FACULTAT D'INFORMÀTICA DE BARCELONA (FIB)
FACULTAT DE MATEMÀTIQUES (UB)
ESCOLA TÈCNICA SUPERIOR D'ENGINYERIA (URV)

Personality Regression from Multimodal Dyadic Data

MASTER IN ARTIFICIAL INTELLIGENCE

Author:
David Curto Janó

Supervisor:
Sergio Escalera Guerrero

Co-Supervisor:
Albert Clapés i Sintes
Sorina Smeureanu

Barcelona, 21 June 2021

Abstract

Personality is made up of broad traits that are relatively stable over time and allow to differentiate one person from another. The most widely accepted theory to model personality is the Big-Five model that defines the traits as a spectrum, allowing to rank and measure differences between individual's personality.

Humans infer personality by observing different verbal and non-verbal cues. We are able to infer the personality of others through the observation of different modalities, capturing patterns from speech, body gestures, facial expressions, among others.

This Master's thesis proposes a multimodal model that extracts audiovisual features using state-of-the-art methods to infer the personality of a target person in a dyadic scenario. The model is trained on the UDIVA dataset [1], a multimodal dataset of non-scripted face-to-face dyadic interactions based on free and structured tasks that elicit different behavior and cognitive workload in the participants. All sessions are conducted in a controlled environment and the personality of the participants is obtained through self-reported assessments.

We investigate the effect of the audio and video modalities for the recognition of the personality separately but also jointly, analyzing the general performance, by session, participant and by task. Furthermore, we also evaluate the effect of adding a larger range of visual and acoustic cues before producing the prediction regarding the performance of the model.

The results from an incremental study show that the performance of the model is improved when combining long-range visual and acoustic features. Showing significant improvements in most metrics compared to the performance of the previous state-of-the-art model [1]. The results are very promising considering that our model has been trained with a smaller part of the data set, fewer modalities and in a multi-task manner (a single model for all tasks).

Acknowledgements

First of all, I would like to express my gratitude to my supervisor, Sergio Escalera, and my co-supervisors, Albert Clapés and Sorina Smeureanu, for their guidance throughout this project. Their supervision, advice and insights have been key to the success of this thesis.

I would also like to thank the Computer Vision Center and the Human Pose Recovery and Behavior Analysis (HuPBA) research group for granting me access to the UDIVA dataset and their computational resources to run all the experiments.

I would also like to thank all my classmates for sharing their knowledge and for always being willing to help out. Without them, I am sure that the experience of this master would not have been the same.

Finally, I would like to thank my family and friends for being my main and unconditional support, especially during the last year that has been very demanding due to the Covid-19 situation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Human personality has long been one of the most studied topics by psychologists. Various theories have been proposed over the years, being the **trait theory** [2] the most widely adopted theory for describing an individual's personality. The trait theory focuses on the idea that personality is made up of broad traits that are **relatively stable over time** and allow to differentiate one person from another.

During the last decades, different trait theories [3, 4, 5] appeared but most of them failed to describe in a clear and concise way the traits to define and assess the personality of humans. However, the theory that began to be adopted was the five-factor theory, started by D.W. Fiske in 1949 and evolved over the last years to what is now known as the **Big-Five** model.

The Big Five model, often referred to by its acronym OCEAN, is one of the most widely accepted and recognized models to describe the personality traits of a person. The principal characteristic of this model is that it defines the **traits as a spectrum** rather than binary categories, thus allowing to rank and measure differences between individual's personality. As its own name indicates the Big Five model defines five different personality traits which are:

- **Openness.** Strongly related to imagination, creativity and curiosity[6]. People who are high in this trait tend to be more adventurous and are open to trying new things whereas people who are low in this trait are more traditional and they may struggle with creative activities.
- **Conscientiousness.** Strongly related to being disciplined, have good impulse control and goal-oriented [6]. People who are high in this trait

tend to be organized and plan ahead all activities. On the other hand, people low in this trait dislike making plans, procrastinate on important tasks, or even fail to complete them.

- **Extraversion.** Strongly related to assertiveness, sociability and high amount of emotional expressiveness [6]. People who are high in this trait like to start conversations and are very friendly. On the contrary, people low in this trait prefer solitude and they struggle to start conversations.
- **Agreeableness.** Strongly related to concepts like kindness, trust, altruism, and affection [6]. People who are high in this trait enjoy helping others, they are empathic with others' problems (high prosocial behaviors). On the contrary, people low in this trait tend to be more competitive, manipulative and they have a lack of sympathy.
- **Neuroticism.** Strongly related to emotional instability and sadness [6]. People who are high in this trait tend to experience stress, anxiety, and mood swings. On the other hand, people low in this trait tend to show a more resilient attitude towards stress and mood changes.

In recent years, interest in an emerging field such as automatic personality recognition has grown considerably. In this field, machine learning models try to recognize the recurring patterns of individuals in order to predict personality, which usually is characterized by the Big-Five traits based on self-reported assessments.

Advances in computer vision and pattern recognition models have opened the door to build deep learning models that can successfully recognize verbal and non-verbal cues and infer the personality trait scores from videos [7, 8].

1.2 Proposal

This Master's thesis presents our multimodal model that extracts audiovisual features using state-of-the-art methods to infer the personality of a target person in a dyadic scenario. The model is trained on the UDIVA dataset [1], a multimodal dataset of face-to-face dyadic interactions recorded in different contexts, each of them related to a different collaborative/competitive task.

The aim of this thesis is to investigate the effect of audio and visual modalities in personality recognition as well as the effect of adding a larger range of visual and acoustic cues before producing the prediction.

Hereunder we list the hypothesis that we intend to verify in this thesis:

- The multimodal model can benefit from using a larger range of visual and acoustic cues before producing the prediction.
- The combination of audio and visual modalities can lead to a stronger personality regressor.

1.3 Overview

In this first chapter, we introduced what is the problem and an overview of our work.

In Chapter 2, we cover the theoretical background required to properly understand our work.

In Chapter 3, we present the related work, doing an extensive analysis of the current state-of-the-art methods in personality recognition.

In Chapter 4, we introduce the dataset used to train our models, the preprocessing steps that have been carried out and we explain the dataset split strategy used to divide the dataset into the train, validation, and test sets.

In Chapter 5, we describe an overview of our multimodal model and then we start describing each modality model. Lastly, we describe our multimodal model with fine-grained details.

In Chapter 6, we present the experimental setup and the training and evaluation strategy. After that, we describe the experiments we have conducted to later do an extensive analysis of their results with a proper discussion.

In Chapter 7, we arrive at conclusions and explain the most relevant points that we extracted from this thesis.

Finally, in Chapter 8, we propose the most relevant future steps to improve our model.

Chapter 2

Background

This chapter covers all the theoretical background concepts that were used to do this project. The concepts are explained starting from more general concepts like Convolutional Neural Networks to more advanced and concrete concepts.

2.1 Convolutional Neural Network (CNN)

A Convolutional Neural Network [9] is a type of Artificial Neural Network (ANN) that is widely used for dealing with data that has a grid-like topology. Although it was developed for two-dimensional data (images), CNNs have been adapted to other types of data such as time-series, which can be seen as a 1D-grid or even videos which can be thought of a 3D-grid having the 2-D dimension for the image plus the extra dimension for the temporal domain.

CNNs work extremely well for grid-like topology data compared to classical ANN due to two important characteristics: **sparse-connectivity** and **parameter sharing**. Instead of connecting all neurons to every single input value as ANNs do, in CNNs the connection is done through smaller local patches of the data also known as **local receptive fields** that **share** the same weights. As a consequence of these two key features, the number of parameters in the network is reduced drastically and once a CNN is trained, it is able to recognize patterns even if they do not appear in the same location.

Typically, a CNN is made of a sequence of three type of layers: **Convolutional layer**, **Pooling Layer** and **Fully-Connected layer**.

