recognition on a subset of UCF-101 with audio and video streams. The novel approach of applying VADR to a balanced mid-fusion Neural Network had not been previously attempted. To create the model, first, a feature extractor was created, which trained the video and audio subnetworks together, before using the subnetworks within a classifier for training on the action recognition task. The explanations, once applied to the Neural Network provided (1) saliency maps of relevance, separated by modality and temporal and non-temporal information, and (2) the proportion of relevance for each modality. The explanations and proportion of relevance provided insight into features the network had learned and could be used by a human to better understand a prediction. The model produced, despite the weak accuracy, could be retrained with little to no modification to the underlying code in order to provide a more concrete set of results using VADR.

2 Introduction

In recent years artificial intelligence (AI) has seen large advances mainly regarding Machine Learning (ML) using Neural Networks (NN) in prediction and classification tasks. One particular area that is being actively researched is Multimodal Machine Learning (MML) which would lend itself to allow an AI agent to better understand the real world, that itself, is experienced with multiple modalities by humans. In order for an AI agent to understand the world as we do, it will also need multimodal capabilities (Baltrušaitis, Ahuja, and Morency 2019). Fusion is one such technique that assists in this capability for AI agents and is when multiple modalities are joined together before an overall prediction is made. MML enables relations to be created between modalities which becomes useful in "real world" scenarios if for instance a modality is missing or contains noise (D'mello and Kory 2015). This would be the case if a sensor became disconnected or poor environments conditions were encountered such as a camera at night.

NN are regarded as black boxes and they are hard to interpret the reason why a particular prediction or classification occurred. The result of this is a lack of trust in the output produced. Within a human agent team, this lack of trust may lead to the human deciding against the agent, even if the agent is correct. Explainable artificial intelligence (XAI) techniques have provided ways of examining these models with visuals, text, examples, and local explanations (Arrieta et al. 2019). One type of XAI is to use decomposition to work from an output back to the input and thus produce a relevance saliency map of the input. XAI, understandably, becomes harder when multiple modalities are concerned due to the best-fit explanation format changing based on the problem and input modality.

An explanation technique called Video-Audio Discriminative Relevance (VADR) developed by a couple of researchers at Cardiff University applied XAI to a video and audio NN to provide relevance of each modality in addition to explanations for both *temporal* and *non-temporal* information. This was introduced in the paper Taylor et al. in press 2020 with myself as a co-author. This paper explored the technique applied to a mid-fusion model with unbalanced subnetworks. The authors demonstrated when an input video with an audio track, they were able to calculate the relevance of each modality to the prediction in addition to producing saliency maps for video and audio. I have therefore used the same technique, but with the novel approach of using balanced subnetworks which explores an open question raised that the size of the input modality does not affect the output relevance.

This project sets out to create a multimodal action recognition classifier and then by using VADR, explain, evaluate, and compare the classifier to the results the authors of VADR found. This paper is set out with sections 3 and 4 detailing background material, section 5 introduces the implementation and training of the classifier, and section 6 to the end presents the results, conclusion, and future work. *This report is in the format of a research paper.*

3 Background

3.1 Multimodel Machine Learning

MML is a branch of ML where instead of a single modality contributing to an output, multiple modalities are used. This enables the ability to relate the modalities which, potentially, may capture reactions between them (Baltrušaitis, Ahuja, and Morency 2019). It is also the case that additional information may be captured that would be missed given a single modality which is what is experienced in the real world where information in one or more modalities may be missing or containing noise (D'mello and Kory 2015). For example in a CCTV system, audio can capture information that is hidden or not in the direction a camera is pointing.

For the purpose of this paper, MML will be in regard to a NN which is an attempt to represent a biological neural network. The NN will optimise its parameters to a given environment in order to minimise its loss. One particular issue regarding MML is data fusion in which one or more modalities are joined together. This is potentially a difficult task due to the possibility of modalities being of different dimensions such as video being in the format of a 3D array while audio, by default, will be a stream of data.

3.1.1 Fusion

The goal of fusion is to combine multiple streams of data in a beneficial way (Roitberg et al. 2019). In this paper, I am only going to be referring to *model-based* fusion in which the fusion of modalities is built directly into the model. In terms of a NN this will often be achieved by some hidden layer of the network (Potamianos et al. 2003). The alternative approach to model-based fusion is *model-agnostic* fusion which is when the fusion approach is not specific to the ML method in use. Various model-agnostic fusion approaches has been used which can be split mainly into *early fusion*, *late fusion* and *hybrid fusion* (D'mello and Kory 2015). Due to the fact this paper covers NN, model-agnostic fusion is not appropriate as the NN itself can apply the fusion of the modalities.

Fusion of data can be applied at various stages of a NN. Three of the main stages where this can occur are early fusion, *mid fusion*, and late fusion.

Early Fusion is known as *feature-level* fusion. It works by combining unimodal features before a learning method is applied (Snoek, Worring, and Smeulders 2005). Early fusion can still extract features from each modality before combination. The difficulty with early fusion is combining features into a single combined representation. As all features are combined early on, there is a single smaller network than mid or late fusion. This aids in training the model as compared to the equivalent mid or late fusion model, there will be fewer parameters. Early fusion is how it is believed biological brains achieve fusion to some degree (Hall and Llinas 1997).

Mid Fusion combines separate networks at a *feature map* level. This means some early features from the networks are taken into consideration for the final output. After the joining of individual networks, additional layers will are added to the joint representation. A simple method to combine the networks would be to concatenate the individual network outputs (Roitberg et al. 2019). The point at which to join separate networks will impact overall performance with Roitberg et al. 2019 noting the deeper the networks are combined, the better the overall performance.

Late Fusion, otherwise know as *decision-level* fusion, is used to fuse data streams in semantic space (Snoek, Worring, and Smeulders 2005). Late fusion will combine outputs after classification which will make a final output based on individual classification of each modality classifier (hard level) or scores (soft level) (Ebersbach, Herms, and Eibl 2017). The overall output could, for instance, be based on a voting system where each classifier makes a prediction, and the most predicted class is used, or the output could be based on the highest or average confidence.

3.2 Fixed Feature Extractors

A NN is comprised of two main parts; a *feature extractor* and a *classifier* (Ren et al. 2015). The roles of these two components are well defined where, as the name suggests, the feature extractor is designed to extract the important features from an input while the classifier will take the extracted features and map them to an output. In the case of a convolutional neural network (CNN), the boundary between the two components could be between where the convolutional layers and pooling end and the fully connected layer(s) start.

The first layer of a feature extractor will be designed to take an input, with each subsequent layer representing more and more complex patterns. For a CNN you may expect the first layer to capture lines and the next layer moving onto basic shapes and edges. If for instance a CNN was trained on cats and dogs, as the network got deeper, the convolutional filters would start to represent noses, ears, mouths, and tails, etc.