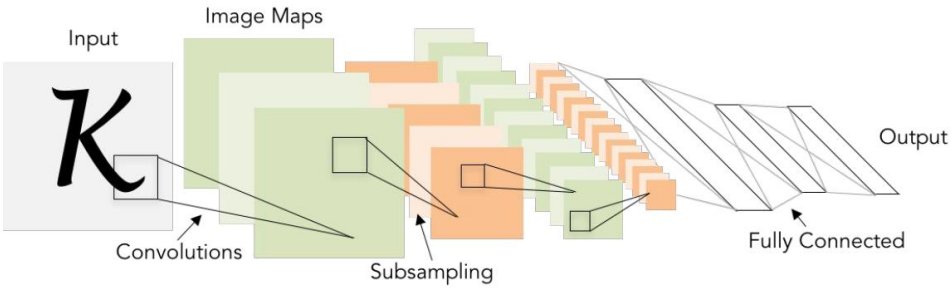


Figure 2.1: Convolutional Neural Network architecture

2.1.1 Convolutional layer

The convolutional layer is the core layer of a CNN. It contains a series of filters known as **kernels** which convolved with the input data produce what is known as a **feature map**. The sparse connectivity of CNNs is achieved by using kernels smaller than the input. The size of these filters is the parameter that will define the spatial extent of the **receptive field**. For example, if our input image I is a two-dimensional $(m \times n)$ image and we use a two-dimensional $(i \times j)$ kernel K, the convolution would be computed as:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \tag{2.1}$$

Usually, CNNs stack multiple convolutional layers in order to extract several feature maps. The first layers of the CNN usually detect low-level features such as edges, color, etc. On the other hand, deeper layers detect high-level features which are more task-dependent such as faces, objects, particular shapes, etc.[10]

2.1.2 Pooling layer

Pooling layers are usually added in-between convolutional layers and they produce a reduced version of the output of the convolutional layer. Just like in convolutional layers, the subsampling is done by sliding a filter over the spatial regions of the input, where the size of those regions is defined by the filter size dimension or **pooling size**. The most used functions to produce the subsampled version are **max-pooling** and **average-pooling**. The first one takes the highest value within the region, whereas average-pooling returns the average within the receptive field.

The main benefits of using pooling layers are the reduction of the com-

putational load, memory usage, and model complexity (thereby reducing the risk of overfitting). Moreover, to some extent, max-pooling layers also make the model local-invariant to translation of the input. The explanation is quite simple, if there are small changes in the receptive field the output of the max-pooling would still be the same.

2.1.3 Fully-connected layer

Fully connected layers are placed at the end of CNNs and are typically used to generate the final output or prediction. A fully connected layer is a layer where the neurons have full connections to all previous layer's activations [11].

2.2 Transfer learning and fine-tuning

Transfer learning is a very common technique to transfer the knowledge that has been acquired from one domain or task and reuse it for a similar domain or task. This technique is widely used on deep learning models where the amount of data is a key factor to get a good performance. Usually, when working on a particular task, the amount of data that you have is quite limited and not enough to train models of high complexity. However, with transfer learning, we can take advantage of a model that was previously trained on a large dataset (e.g ImageNet for images) and reuse these learned feature maps for our task without the need to training the whole architecture from scratch. The two most popular transfer learning strategies for deep learning models are:

- **Use the pre-trained model as a feature extractor**: Deep learning models are layered architectures of different levels that learn different features at each level. For supervised models, at the end of the architecture, there is placed a FC layer to get the corresponding output. That final layer is specific to the task on which the model was trained. So, when using the pre-trained model as a feature extractor, the FC layer is removed, and features corresponding to the previous layer are extracted. This strategy does not require to re-train the pre-trained model as all the layers are frozen (weights fixed).

- **Fine-tune the pre-trained model**: this strategy consists of unfreezing some of the last layers of the pre-trained model and train them with the new samples. The main advantage of this strategy is that it allows us to

fine-tune the higher-order features, making them more suitable for the new purpose.

With transfer learning, we can take advantage of using pre-trained state-of-the-art models like ResNet, VGG and adapt them to our domain or task, not only improving the performance but also reducing the training time. These are the main reasons why our proposed models will use these transfer learning strategies.

2.3 Recurrent Neural Networks (RNNs)

Recurrent neural networks [12], or RNNs, are a class of neural networks that are designed for processing sequential data, being able to **process sequences of variable length**. The main difference from other neural network architectures is their memory capability, as they **use information from prior inputs**, also known as prior context, to influence the current input and output. A visual representation of this process can be seen in Figure 2.2, at each time-step the node receives as input not only the current data but also the information from the previous state, also known as the hidden state (h).

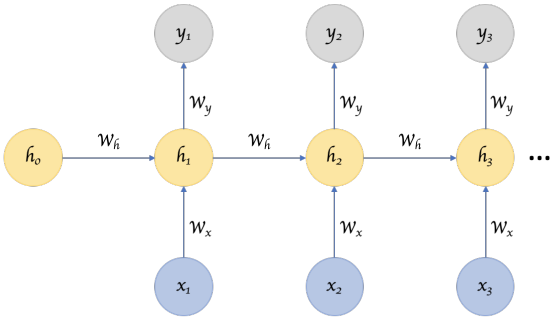


Figure 2.2: Recurrent neural network

Another unique characteristic of RNNs is that they share the parameters' weights (w) across time. Nonetheless, the sharing of weights inherently makes vanilla RNNs tend to have two problems during training: **exploding gradient** and **vanishing gradient** [13]. The magnitude of the gradients increases/decreases exponentially when propagating through longer sequences. These problems make RNNs have quite limited memory, thus making it difficult to learn long-time dependencies.

2.3.1 Long short-term memory (LSTM)

LSTM [14] is a modified version of the vanilla RNN architecture [12] that was proposed by Sepp Hochreiter and Juergen Schmidhuber to solve the vanishing gradient problem. The LSTM architecture addresses the problem of learning long-term dependencies by defining a unit composed of a cell and three control gates (input, output, and forget), used to control the flow of information. The LSTM architecture can be observed in Figure 2.3, while its constituent components are described below.

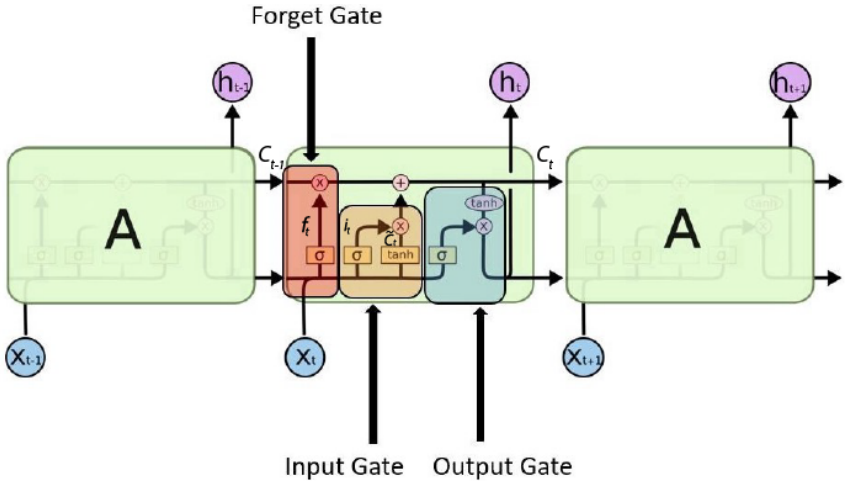


Figure 2.3: LSTM Network

The cell state (C_t) acts as a long-term memory as the updates are carefully controlled by the gates. The forget gate controls which information should be kept from previous outputs, the sigmoid function used will output values ranging from 0 (gate closed - forget) to 1 (gate open - keep) for each number in the cell state C_{t-1}.

The input gate receives the previous hidden state h_{t-1} and the current input x_t and goes through a sigmoid layer to decide which values will be updated. After that, a *tanh* layer creates a vector of new candidate values C_t which is later multiplied by the i_t. The result from this operation is added to the previous cell state C_{t-1}, updating it with the current input information thus producing the current cell state C_t. Finally, the output gate is responsible for deciding what to output considering which parts of the cell state should be passed, so at the end, the final output is going to be a filtered version of the cell state C_t as it is multiplied by the output of the sigmoid gate.

2.4 Transformer

The Transformer model proposed by Vaswani et al. [15] is one of the major advances in the last decade in the Natural Language Processing (NLP) field. It is a novel model that gets rid of the recurrent architecture and it is based solely on multi-head self-attention mechanisms. Even though this model was created originally for the NLP field, its novel attention mechanism has been adapted successfully to other domains such as image classification [16] or video understanding [17] among many others.

2.4.1 Architecture

The architecture follows the classical encoder-decoder structure but instead of using RNN for each block, it uses stacks of multi-head self-attention and point-wise fully connected layers in both the encoder and decoder. Each of them is made of a stack of N_x identical layers, as can be seen in Figure 2.4.

- **Encoder:** each of the encoder layers has two sub-layers or components. The first one is a multi-head attention mechanism (see section 2.4.3) and the second is a feed forward neural network. Around these components, a residual connection is used followed by a layer normalization. The encoder is fed with the embeddings of the original input sequence to which positional information was added. More details about the positional encodings are given in Section 2.4.2.

- **Decoder:** the decoder follows the same structure of the encoder but it has an extra sub-layer and some variations. The extra sub-layer performs multi-head attention that attends over the output of the encoder stack. The modifications are that in the decoder, the outputs embedding are shifted right one position and instead of using a multi-head attention mechanism it adds a masking component. This is done to restrict access to future tokens so that the predictions for a particular position will only be based on previous tokens. Around the sub-layers, residual connections plus layer normalization are implemented, the same as in the encoder.

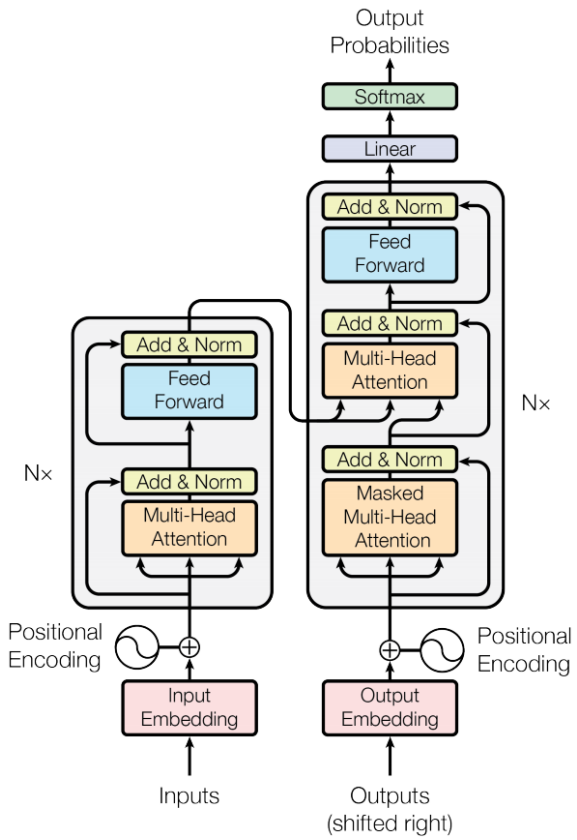


Figure 2.4: Transformer architecture

2.4.2 Positional Encoding

The Transformer model does not use any convolution or recurrence, so it lacks any sequential or positional information. In order to have some information about the relative position of each token in the sequence, a "positional encoding" is added to both the input and output embeddings.

Embeddings are representations of words or tokens in a d-dimensional space where tokens that have similar meaning will be closer in the space. To incorporate information about the relative position of the tokens within the sequence they use positional encodings. These encodings have the same dimension as the embeddings, and they are summed to the input and output embeddings. The resulting embeddings will combine both properties so that words will be closer based not only on the meaning similarity but also on their sequence position.

The positional encodings can either be fixed or learnable, absolute or relative. In the original paper, they present absolute fixed positional encodings that are generated using sine and cosine functions of different frequencies:

$$\begin{aligned} PE_{(pos,2i)} &= \sin\left(pos/10000^{2i/d_{\text{model}}}\right) \\ PE_{(pos,2i+1)} &= \cos\left(pos/10000^{2i/d_{\text{model}}}\right) \end{aligned} \tag{2.2}$$

2.4.3 Attention

The major innovation brought by the the Trasformer model is the multi-head self-attention mechanism. The attention function receives as input three vectors: Q(Query), K(Key) and V(Value). The output of the attention is computed using a weighted sum of the values, where the weights are based on the queries and keys vectors. This particular attention is named "Scaled Dot-Product Attention" as it computes the dot product of the query with all the keys and divides each one by the square root of the dimension. Lastly, a Softmax is applied to obtain the weights on the values, which are the attention scores. The equation to compute the attention weights is:

$$Attention(Q, K, V) = softmax_k(\frac{QK^T}{\sqrt{d_k}})V \tag{2.3}$$

The Multi-head attention consists of splitting the attention into h heads by linearly projecting the embeddings into h different sets of Queries, Keys and Values. Once the splits and projections are done, the Scaled Dot-Product attention of the split h heads is computed in parallel. The results are concatenated and projected to obtain the final values. The main advantage of splitting the heads is that each head is able to attend to different positions of different representation sub-spaces, thus improving its ability to capture different semantics than if only using one head.

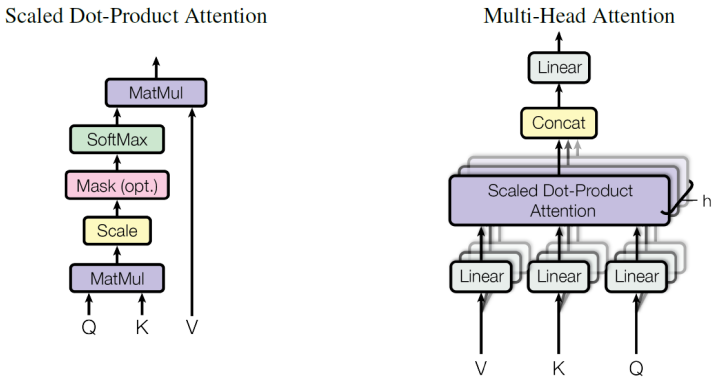


Figure 2.5: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel

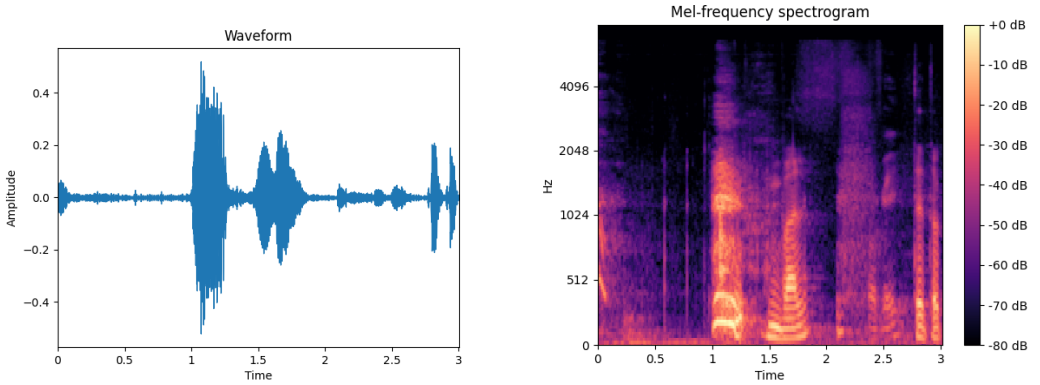
2.5 Mel spectrogram

Audio machine learning models were mainly based on applying digital signal processing techniques to extract audio features. The appropriate choice of what features to extract to solve a particular problem required domain expertise in the audio field. Now, thanks to the continuous improvement that deep learning models have had in the field of audio, it is no longer necessary to apply these digital signal processing techniques to extract hand-crafted audio features. In fact, one of the main advantages of using Deep Learning models for audio is that we can previously convert the raw audio into an image-like structured data and then apply modern CNN architectures to extract audio features.

The conversion from raw audio to image is done through the generation of spectrograms. Spectrograms are generated by splitting the sound signal into smaller time splits (windows) and applying Fourier Transform to each window to later combine all results into a single image. Basically, a spectrogram is a visual representation represented as a heat-map that captures the audio information by decomposing the signal into the frequencies of the audio signal.