One of the main drawbacks for NN, in general, is the need for vast amounts of data and long periods to train. (Hertel et al. 2015) proposed and tested the ability to take just the feature extractor of CNNs and

retrain the classifier section. This, therefore, would not take as long to train and require a far smaller dataset as (Hertel et al. 2015) stated, a number of the features extracted in different feature extractors were in-fact very similar. In this particular example, the authors managed to yield greater accuracy when using a pre-trained feature extractor. This can be attributed to the original training for the feature extractor being on a far larger dataset than the classifier in the subsequent *fine tuning*. This resulted in the feature extractor having more defined features that may have not been possible without training on the larger dataset.

3.3 Explainable Artificial Intelligence

Another shortcoming of NNs, which affects the ability for humans to trust their predictions, is explainability. A NN is inherently a black box given a NN is comprised of multiple nonlinear layers. Determining why a prediction was made is itself a difficult task (Benitez, Castro, and Requena 1997). XAI techniques have provided ways of examining these models with visuals, text, examples, and local explanations (Arrieta et al. 2019).

One such way of providing an explanation is with a saliency map which highlights the sections of input with visual indicators for the sections that contributed to the output. Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro, Singh, and Guestrin 2016) is an example of an algorithm that can provide a saliency explanation and will approximate a prediction with a black-box model using an interpretable one i.e. linear models or decision trees. LIME aims to offer a user an interpretable explanation of why the output was what it was. It may not be globally fidelity as this remains a complex problem, but it can provide local fidelity.

A second method for XAI is Layer-wise Relevance Propagation (LRP) (Bach et al. 2015) where a model output is decomposed into the contributions of its input (Montavon et al. 2017). This method is used in Deep Taylor Decomposition (DTD). The original authors of DTD demonstrated it with CNN where the method would produce a pixel-wise heatmap as the explanation. This is achieved by distributing the output of a NN onto its respective input with respect to the relevance. LRP and DTD relevancy output will be equal to the relevancy of the model prediction. Whereas LIME has positive and negative relevance, DTD only provides positive relevance as one of its constraints is to be *consistent*, that is the output has the same relevance value as detected by the model.

DTD will perform one backward pass (go from a prediction through a NN to the input), and on each layer perform Taylor decomposition in a divide-and-conquer manner. This is done as Taylor decomposition becomes less accurate with additional layers due to the difficulty in selecting a suitable root point, and performing it on sub-functions alleviates this issue and as such, DTD creates a more focused explanation.

4 VADR: Video-Audio Discriminative Relevance

VADR is a technique that provides a greater understanding of input modality contributions and temporal and non-temporal relevance to audio and video models. Explanations produced are in the form of saliency maps like those of DTD. This was demonstrated in Taylor et al. in press 2020 on a mid-fusion CNN performing action recognition. The saliency explanations however separate modality, temporal and non-temporal relevance from one another providing insight into features a model has learned and as a possible use case, focusing the relevance for a human to quickly identify the important part of an input. VADR accepts a video input consisting of frames over time which is stacked to create a tensor of $frames \times height \times width \times channels$, and audio transformed into a spectrogram. Temporal relevance will be removed from all other relevance which results in the separation of these two relevance types. A video consists of a number of frames and audio plotted with changes in frequency over time, this separation of temporal relevance is possible.

4.1 Proportion of Relevance

The relevance proportion of each individual modality can be worked out with equation 1. This will assume relevance is defined as the constant C which comes from the gradient of the input (derived from loss), multiplied by the input: $R = c \times x$. As modalities may be of different sizes, VADR will account for any unbalance by weighting the proportion relevance by the size of the modality feature vectors. The input feature vectors $X_{V,A}$ are given as X_S.

$$\sum R_{S_{weighted}} = \frac{\sum R_S}{\sum\limits_{S \in V,A} C_S} \tag{1}$$

where

$$C_S = \frac{\sum R_S}{\sum X_S} \tag{2}$$

The purpose of this formula is to provide insight into a model's bias towards input modality. This formula output is also used for *selective relevance* which is what provides the saliency maps used in VADR explanations. Equation 1 can find out relevance on a per sample, class, or dataset basis.

The authors of VADR demonstrated a proportion of relevance with a two-stream mid-fusion action recognition classifier on a subset of UCF-101[1]. The input streams were for video and audio. Here they showed audio and video were both relevant, with one or the other modality becoming more relevant depending on the class. In classes such as Hammering where audio is expected to be highly related to the task, audio relevance generally was higher than video relevance. This was then switched for classes like ApplyEyeMakeup where visual information is more descriptive than audio information.

4.2 Selective Relevance

The relevance maps that makeup VADR explanations are produced using a technique called Selective Relevance (SR) which was introduced in (Hiley et al. 2020) as the solution to previous XAI techniques such as DTD being unable to separate different types of relevance (Hiley et al. 2019). DTD and other XAI techniques, for example, worked well on images, but when given a video would combine spatial and temporal relevance, making it hard to identify exactly what was important. SR builds on other relevance methods such as DTD, but with the addition of separating relevance over time. This is achieved by taking the derivative of the relevance in each dimension. For video, this would be over the x, y, and time dimensions. Where the derivative changes dramatically over time, the relevance is temporal. When removed from the total relevance, temporal and non-temporal relevance has been separated.

A technique that SR uses is to use a Sobel operator, which is a handcrafted convolution, that will detect edges in a selected dimension when passed over an image. This will output a grey-scale representation of the edges. If instead of passing the Sobel operator h'_t over an image, it is instead passed over the relevance map R, the result will be a pixel-wise derivative G_t with which, non-temporal relevance can be filtered from temporal relevance by applying n standard deviations represented by σ (Taylor et al. in press 2020).

$$G_t(R) = h'_t * R \tag{3}$$

$$R_t = \{r_{ijk} | G_t(r_{ijk}) > \sigma\} \tag{4}$$

The n standard deviations σ to separate temporal and non-temporal relevance are user selected. When applied to the temporal edge map, any value which is greater than σ is set to 1 and 0 otherwise. This method can be applied to both 3D inputs (video) and 2D inputs (audio) as its only constraint is the size of the dimension remains the same.

5 Implementation

For this paper, I have implemented an action recognition classifier which was achieved in two stages. The first stage was to create a feature extractor heavily based on (Korbar, Tran, and Torresani 2018). This is a two-stream network with a subnetwork for the audio and video modality. Each of the subnetworks, once trained, was loaded into the second stage of the model which transformed them into the classifier. This is detailed more in section 5.2.

The implementation of the classifier was based on the implementation in (Taylor et al. in press 2020). It shares code for the dataset, video and audio transformations, and dataset processing method. The shared code has been modified to allow the differences that this paper exhibit in comparison to (Taylor et al. in press 2020).

5.1 Feature extraction

The authors of Korbar, Tran, and Torresani 2018 created a model to learn the connection between audio and video from self-supervised learning. They aimed to create a binary classifier for temporal synchronisation between audio and video but it was also adapted for action recognition, where it was discovered that their model also acted as a suitable feature extractor for such tasks, offering an improvement in accuracy compared to training from scratch. As this paper uses a model for action recognition using audio and video inputs, recreating this work would provide a suitable classifier for XAI to be added and analysed.

5.1.1 Dataset

For the feature extractor to learn an accurate representation of the dataset which would not be easily misled by distortion in future examples, data was augmented before feeding it into the feature extractor. This has two main benefits as it helps stop a model from overfitting to just the data it was trained on and it artificially increases the size of the dataset. The argumentation methods applied to audio and video were the same as (Köpüklü et al. 2019a) and (Hershey et al. 2017) which was also used in (Taylor et al. in press 2020). This randomly flipped the input video, performed a random crop, and normalised the video. The audio was transformed into a Mel-spectrogram with log-scaling. An additional modification to the

[1]UCF-101 is a dataset of 101 different activity categories taken from YouTube. See `https://www.crcv.ucf.edu/data/UCF101.php`.

input data converted the video to 16 frames for every second at a resolution of 320 pixels by 240 pixels with audio set to 16k samples per second. These two modifications to the data were made to transform the data into a consistent format that could easily be passed through the feature extractor. The feature extractor was trained on 1 second long clips.

As detailed in (Korbar, Tran, and Torresani 2018), the feature extractor is trained on different difficulties of samples in a method called curriculum learning, full details of which are in section 5.1.3. This capability required the dataset to be configured to not only fetch data for training but also adapt the data retrieved depending on the situation. A particular sample of data can either be *positive*, *easy* or *hard* where easy and hard samples are both *negative*. Positive samples have in sync audio and video, easy samples take audio and video from different clips and hard examples have audio and video from the same clip although there is at least half a second gap between the audio and video.

5.1.2 Architecture

The feature extractor uses two sub NN using Mobilenet V1 architecture (Howard et al. 2017) which, as much as it is not state-of-the-art in terms of performance, has fewer parameters to train, and thus converges faster and can run on less powerful devices. The complete architecture is detailed in figure 1. The loss function used to create the feature extractor is contrastive loss which is distance-based. This works by the output of each sub NN feeding into this function with the value of the loss denoting how different they are from each other. Contrastive loss makes it possible to produce two networks with similar outputs, with the benefit to this feature extractor is being able to match audio and video for the same action. This loss function is detailed in equation 5 where $Ytrue$ is 1 if the sample is positive else 0 and D is the Euclidean norm. For training, the margin was set to 0.99. The loss, like a standard NN, is backpropagated, although this time through each of the sub-NN.

$$contrastiveLoss = Ytrue \times D^2 + (1 - Ytrue) \times max(margin - D, 0) \quad (5)$$

$$Ytrue = \begin{cases} 1 & \text{if sample is positive} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

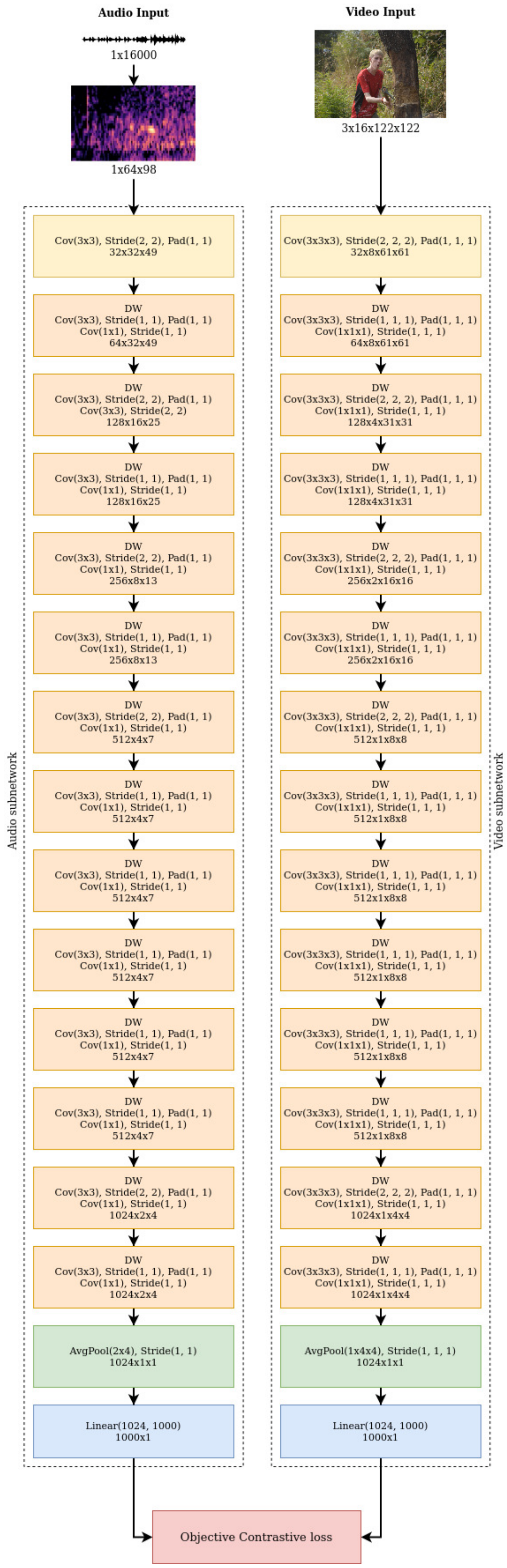

Figure 1: Two Mobilenet V1 sub networks with contrastive loss

5.1.3 Training

The feature extractor was trained on a subset of UCF-101 with 51 classes which featured both audio and video modalities. The dataset was split into train, validation, and test subsets with the ratio 70:20:10 respectively. Only train and validation was used during training of the feature extractor, the validation subset of the dataset was not used.

As the authors of (Korbar, Tran, and Torresani 2018), the feature extractor trained used curriculum learning which is to say training started with an easier task before moving on to a harder one. In this case, the feature extractor started training with only positive and easy samples for a set number of epochs before adding in hard samples. The training ran for a total of 130 epochs with the first 50 epochs only using positive and easy samples with a 50:50 probability of each sample type being used. After this point, hard samples were added with a probability of 50:25:25 for positive, easy, and hard samples respectively.