The most used representation is the Mel Spectrogram as it adds some variations that make the resulting spectrogram more accurate to how humans hear sounds. Humans' hearing does not work on a linear scale, we have an increased sensitivity to detect changes at lower frequencies whereas we struggle to detect differences of the same magnitude at higher frequencies. The Mel scale tackles this problem by defining a unit where equal distances in pitch sounded equally distant to the listener.

The Mel Spectrogram replaces the Frequency scale with the Mel Scale and optionally uses the Decibel scale instead of amplitude to produce the colors. A sample conversion from a raw audio signal to its corresponding Mel Spectrogram can be seen in Figure 2.6b



(a) Raw audio signal (b) Corresponding Mel Spectrogram

Figure 2.6: Raw audio signal and its corresponding Mel Spectrogram

The Mel Spectrogram is the audio representation that our audio models will use in order to extract the corresponding audio features.

Chapter 3

Related Work

This chapter reviews related work on personality recognition, dividing the literature review according to the modality used.

3.1 Visual modality

As in any task where characteristics from an image or a sequence of images have to be extracted, Convolutional Neural Networks (CNN) is the preferred choice due to its state-of-the-art performance. When using still images, without using the temporal information, most of the works focus on facial features, as it has been proved that the face provides most of the descriptive information for the recognition of personality traits [18, 19].

Ventura et al.[20] studied why CNN architectures achieved great performance in the complex problem of personality recognition. Activation maps were used to understand what are the parts that CNN focuses on, demonstrating that activation is always centered around the face, and focus on parts like eyes, eye-brows, and mouth among others to predict the personality traits.

Yan et al.[21] studied the relationship between facial appearance and trustworthiness impression. Up to sixteen features are extracted from five face regions: eyebrow, eye, nose, mouth, and face shape. A combination of methods is used to extract those features, such as Histogram of Gradients (HoG) for describing the eyebrow shape or Euclidean Distance (ED) to extract the width of eyes. The relationship between the face features and personality impression is found using Support Vector Machine (SVM).

Gürpinar et al.[22] combined deep facial and ambient features to infer personality traits. Before extracting facial features, IntraFace [23] is used to

extract face landmarks. These are later used to define a bounding box based on which the image will be cropped to contain only facial information. The corresponding region is fed to a pre-trained CNN model named VGG-Face [24] that is already specialized for extracting facial features. Ambient features are extracted using a VGG-19 [25] network pre-trained for an object recognition task and then fused with the extracted deep facial features. The fused features are fed into a kernel Extreme Learning Machine (ELM) regressor [26].

It is worth mentioning that the vast majority of studies that use visual data take little advantage of the temporal domain, if present. Most works do not exploit spatiotemporal feature extraction models like 3D CNNs such as R(2+1)D [27], I3D [28], or other models that are the state-of-the-art for tasks that use the same input data (e.g action recognition).

3.2 Audio modality

Prosodic features have been widely used to analyze human speech. Polzehl et al.[29] show that extracting prosodic features like intensity, pitch, spectrals, Mel-Frequency Cepstral Coefficients (MFCC) among others, and feeding them into a SVM regressor can accurately predict personality trait scores. Moreover, they show that some features are more important to certain traits than others. For predicting Openness and Conscientiousness traits, the features extracted from MFCC were the ones that get higher scores. On the other hand, for Extraversion and Agreeableness traits, pitch features were the most important whereas for predicting Neuroticism trait scores, the best performing features were those obtained from loudness and intensity analysis.

Park et al.[30] proposed novel audio features that not only consider speech features but also sound and lexical characteristics. Features like averaged reaction time, averaged pitch frequency and averaged sound power were used to differentiate between extroverted and introverted people. In their findings, they show that there are significant differences between the average reaction and total reaction time between introverted and extroverted people while answering the same set of questions. Their findings corroborate previous studies that observed that introverted people tend to produce longer silences when facing complex verbal tasks [31, 32] whereas extrovert people have faster processing brains [33], even though this also makes them more susceptible to make errors [34].

Most personality recognition models based solely on audio used to perform the recognition in two separate steps: first, extract the features and

then train a regressor model. However, with the increase in performance of CNN architectures, end-to-end approaches have been gaining popularity as the audio features can be extracted by CNN itself by using as input the image-like Mel Spectrogram representation (see Section 2.5). Carbonneau et al.[35] demonstrate the advantages of feature learning models, achieving state-of-the-art results on SSPNet Speaker Personality Corpus [36] using a single model purely based on CNN and using Mel-Spectogram as input. The appearance of new CNN architectures trained on Mel Spectrogram representations like Vggish [37] has increased the use of these CNN-based models to extract audio features. Aslan et al.[38] directly use the pre-trained Vggish model to extract high-level audio features achieving state-of-the-art results[1] on the ChaLearn First Impressions V2 challenge dataset [39].

3.3 Multimodality

Most of the multimodal models for personality recognition combine audio and visual features, although there are a few that also include text [40, 38]

Zhang et al.[8] proposed the Deep Bimodal Regression (DBR) framework for apparent personality analysis. The framework treats the videos and splits them into two modalities: the visual modality (frames) and the audio modality (speech). For the audio modality, the log filter bank features are extracted and fed into a linear regressor model to produce the OCEAN trait values. For the visual modality, a modified version of CNN named Descriptor Aggregator Network (DAN) [41, 42] is used. The main modification is the max-pooling and average pooling that is done on the last convolution layer to produce a single 1024-dimensional vector (512 from avg-pooling, 512 from max-pooling). On top of that, a fully connected layer is added to produce the OCEAN scores. Audio and visual modalities are lately fused by averaging their predictions. This framework achieved the highest score on the ChaLearn Looking at People challenge (2016) [43].

Subramaniam et al.[7] propose two bi-modal end-to-end architectures that use temporally ordered audio and stochastic visual features from frames. Each video is split into N non-overlapping partitions. From each partition, the audio is extracted as the auditive information and a frame is randomly selected as the visual information. For the audio modality, 11 hand-crafted features are extracted like Zero Crossing Rate, MFCCs, energy among others. For the visual modality, a CNN is used to extract the visual features. Unlike the

[1] The overall model also includes visual information

previous method, the modalities are not lately fused, the audio and visual features are concatenated and then passed through an LSTM, allowing the model to learn temporal patterns. The outputs of the LSTM are fed into a fully connected layer that produces the corresponding OCEAN values.

3.3.1 UDIVA model

The closest work to the current thesis is the one of Palmero et al.[1], as the experiments we conducted are on the dataset they introduced. Furthermore, we also perform an extended analysis of their model.

The dataset used to train their model, named UDIVA, is a non-acted dataset where interlocutors grouped in pairs perform competitive and collaborative tasks (see Section 4.1 for more details)

The main difference of this model, concerning those discussed previously, is the use of contextual information to infer the personality of the target person. It uses visual and metadata information from the other interlocutor (extended context data) to predict the personality of the target interlocutor. The modalities used are visual (face, local context and extended context chunks), audio and metadata. The visual features are extracted by a 3D-like CNN network named R(2+1)D [27] pre-trained on IG-65M dataset [44]. Audio features are extracted using the Vggish network [37] pretrained on a preliminary version of Youtube-8M [45]. The modalities, like in [7] are early fused and the temporal features are learned by using a Transformer network (Tx) [15] composed of two Tx units one of them using local context keys and values and the other using extended context keys and values. The query for both units is the same and comes from the face and metadata features (face query features) after passing through a fully connected layer. The outputs of the two units in the last Tx layer are concatenated and fed to a FC layer to regress the per-chunk OCEAN scores. The overall architecture is presented in Figure 3.1.

We want to emphasize some details about the training strategy of this model. The first one is that it is **not an end-to-end model** since the visual and audio features models are frozen. The second one is that the model is **trained on each task separately**. The last and most important is that **the model does not model long-term temporal dependencies**, it only models local dependencies at chunk level through the R(2+1)D architecture. These are some of the differential key points between our proposed models and theirs.