The learning rate was set to 0.01 and reduced by a factor of 10 on the 40th, 55th, 65th, and 70th step. Weight decay was set to 0.9, 0.9, and 0.1, respectively, and batches were made up of 16 samples. The model was implemented in Pytorch[2] and trained on a single NVIDIA 1080ti GPU. Training took 4 hours and 48 minutes with a final validation loss of 0.2919. At this point, the loss does not equate to much, but during training, it should be noted that the loss over time decreased for both the train dataset split and the test dataset split which is a good sign the feature extractor has not overfit.

5.2 Action recognition

For action recognition, a total of two classifiers were trained. Section 5.2.1 to 6 discusses the second of the two classifiers as the first classifier heavily overfit on the dataset. This resulted in a seemingly good accuracy of 89% but as all similar models (Korbar, Tran, and Torresani 2018, Taylor et al. in press 2020) achieved a lower accuracy while making use of a larger dataset for the feature extractor, it became clear the classifier had picked up on some feature(s) that inflated its accuracy. A NN overfits because the model discovers additional features than are required for it to function correctly Hawkins 2004. This could be the case by including poor training data such as including a ruler in all positive example images for medical diagnosis while negative samples never include a ruler, or overfitting could be attributed to the training method used. Between my two classifiers, the change was with the size of each subset of the dataset. The first classifier mistakenly used a train:validation:test split of 70:10:20 instead of the corrected split of 70:20:10.

5.2.1 Model

The action recognition classifier uses the same mobilenet V1 architectures as trained in section 5.1 for audio and video. The final layer of each sub-network overwise know as a *feature vector* were concatenated together with an additional fully connected layer added after to make it a mid-fusion model. The fully connected layer was of size 51 to match the number of action classes in the dataset. Each feature extractor feature vector size is 1000 which makes it possible to evaluate the relevance of each input modality with a balanced network. It was found that an unbalance did not seem to cause an issue in (Taylor et al. in press 2020) and with my classifier architecture, it will be possible to verify that. The classifier architecture can be seen in figure 2.

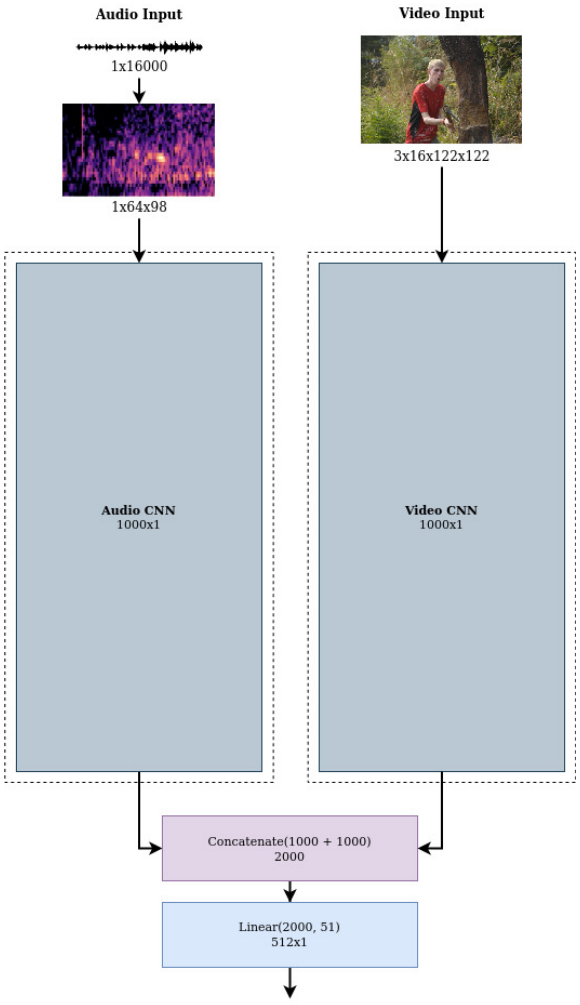

Figure 2: The model uses two pre-trained MobileNet V1 architectures as feature extractors with the output concatenated and passed through one additional fully connected layer.

[2]Pytorch is an ML framework for rapid development in Python. See `https://pytorch.org/`.

5.2.2 Training

For training, the classifier was fine-tuned with the same subset of UCF-101 as the feature extractor. Each of the subnets weights was loaded in from pre-trained models, but due to the small size of UCF-101, the weights for the video stream were not suitable and would result in the model achieving a maximum of around 6% accuracy. Instead, weights from a pre-trained Mobilenet V1 classifier trained on Kinetics[3] was imported from Köpüklü et al. 2019b. The audio stream weights were still suitable from the feature extractor that was trained in section 5.1, although this does have a negative impact on the overall accuracy. The subnet weights were frozen during training for action recognition with the exception of the feature vectors and the additional fully connected layer after fusion. Fine-tuning used cross-entropy loss and stochastic gradient descent.

The learning parameters for this stage of training remained mostly the same as the feature extractor with the exception of the learning rate being set to 0.1, and decay steps at the 55th, 80th, 90th, 100th, 110th, 120th and 130th epoch. The batch size was set to 32 and training ran for 140 epochs. This achieved an accuracy of 49.94% on the test hold-out dataset split which, although it is lower than other research has achieved, still demonstrates it has learned representation of the data as random guessing would reach an accuracy of around 1.9%. Every sample the model used was positive.

6 Results

The overall accuracy the classifier achieved in comparison to other researcher papers is shown in table 1. The accuracy comparison is made between a selection of NN that have different architectures trained to perform action recognition with UCF-101. All of the NN includes the fusion of video and audio. As mentioned previously, the classifier detailed in this paper is not expected to reach state-of-the-art performance. Compared with other models that used fusion for visual and audio information on UCF-101, the classifier achieves a lower level of accuracy. This can be attributed to the training of the feature extractor on UCF-101 as the audio subnetwork did not contain a diverse set of features. This particular point is more clearly seen when applying VADR to the results. I have also included a larger NN in table 1. This was trained using visual and optical-flow streams and demonstrates the accuracy expected with larger models.

Table 1: Method comparison

Method	Accuracy
AVTS (Korbar, Tran, and Torresani 2018)	87%
MobileNet + VGGish (Taylor et al. in press 2020)	81.5%
Two-Stream I3D (Carreira and Zisserman 2017)	93.4%
Two-Stream MobileNet	**49.94%**

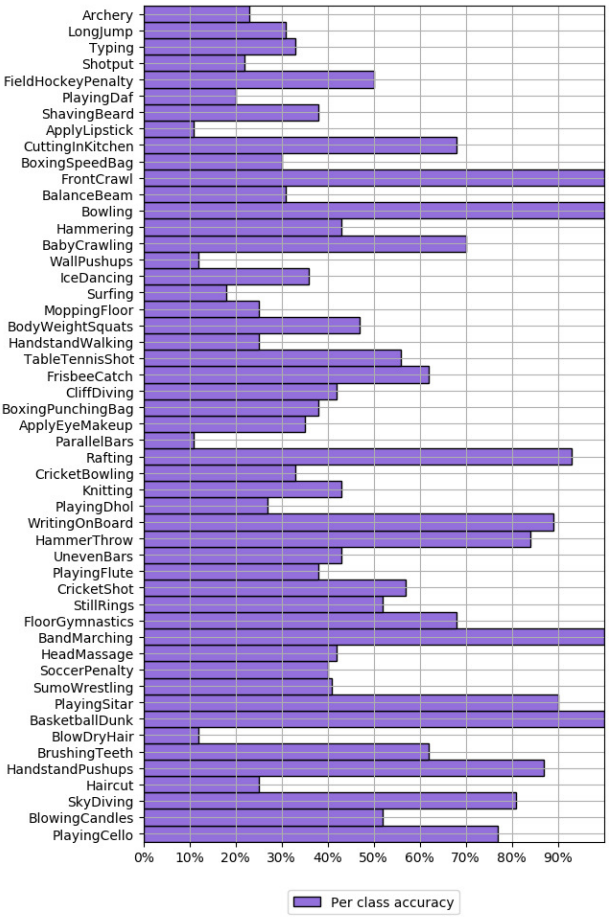

Figure 3: Per class accuracy on subset of UCF-101

Out of the 51 classes, the classifier performs best on BandMarching, BasketballDunk, FrontCrawl, Bowling, rafting, and PlayingSitar with all but PlayingSitar and Rafting achieving 100%. PlayingSitar and Rafting were both above 90%. All of these classes' accuracy can be attributed to the training data either only containing samples from a particular direction and/or the class visuals being distinct compared to other classes. In comparison, ApplyingLipstick and ApplyEyeMakeup both had similar visual streams and

[3]Kinetics is a dataset of 600 different activity categories each with at leaset 100 samples. See `https://deepmind.com/research/open-source/kinetics`.

thus their accuracy was 11% and 35% respectively. A full breakdown of per-class accuracy can be found in figure 3.

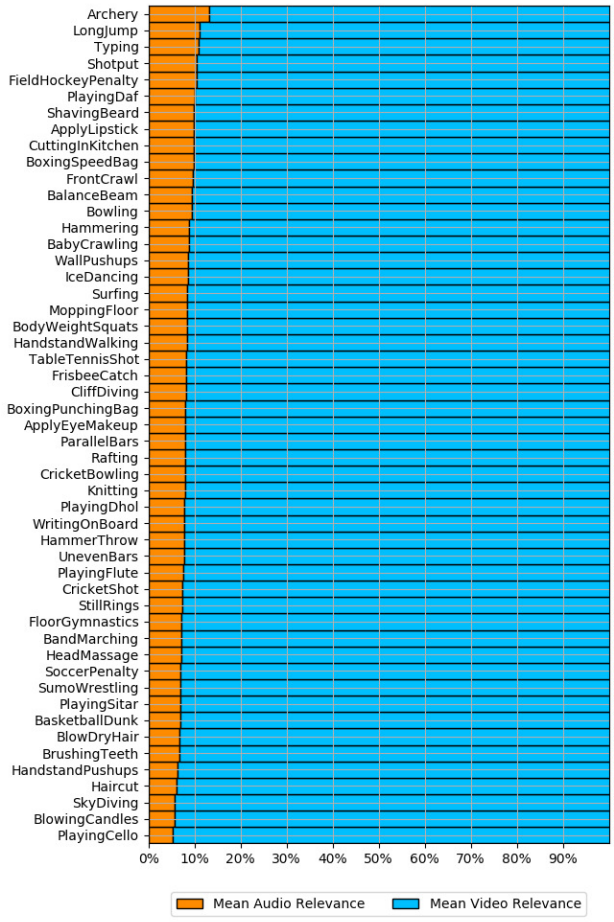

Figure 4: Mean modality relevance for our model on subset of UCF-101

For further analysis of the classifier, an additional technique called VADR explanations was added. I have used a modified implementation of VADR from (Taylor et al. in press 2020) for this section. Figure 6 shows the class-wise relevance between audio and video. This was calculated with equation 1 to get the proportions of audio to video in regards to the contribution to the prediction. From this, note the relevance is very much dependant on the video stream and in contrast, the authors of VADR found much higher audio relevance in their model. The different results I found were due to the audio feature extractor. As it was trained on a very small dataset, the features it represents were not diverse which resulted in several classes that a human would expect to be audio strong ending up reversed. The strongest cases of these include classes with instrument playing. If the classifier model was retrained using an audio stream trained on a larger dataset such as Kinetics, the audio stream would be expected to shift closer to results the authors of VADR achieved and an overall accuracy similar to (Korbar, Tran, and Torresani 2018), depending on the model architecture.

Despite the shift in video relevance, the classifier does seem to be starting to use the audio stream on action classes with more distinct sounds, such as CuttingInKitchen, Bowling, and FrontCrawl while classes such as SkyDiving are on the lower end of audio relevance due to irrelevant or noisy audio. In the case of Skydiving, the audio in most of the video samples have music as the audio track.