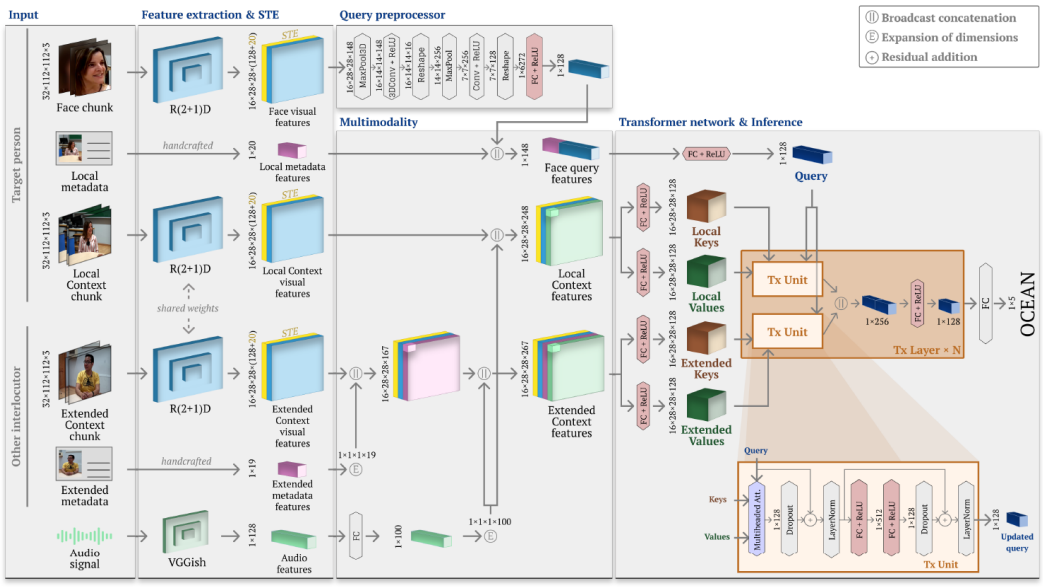


Figure 3.1: UDIVA proposed model

3.4 Summary of the literature review

In this Chapter, we have reviewed some of the literature on personality recognition. This is an increasingly popular research area, but due to the limitation of existing datasets, the number of research papers on this field is scarce.

Regarding the extraction of visual features, we have observed that most of the methods directly focus on the face region as it has been proved to be where more relevant information can be found. The preferred architectures for extracting facial features from images, unsurprisingly, are CNNs. Most works take advantage of CNN architectures that are already specialized for this task, such as Vgg-Face.

In the audio modality, we have seen that prosodic features perform extremely well for this task and the majority of the models are trained using a two-stage approach: (1) feature extraction, (2) regressor model. However, thanks to the evolution in the deep learning field and especially in Computer Vision, feature learning models have been widely adopted to perform audio feature extraction by using Mel Spectrogram representation. This allows the training of end-to-end architectures using the audio modality for personality regression tasks.

When fusing modalities we observed two different approaches, the first

one and the most common is to extract the features from each modality and then fuse them by averaging their predictions (late fusion). The second and most advanced approach is by fusing the modalities at an early stage and use a temporal feature extraction model like LSTM or Transformer to learn the temporal patterns and provide a more ”intelligent” way of fusing modalities.

Finally, we presented the UDIVA's multimodal model. It is the first model explicitly implemented for the dataset that we are using. We will take advantage of this to fairly compare and analyze the performance of both models (theirs and ours) on the same dataset, the comparison is done in Chapter 6.

Chapter 4

Data

Data is a key element in any machine learning project and even more when working with deep learning models that are "data-hungry". However, it is preferable to have a decent amount of data, but representative for the task, than a large amount of data, what is known as quality over quantity.

In this section we start by introducing the multimodal dyadic dataset, we continue explaining how the data has been pre-processed to ensure that is suitable for the models and we finish by describing the dataset splitting strategy and showing the corresponding splits of the dataset into the train, validation and test sets.

4.1 UDIVA dataset

The UDIVA dataset (Understanding Dyadic Interactions from Video and Audio Signals) [1] is a non-acted multimodal dataset that consists of face-to-face dyadic interactions. The dataset is made up of a total of 188 dyadic sessions where 147 participants[1] grouped in pairs had to carry out a series of free and structured tasks within a controlled environment. The dataset also includes **self-reported** and **peer-reported** personality assessments.

The self-reported personality was assessed using different standardized questionnaires depending on the age of the participant. In the case of participants below 9 years old, the Children Behavior Questionnaire (CBQ) [46] was filled by their parents. Participants in the range between 9 and 15 years old

[1]Participants gave consent to be recorded and to share their collected data for research purposes, in compliance with GDPR https://ec.europa.eu/info/law/law-topic/data-protection_en

completed the Early Adolescent Temperament Questionnaire (EATQ-R) [47] whereas participants aged 16 or above completed the Big-Five Inventory [48] as well as the Honesty-Humility axis of the HEXACO personality inventory [49].

At the end of each session, all participants over 8 years of age had to complete the same personality questionnaires but this time evaluating the participant with whom they had been interacting during that session.

4.1.1 Dataset statistics

In this section, we briefly introduce some statistics regarding the distribution of participants in terms of gender, age, nationality and spoken language[2]. Regarding the sex of the participants, equity between genders is very high (55.1% men, 44.9% women). The age of the participants ranges from 4 to 84 years, the mean age being 31.29 years old. Participants come from 22 different countries, with Spain being the country with the highest representation (68%). The most used language during interactions is Spanish (71.8%) followed by Catalan (19.7%). To create the pairs, participants were matched according to their availability and language, and ensuring a close-to-uniform distribution using up to 60 variables such as age, gender, relationship between interlocutors, among others [1].

4.1.2 Dyadic session structure

A dyadic session consists of 5 different tasks that were specifically selected by psychologists to capture the variety of individual and dyadic behaviors and the cognitive workload that these tasks cause to the participants.

- **Talk**: participants were instructed to speak on any topic for approximately 5 minutes. This task allows analysis of common conversation constructs, such as turn-taking, synchrony, empathy and quality of interaction, among others [1].
- **Animals game:** each participant had to ask the other participant 10 yes/no questions to try to find out which animal was on his forehead. This game reveals cognitive processes (e.g. thinking) [1].
- **Lego building:** participants had to build, following the instructions booklet, a certain construction using Lego pieces. This task encourages

[2]For a more detailed analysis please refer to the Section 3.2 on the UDIVA paper[1]

collaboration, cooperation, joint attention, and leader-follower behaviors, among others [1].

- **Ghost blitz card game:** participants had to select, from a set of 5 figures, the one whose color and shape was not shown on a selected card. They played 1 card per turn, competing among themselves to be the first to select the correct figure. This task encourages competitive behavior and allows analysis of cognitive processing speed, among others [1].
- **Gaze events:** participants followed instructions given by a proctor, to look at each other's faces, at a static/moving object, or anywhere else while moving their head and eyes. This task serves as ground truth for gaze gestures and facial modeling with varied head poses [1].

Figure 4.1: Sample image of each task. From left to right: Talk, Lego, Animals, Ghost, Gaze

All sessions were recorded in a controlled environment where participants had to sit around a table forming a 90-degree angle between them, close enough to be able to carry out the previously mentioned tasks. The recording environment (see Figure 4.2) consisted on six mounted with a tripod named **GB**: General Rear camera, **GF**: General Frontal camera, $\mathbf{HA_{1,2}}$: individual High Angle cameras and $\mathbf{FC_{1,2}}$: individual Frontal cameras, two cameras (one per participant) placed around the neck, two lapel microphones(one per participant) and an omnidirectional microphone placed in the middle of the table.

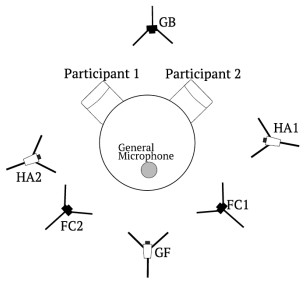


Figure 4.2: Recording environment.

4.2 Dataset preprocessing

When dealing with multimodal data, preprocessing the data is an extremely important step for making the data suitable for training deep learning models. This preprocessing part was not part of this thesis and was done by the authors of the UDIVA dataset paper [1].

The input data that is used for this master thesis comes from the individual camera video (FC1 and FC2) which were recorded at 25 fps and with a definition of 1280 x 720 pixels and the synchronized audio from the corresponding lapel microphone. As can be seen in Figure 4.3, each video is split into chunks of around 3 seconds and the corresponding frames and audio are extracted. Images are re-scaled to a definition of 224 x 224 pixels.