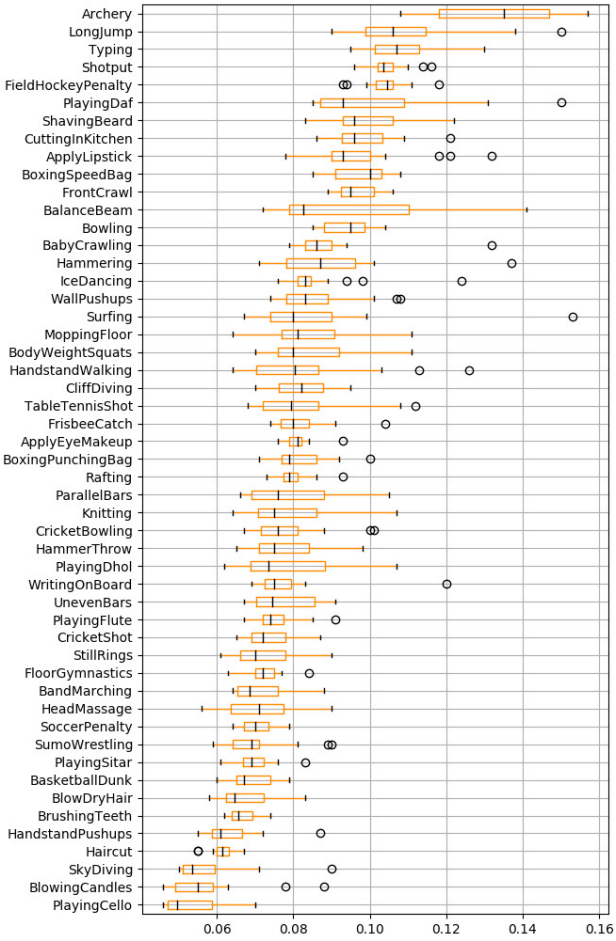

Figure 5: Boxplot audio modality relevance per class on the subset of UCF-101

Exploring audio relevance further we can start to analyse why some classes have higher audio relevance than others. Figure 5 shows us a bit more about the distribution of audio relevance per class. Video relevance has not been displayed as it is just the inverse of the audio relevance. From this boxplot, we can see a general trend towards a larger range of maximum and minimum audio relevance as class accuracy decreases when referring back to figure 3. The average accuracy of classes under 8% audio relevance is 58.4%, between 8% and 9.5% relevance is 44.6% and relevance above 9.5% has an average accuracy of 38.1%. It should be understood, the correlation between audio relevance and class accuracy does not mean causation. As the classifier struggles to identify features using the more relevant video feature extractor, it appears to be rely-

ing more on the weaker audio feature extractor and it would appear as if this often does not benefit the classifier's prediction. This is an artifact of using a feature extractor trained on UCF-101. It is not possible to draw any concrete, generalised, audio vs video effect on overall accuracy from this classifier. Referring back to the authors of VADR, their model showed a different story where audio was more important that video for approximately half of the classes.

Applying VADR explanations to the classifier allows us to explore the features that impact the prediction most. On average the classifier seems to use temporal and spatial information equally from video information as the value of σ was set at 2.0. Audio appears to be more spectral with a σ value of 0.25 but considering the low relevance of this modality, with the classifier, it is difficult to tell if this is indeed the case. The temporal and non-temporal results were unexpected in comparison to the VADR paper as the author's results shown much higher spatial relevance in the visual modality and higher temporal in the audio modality. Both the classifier detailed in this paper and the VADR paper uses the same pre-trained video subnet and thus you would expect very similar, if not the same relevance for that modality. The primary difference in the visual modality is very selective spatial relevance. The authors of VADR found almost all spatial information relevant to the reduction of temporal relevance. Temporal relevance as much as it was not as important, proved useful in terms of an explanation as it successfully would highlight the movement of the subject such as a boxing swing, or a diver jumping off a cliff.

In comparison figure 6 and figure 7 shows the relevance with the classifier demonstrated in this paper. In figure 6, the temporal relevance is primarily over the diver as he jumps and falls with spatial relevance highlighting the shape of the diver, the diver's feet, and shorts. Highlighting feet and shorts goes to show the model has picked up features that relate to a few classes in the dataset, but not necessarily a feature a human would identify as relevant. For this particular example, audio shows little to no audio features that would be useful for a human. This is expected, as over this particular example there is music playing as the audio track.

In figure 7, video temporal relevance highlights movement of the person's arm as highly relevant, in addition to the hammer and the piece of wood being hit. Spatial relevance, similarly to the diver, displays the shape of the subject being relevant with the arm, hammer, and piece of wood as most relevant in this example. Audio relevance is interestingly picking up the impact of the hammer which paired with the fact hammering, as a class, gets reasonably high accuracy with higher audio relevance, it would suggest the model has started to pick up the impact sound as a feature.

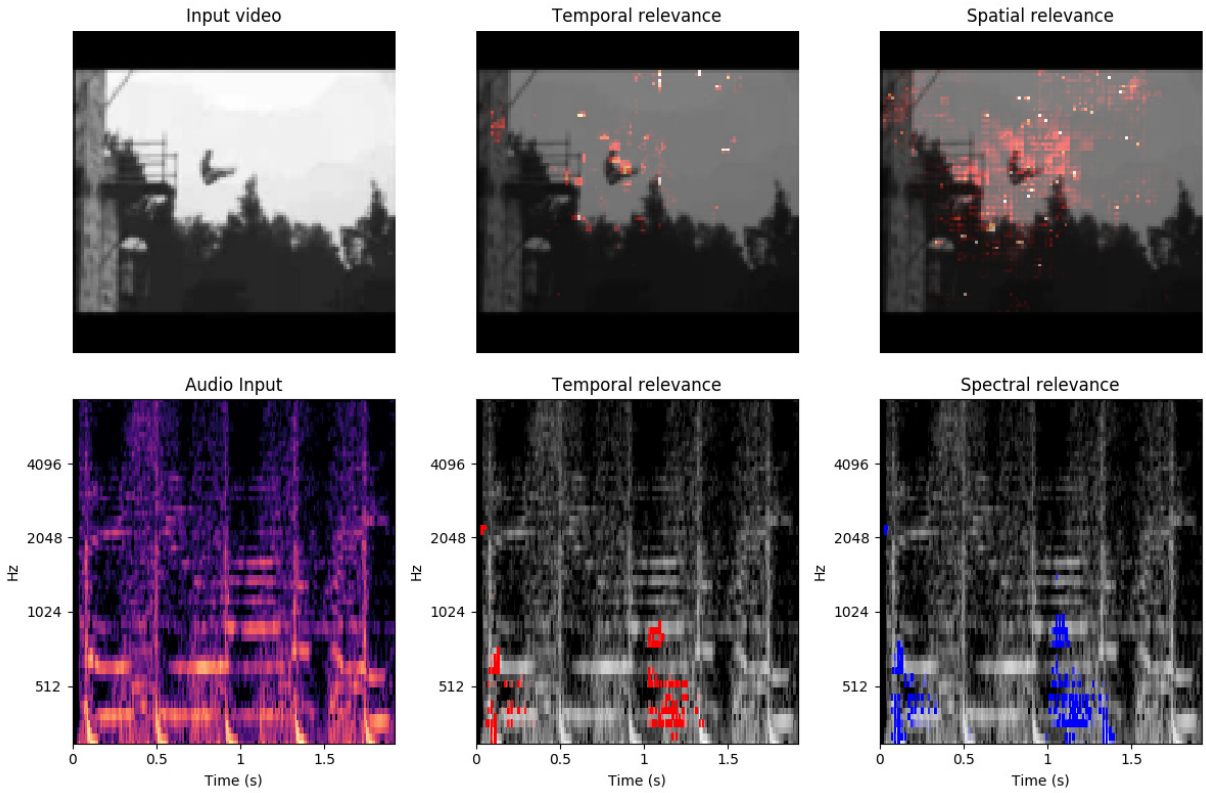

Figure 6: Cliff Diving with VADR explanation. The selected frame is from 0.6 seconds in from the start of the sample.