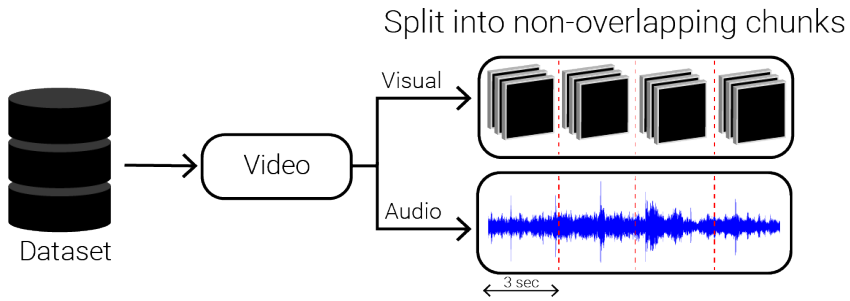


Figure 4.3: Preprocessing overview. Videos are split into non-overlapping chunks of 3 seconds. Visual information (frames) and audio are extracted separately from each chunk.

UDIVA model [1] was inspired by the Video Action Transformer [50] that uses 64 frames per chunk (3 seconds). As the UDIVA used the R(2+1)D backbone to extract the spatiotemporal features, which only uses 32 images, a stride of 2 was applied to generate the chunks. Therefore, making sure that each chunk encodes the same time-window as in Video Action Transformer model [50]. This preprocessing is equivalent to down-sample the original video from 25 fps to 12.5 fps.

4.3 Dataset split

To obtain good performance with any model, the quality of the data is extremely important. A proper balance between the sets is essential if we want to get a model that is able to generalize. In this section, we describe the

strategy followed to divide the dataset into the corresponding train, validation and test sets. We must emphasize that the dataset split has not been part of this thesis and was done by the authors of the UDIVA paper [1]. The split strategy tried to minimize the costs using a greedy optimization strategy to have a similar distribution in terms of participant and session characteristics between the sets. As stated in Section 5 of the supplementary material of the UDIVA paper [51], the method tried to minimize the costs to:

1. Ensure that distributions among splits were not different employing a Kolmogorov-Smirnov significance test
2. Ensure that Pearson's correlation of gender, age and personality values among splits did not differ by a large margin
3. Attempt to have a uniform distribution in validation and test with respect to age and gender to correct selection bias
4. Attempt to have a close-to-uniform distribution of group combinations
5. Try to maximize the number of sessions without losing participants while considering also the train/validation/test ratio

It is important to mention that the optimization algorithm ensures that there are **no participants repeated between sets**.

From the splits mentioned, we have only used the subset of data from **participants over 15 years old** and we **discard all Gaze task instances** (same as in [1]). The decision to remove instances from the Gaze task was made because there were very few personality indicators in it due to the task design[1]. On the other hand, the decision to use only the subset of participants older than 15 years is for having a personality trait score with the same representation: the Big-Five traits. Finally, the last data cleaning step is the removal of chunks in which the participant does not appear.

The distribution of the self-reported personality traits over the sets, after removing Gaze entries and all the sessions from participants under 15 years old, is presented in Figure 4.4.

In terms of the number of instances, the training set represents the 80% of the total. Typically, the split percentage for validation and test is the same but in this case, the validation set is almost double in terms of the number of instances compared to the test set. Nonetheless, the latter has a better personality trait balance, so it will be more representative for evaluating the performance of our models. A complete overview of the statistics of the dataset splits can be seen in Table 4.1

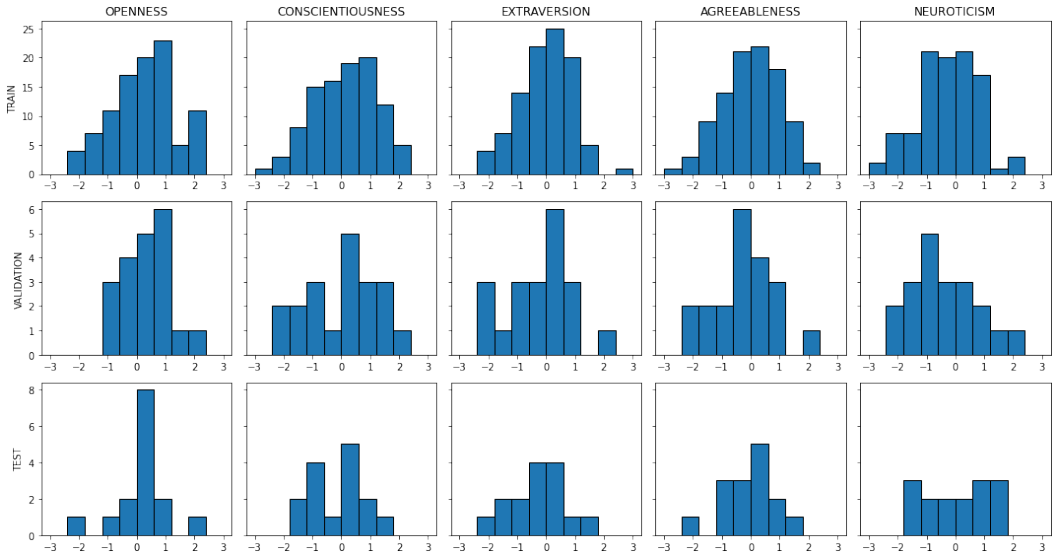


Figure 4.4: Distribution of the self-reported personality traits (OCEAN) over the train, validation and test splits. X axis represents the values of each personality trait. Y axis represents the number of participants.

Split	**#Participants**	**#Sessions**	**#Instances**
Train	99	116	94960
Validation	20	18	15350
Test	15	11	7870

Table 4.1: Dataset split statistics

Chapter 5

Models

5.1 Multimodal model

In this section, we present the structure of our multimodal model that receives as input a chunk of a video with its corresponding sound and outputs the personality prediction. The description of the model is first presented divided into each of the sub-models that make up the final model: the visual model, the audio model, and the sequence model. Finally, we present the multimodal model and explain how the modalities are fused.

5.1.1 Visual feature extraction model

To extract the visual features we have opted for using one of the state of the art methods for action recognition and video understanding: the R(2+1)D model [27].

The R(2+1)D model compared to other well-known 3D architectures like I3D [28] or C3D [52] proposes a novel type of convolution the (2+1)D that approximates the 3D convolution by factorizing the computation into a spatial 2D convolution (frames) followed by a temporal 1D convolution (time). This approximation has shown significant performance improvements over the previously mentioned models in a wide range of datasets such as Kinetics [53] or Sports-1M [54].

We have used the R(2+1)D with 34 layers pre-trained on the IG-65M [44] (65 million Instagram videos) dataset and fine-tuned on the Kinetics dataset [53]. The model architecture is represented in Figure 5.1, it is composed of 5 (2+1)D convolution blocks, a space-time (3D) average pooling layer, and on top of that a fully connected layer. We adapt this architecture by removing the

classification layer (fc) and **freezing** the first 4 (2+1)D convolution blocks and **fine-tune** with our data only the last (2+1)D convolution block. Nonetheless, the last block contains more than half of the total parameters of the model (39.1 M out of 63.5M).

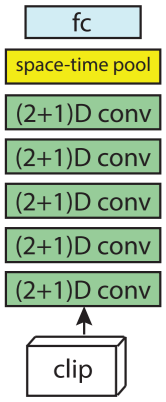


Figure 5.1: R(2+1)D Architecture

The input of the model is the 32 frames of the corresponding video chunk which corresponds to approximately around 3 seconds of the original video. The frames have a resolution of 224×224 pixels, we normalize the pixel values in the range $[0, 1]$. The produced features after feeding the input into the R(2+1)D backbone is a single 512-dimensional vector that from now on we will refer to as the **visual features** vector.

5.1.2 Audio feature extraction model

Nowadays, most of the state-of-the-art audio classification models use CNN architectures to extract the audio features. To do so, the audio signal must be first converted into an image-like representation, in our case the Mel Spectrogram. As mentioned in Section 2.5, the Mel Spectrogram is a heat-map where the color is represented by the decibels of the sound, the Y-axis represents the sound frequency and the X-axis represents the time or duration of the sound(in seconds).