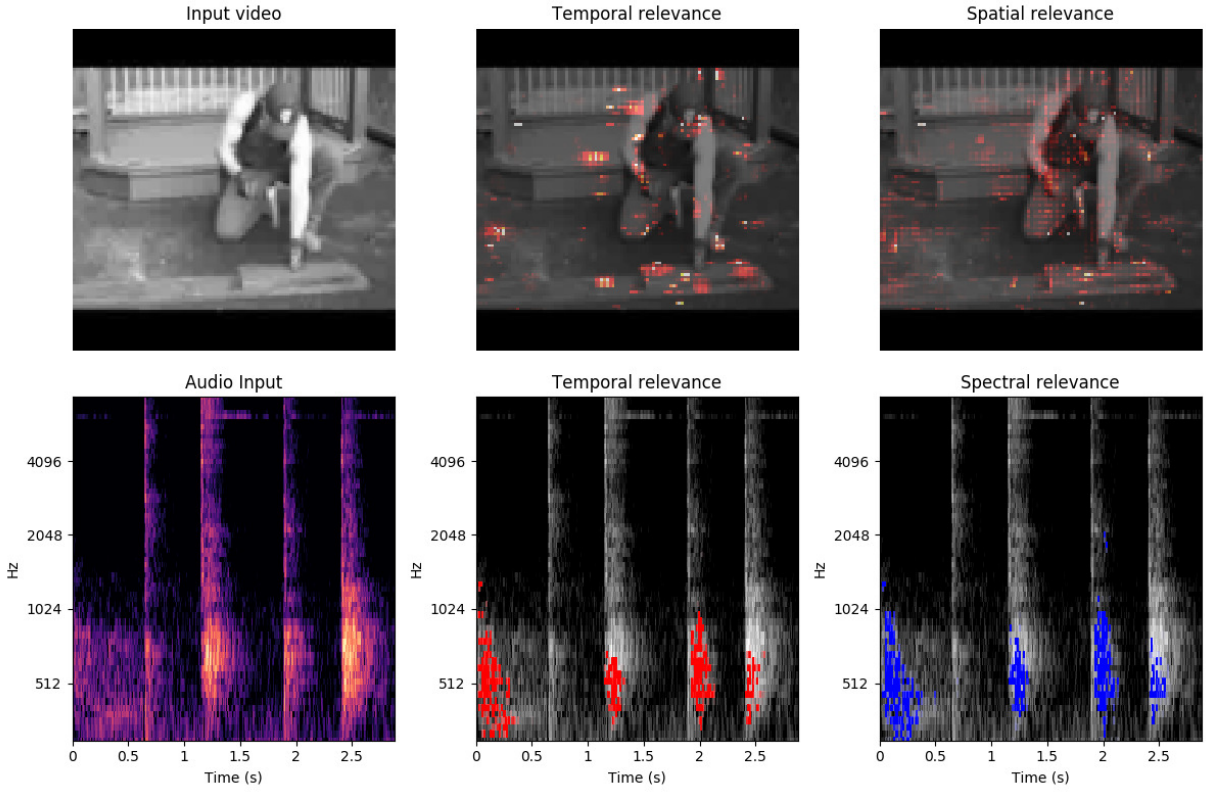

Figure 7: Hammering with VADR explanation. The selected frame is from 0.6 seconds in from the start of the sample.

7 Conclusions and future work

The goal of this paper was to explore XAI for Multi-Sensor Information Fusion. The model created was a classifier on a subset of the UCF-101 dataset using the 51 classes that contained video and audio. The classifier trained successfully converged on a solution, able to achieve a reasonable, although on the lower end of the spectrum, level of accuracy using audio and video streams for input. The classifier used mid-fusion to combine the modalities and each of the subnetworks is efficient in terms of required compute which will enable the classifier to run on less powerful hardware. If the model was to be retrained on a larger dataset, this project has provided the required code to begin the process.

VADR, a recently introduced method to provide reasoning for both temporal and non-temporal explanations was applied to the classifier, which allowed us to explore the features the model had discovered. This classifier primarily used video as the most relevant modality and then temporal and non-temporal information, approximately, equally.

There are still areas for future work which are outlined in the following:

1. As discussed in section 6, audio relevance is much lower than desired due to training the feature extractor on UCF-101. The aim of the feature extractor for this project was to improve the overall accuracy of the classifier. This was designed to use transfer learning in which a larger, but a similar task is originally learned before being mapped over to the final task (Raina, Ng, and Koller 2006). Producing a new feature extractor, trained on a larger dataset would result in a classifier with higher accuracy, and with much more conclusive results can be made. The work in this paper sets a functional codebase for this and Kinetics would be a suitable dataset for training.

2. Further development to optimise the implementation for training would offer a few speedups when creating the feature extractor and classifier. At this time, training can only make use of a single GPU and has a slow pipeline for providing data to the GPU. Adding multi GPU support, in theory, could yield between 1 and 2 times speedup for training, and improving the data pipeline would mean the GPUs are not left idling for long periods of time. This work would not affect the overall outcome but will speed up getting there and avoid the approximate of 30 days to train the feature extractor with the current configuration.

3. We know that modalities are related to each other with one of the earliest works in the area looking at audio-visual information is speech Mcgurk and Macdonald 1976. Therefore if one or more modalities are missing, then the remaining modality or modalities should still be used to give a good prediction. Further research looking at the relevance shift if one modality is missing or noise added would provide insight into model resilience and how relevance changes to new, challenging situations.

4. Creating a live demo of the classifier with VADR would make the research more accessible to people without a technical background in the subject area. This could also demonstrate the lower compute requirements if the demo was performed on a device such as an NVIDIA Jetson Nano[4].

8 Reflection

Generally, this project has been a success. Section 5 has created the groundwork for a more robust classifier to be trained in the future, and have added explanations using VADR in section 6 which assists in human understanding. That said, The classifier I produced did not use a robust feature extractor which would have lead to greater accuracy with the classifier and VADR explanations.

The feature extractor detailed in section 5.1 was originally trained on Kinetics600 in an earlier iteration, but unfortunately during development a few of the design choices I made caused issues, and as such, I was unable to get the model to converge. The largest of these was in the transformations of the training data and by the time I had realised there was an issue, there was no longer enough time to make a correction and start training again. The training was set to last 90 epochs, and on the hardware configuration I had access to, this would take approximately 30 days to complete.

Adding VADR to my model's output turned out to be a larger task than I initially envisioned. I had access to and used a pre-existing implementation of VADR which, without, I would have never been able to complete myself before the deadline. I still faced a major challenge when adding this to my classifier as the output originally to myself looked correct, but in comparison to the original work for VADR, contrasted significantly. After a deeper look into the difference,

[4] A single board computer capable of running NN. See `https://developer.nvidia.com/embedded/jetson-nano-developer-kit`.

no issue was raised with the VADR implementation and thus the difference was down to different features the two classifiers had learned.