In order to extract audio features we propose two different models: the **Vggish** [37] model and the **ResNet-18** [55]. Both models receive as input the corresponding Mel-Spectrogram of the audio chunk (3 seconds) and return a 128-dimensional vector in the case of the Vggish model and a 512-dimensional vector in the case of ResNet-18. The audio feature extraction is presented in Figure 5.2, the input is the raw 3-second audio signal which is converted into

a suitable representation, the Mel Spectrogram, which is fed into the CNN model that will produce the **audio features'** vector.

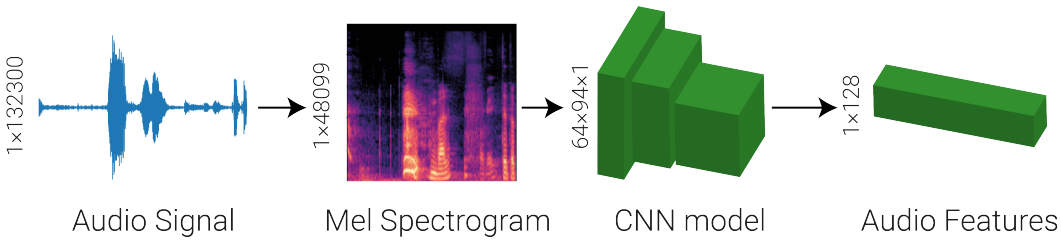


Figure 5.2: Audio feature extraction process

To obtain the Mel spectrogram, we have used the same steps as in the Vggish method. The audio signal is re-sampled from 44100 Hz to 16000 Hz, then the spectrogram is computed using 64 Mel bins that cover only the range of 125-7500 Hz, which is an approximation of the human speech sound range.

5.1.2.1 Vggish

The Vggish model is a modified version of the well-known VGG [25] model, in particular, Confgiuration A with 11 weight layers. The minor modifications are: changing the input size to 96×64, removing the last group of convolution blocks (going from 5 to 4), and finally the replacement of the last 1000-wide fully connected layer to a 128-wide fully connected layer that acts as a compact representation layer(embedding).

This model has been chosen because it is specifically designed for extracting audio features and is pre-trained on a large Youtube dataset, a preview version of Youtube-8M [45]. We use this model as a **feature extractor**, converting our audio input into a semantically meaningful 128-dimensional audio feature vector.

5.1.2.2 ResNet-18

The second audio model is the ResNet-18 model which belongs to the family of ResNet [55] architectures. ResNet-like architectures have a strong advantage over Vgg-like architectures: the residual block. There most classical representation of how a residual block looks can be seen in Figure 5.3, even though there are several variations. Similar to LSTM cell, where the gates control the data flow from previous steps, Residual Networks reduce the vanishing gradient problem by adding residual blocks. These blocks introduce skip

connections which enable the creation of deeper models, not only with fewer parameters but also with better performance than other CNN architectures like VGG.

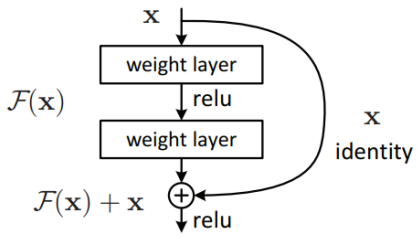


Figure 5.3: Residual block

The ResNet-18 is a network with 18 layers: the first 7x7 convolutional layer, 4 modules of 4 convolutional layers (each module has two residual blocks) and the final fully connected layer. We use the ResNet-18 model pre-trained on ImageNet but removing the last fully connected layer. It is important to highlight that ResNet-18 necessarily requires **fine-tuning** due to the difference between the domain in which it has been pre-trained and our domain. However, as this model is much smaller we will use the pre-trained weights as a warm-up for training the whole network with our audio data.

We have used two different transfer learning strategies for each model. The Vggish model allows us to directly extract the features of our audio without the need for fine-tuning, using it as a **feature extractor**. On the other hand, for the ResNet-18 we apply a **fine-tuning** strategy due to the difference between the domain in which it has been pre-trained and our domain, but this approach allows us to create an end-to-end model. A detailed analysis between the performance of these two strategies is done in Section 6.4

5.1.3 Sequence model (Encoder)

The sequence model or encoder is implemented to produce a long-term context vector of consecutive video chunks. The input of the sequence is the concatenation of the corresponding audio feature vector and the visual feature vector of a single video chunk after passing each of them through a linear layer of 128 units with ReLU as a non-linear activation function. The sequence input can be represented as $\mathbf{s}_{in} \in \mathbb{R}^{256}$ where $\mathbf{s}_{in} = (\mathbf{v} \in \mathbb{R}^{128} \oplus \mathbf{a} \in \mathbb{R}^{128})$ being $\mathbf{v}$ the visual feature vector and $\mathbf{a}$ the audio feature vector.

When talking about the sequence model, we will often use the terminology time step but we must point out that this is not equivalent to the seconds of

the video but to the number of sequence inputs. Therefore at time step $t = 0$ we will be processing the first chunk (3 seconds) of a list of consecutive video chunks.

The goal of the encoder is to process a sequence of inputs, video chunks, and encode all the information into a fixed-length vector (context) which in our case it is a 128-dimensional vector. We implement two different sequence architectures that have been widely used to solve this kind of problem: LSTM and Transformer.

As mentioned in Section 2.3.1, LSTM is a type of recurrent neural network that is specially designed to solve the vanishing gradient problem. This is extremely important as we want to study how the model performs while modifying the sequence length, and this kind of problem may occur if using vanilla RNN. This first encoder model is an LSTM with depth one (no stacking) and 256 hidden units. The context vector is retrieved using the last hidden state, what is known as a many to one LSTM configuration (see Figure 5.4).

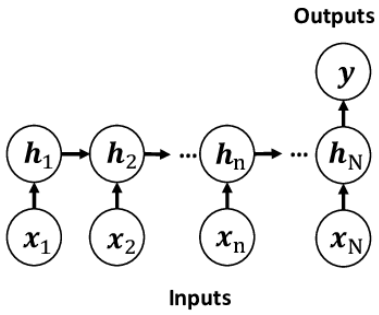


Figure 5.4: Many to one LSTM structure

On the other hand, the second model uses as sequence model the classical Transformer Encoder block. If we recall the theory about Transformer (see Section 2.4), this model does not handle the data in a sequential way like LSTM, so we must explicitly add some information about the order. The positional encoding is a vector of the same dimensionality of the input that is added to include positional information. There exist multiple ways to encode the position information [56] but we use the original Transformer's paper [15] positional encoding formulation, which uses using sine and cosine functions of different frequencies (see Equation 2.2) to encode the order of the sequence.

The Transformer encoder is composed of $Nx = 2$ encoder layers (see Figure 5.5) with 4 attention heads each and using 256 units for every feed-forward network within the architecture. The context vector of the sequence is obtained

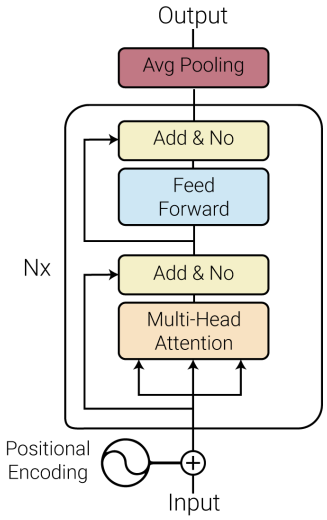


Figure 5.5: Transformer Encoder block with an additional Avg. Pooling layer to produce the context vector

using an Average Pooling layer over the transformer outputs, which results in a single 256-dimensional vector.

5.1.4 Multimodal model overview

The multimodal model is composed of 3 basic modules: a module that is in charge of extracting the visual characteristics of the video chunk, another that is in charge of extracting the audio characteristics of the video chunk and finally a module that is in charge of sequentially treating the extracted characteristics by the modules mentioned above to create a summary or a context vector of the multiple video chunks information using a greater time range. Where the time range is defined by the number of time steps.