Going into this project I already had some experience in ML and in particular in Reinforcement Learning where an agent learns from experiencing an environment over and over again, maximising some reward function. The switch to MML and XAI, all within a framework I had not used before (my previous experience had been in an alternative to Pytorch), resulting in having to rapidly learn new skills and read up on some recommended literature. This experience was a challenge at times, but a very useful extension to my knowledge in the field.

Overall I am pleased with what I have produced throughout this project. I have made significant progress despite only reaching the initial goal in comparison to my initial plan. The target goal was to make an addition to explainability for multimodal data fusion which would have likely taken the form of a second model. The work required to reach the initial goal turned out to be more challenging than initially thought and so it became the entire project. If I was to also work on the target goal, then I would have ended up with two semi-formed solutions.

9 Acknowledgements

During this project, I would like to thank my supervisor Prof Alun Preece and two PhD students Harri Taylor and Liam Hiley for their continued support throughout. They have all kept me pointed in the right direction, giving me hints of what to try next and useful research papers to read. It's safe to say that this project would have not turned out the way it did without them.

This work was presented to a group of researches at Cardiff University in early May, who's questions on the research helped to fill in a few gaps and make the results clearer.

References

Arrieta, Alejandro Barredo et al. (2019). *Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI*. arXiv: 1910.10045 [cs.AI].

Bach, Sebastian et al. (2015). "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation". In: *PLOS ONE* 10.7, pp. 1–46. DOI: 10.1371/journal.pone.0130140. URL: https://doi.org/10.1371/journal.pone.0130140.

Baltrušaitis, T., C. Ahuja, and L. Morency (2019). "Multimodal Machine Learning: A Survey and Taxonomy". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.2, pp. 423–443. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2018.2798607.

Benitez, J. M., J. L. Castro, and I. Requena (1997). "Are artificial neural networks black boxes?" In: *IEEE Transactions on Neural Networks* 8.5, pp. 1156–1164.

Carreira, Joao and Andrew Zisserman (2017). *Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset*. arXiv: 1705.07750 [cs.CV].

D'mello, Sidney K. and Jacqueline Kory (2015). "A Review and Meta-Analysis of Multimodal Affect Detection Systems". In: *ACM Comput. Surv.* 47.3. ISSN: 0360-0300. DOI: 10.1145/2682899. URL: https://doi.org/10.1145/2682899.

Ebersbach, Mike, Robert Herms, and Maximilian Eibl (2017). "Fusion Methods for ICD10 Code Classification of Death Certificates in Multilingual Corpora". In:

Hall, D. L. and J. Llinas (1997). "An introduction to multisensor data fusion". In: *Proceedings of the IEEE* 85.1, pp. 6–23.

Hawkins, Douglas M. (2004). "The Problem of Overfitting". In: *Journal of Chemical Information and Computer Sciences* 44.1. PMID: 14741005, pp. 1–12. DOI: 10.1021/ci0342472. eprint: https://doi.org/10.1021/ci0342472. URL: https://doi.org/10.1021/ci0342472.

Hershey, Shawn et al. (2017). "CNN Architectures for Large-Scale Audio Classification". In: *2017 Ieee International Conference on Acoustics, Speech and Signal Processing (Icassp)*. IEEE, pp. 131–135.

Hertel, L. et al. (2015). "Deep convolutional neural networks as generic feature extractors". In: *2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–4.

Hiley, Liam et al. (2019). *Discriminating Spatial and Temporal Relevance in Deep Taylor Decompositions for Explainable Activity Recognition*. arXiv: 1908.01536 [cs.LG].

Hiley, Liam et al. (2020). *Explaining Motion Relevance for Activity Recognition in Video Deep Learning Models*. arXiv: 2003.14285 [cs.LG].

Howard, Andrew G. et al. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. arXiv: 1704.04861 [cs.CV].

Köpüklü, Okan et al. (2019a). "Resource Efficient 3D Convolutional Neural Networks". In: *arXiv preprint arXiv:1904.02422*.

— (2019b). "Resource Efficient 3D Convolutional Neural Networks". In: *arXiv preprint arXiv:1904.02422*.

Korbar, Bruno, Du Tran, and Lorenzo Torresani (2018). *Cooperative Learning of Audio and Video Models from Self-Supervised Synchronization*. arXiv: 1807.00230 [cs.CV].

Mcgurk, Harry and John Macdonald (1976). "Hearing lips and seeing voices". In: vol. 264, 746–748. DOI: 10.1038/264746a0. URL: https://doi.org/10.1038/264746a0.

Montavon, Grégoire et al. (2017). "Explaining nonlinear classification decisions with deep Taylor decomposition". In: *Pattern Recognition* 65, 211–222. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2016.11.008. URL: http://dx.doi.org/10.1016/j.patcog.2016.11.008.

Potamianos, Gerasimos et al. (2003). "Recent advances in the automatic recognition of audiovisual speech". In: *Proceedings of the IEEE* 91, pp. 1306 –1326. DOI: 10.1109/JPROC.2003.817150.

Raina, Rajat, Andrew Y. Ng, and Daphne Koller (2006). "Constructing Informative Priors Using Transfer Learning". In: *Proceedings of the 23rd International Conference on Machine Learning*. ICML '06. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 713–720. ISBN: 1595933832. DOI: 10.1145/1143844.1143934. URL: https://doi.org/10.1145/1143844.1143934.

Ren, Shaoqing et al. (2015). *Object Detection Networks on Convolutional Feature Maps*. arXiv: 1504.06066 [cs.CV].

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016). ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: Association for Computing Machinery, 1135–1144. ISBN: 9781450342322. DOI: 10.1145/2939672.2939778. URL: https://doi.org/10.1145/2939672.2939778.

Roitberg, Alina et al. (2019). "Analysis of Deep Fusion Strategies for Multi-Modal Gesture Recognition". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Snoek, Cees G. M., Marcel Worring, and Arnold W. M. Smeulders (2005). "Early versus Late Fusion in Semantic Video Analysis". In: *Proceedings of the 13th Annual ACM International Conference on Multimedia*. MULTIMEDIA '05. Hilton, Singapore: Association for Computing Machinery, 399–402. ISBN: 1595930442. DOI: 10.1145/1101149.1101236. URL: https://doi.org/10.1145/1101149.1101236.

Taylor, Harrison et al. (in press 2020). *VADR: Discriminative Multimodal Explanations for Situational Understanding*. Note: Available in submission archive.