The modular design of our multimodal model allows us to carry out different experiments ignoring some parts and testing a multitude of configurations. To go into a little more detail, this section shows an example of what the Multimodal model could be, taking into account that both the sequential and the audio modules have two possible variants. In the case of audio, we have the possibility of using either the features extracted from the Vggish model or the ones extracted from the ResNet-18 model. Similarly, in the case of the sequential module, we have two variations, the first is based on the use of a recurrent neural network such as the LSTM, while the other model is mainly based on the Self-Attention mechanism of the Transformer network.

In Figure 5.6, we present how the multimodal looks, using the Vggish as the audio feature extraction model.

The input of the visual feature extraction model is the 32 consecutive frames(video chunk $\approx$ 3 seconds) of the target participant, whose personality we want to predict. Let define the local context chunk frames as $\mathbf{X}_{FL} \in [0, 255]^{32\times224\times224\times3}$. Before feeding $\mathbf{X}_{FL}$ into the R(2+1)D backbone, the pixel values are normalized in the range of [0, 1]. The R(2+1)D model provides us a visual feature vector $\mathbf{v} \in \mathbb{R}^{512}$. To have the same importance between modalities (in terms of feature dimension) the visual feature vector $\mathbf{v}$ goes through a fully-connected (FC) layer of size 128 and a ReLU activation function layer to introduce non-linearity to the network. Let's denote the reduced visual feature vector as $\mathbf{v}' \in \mathbb{R}^{128}$

On the other hand, the Vggish model provides an audio feature vector $\mathbf{a} \in \mathbb{R}^{128}$ encoding the audio information of the video chunk. Following the recommendations on the Vggish implementation page[1], we send the embedding through a fully-connected (FC) layer of size 128 and a ReLU activation function layer (introducing non-linearity). The resulting vector, $\mathbf{a}' \in \mathbb{R}^{128}$ is fused with the reduced visual feature vector $\mathbf{v}'$ through a concatenation operation forming the audiovisual feature vector $\mathbf{f} \in \mathbb{R}^{256}$ where $\mathbf{f} = (\mathbf{v}' \in \mathbb{R}^{128} \oplus \mathbf{a}' \in \mathbb{R}^{128})$

The sequential encoder model processes L audiovisual feature vectors, being L the sequence length or, in other words, the number of time-steps. We can define the input of the sequential encoder as $s = [f_0, f_1, \cdots, f_{L-1}, f_L]$. The sequence length L is a fixed parameter, so all the sequences will have the same length. The output of the encoder block is what is known as the context vector $\mathbf{c} \in \mathbb{R}^{256}$. The context vector goes through a FC layer of size 128, thus compacting it, and a ReLU activation layer to finally go through the last FC layer of size 5 that produces the OCEAN trait's prediction ($\hat{y} \in \mathbb{R}^5$) of the target participant.

[1]Vggish implementation details and usage recommendations can be found in: https://github.com/tensorflow/models/tree/master/research/audioset/vggish

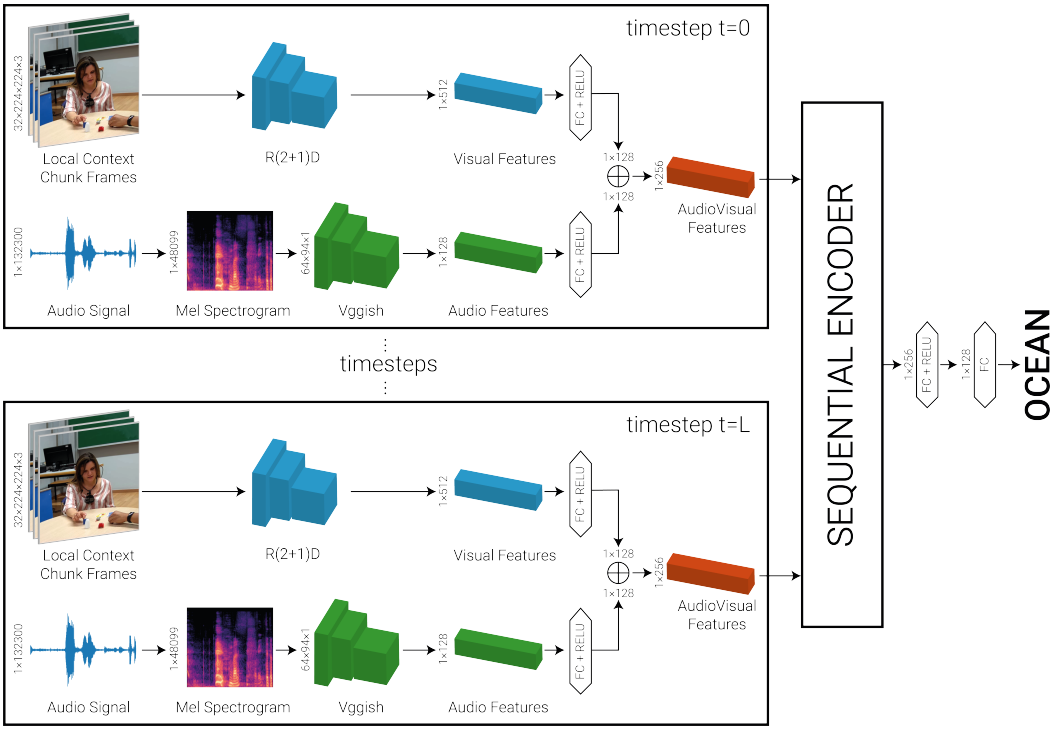


Figure 5.6: Multimodal model overview, using Vggish as the audio feature extraction model

Chapter 6

Experiments and Results

In this chapter, we begin by presenting the experimental setup that describes the training and evaluation strategies that we follow to assess the performance of our models. Then, we describe the various experiments we have carried out and analyze the results obtained by the different models defined in Chapter 5.

6.1 Experimental setup

This section describes the experimental setup used to evaluate the performance of our models. As mentioned in Section 4.3, the data used to train and evaluate our models comes from the sessions of participants over 15 years all and all tasks except Gaze. In order to recognize the personality, we use the frontal camera view $FC_{1,2}$ chunks (frames and audio). The ground truth personality labels are obtained using the self-reported BFI-2 questionnaire [48] and converting the answers into the corresponding OCEAN trait score values. Models are trained in end-to-end fashion and for all tasks, except Gaze.

6.1.1 Dataset split

Due to the size of the models, limited computational resources and time constraints, we opted to split the training set to carry out a greater number of experiments and to be able to make a better analysis of the different models. We used a stratified splitting based on session that consisted of selecting the first N chunks of each session where N is computed as:

$$N = \frac{N_{chunk} \cdot P}{N_{session}} \tag{6.1}$$

Where:

N: is the number of chunks to select per session

N_{chunk}: is the total number of chunks in the set

P: is the subsampling percentage value [0,1]

$N_{session}$: is the total number of sessions in the set

The N_{chunk} depends on the number of timesteps selected (L). As mentioned in Section 4.3, after preprocessing a video, there may be a temporary discontinuity with regards to chunks because the participant does not appear in one or some of them. To maintain temporal continuity, when creating the dataset for each split, we make sure that all t chunks are temporally continuous.

This strategy ensures that all sessions will be included in the training set and all will have the same number of continuous chunks. Selecting the first part of videos is the default strategy in domains like action recognition. In the personality field, the results obtained by Teijeiro-Mosquera et al.[57] suggested that personality was better predicted when using features of the beginning of the video.

The split percentage value for the training set is set to $P = 0.3$[1] whereas for the validation set the split percentage value is set to $P = 0.5$.

6.1.2 Training strategy

Training deep learning models is a complex task due to the huge amount of hyperparameters that must be set. A grid search of all the hyperparameters is not feasible as it would take an excessive amount of time and it is not the aim of this thesis. Instead, we define the hyperparameters to a certain value and use the same for all the models.

The training strategy followed consists of monitoring the loss on the validation set and stop the training when the validation loss stops decreasing after a certain amount of epochs or when the maximum number of epochs is reached[2], this strategy is known as **early stopping**. The loss is evaluated as the **Mean Squared Error** between the predicted personality trait score and the ground truth labels (see Equation 6.2).

[1]This percentage value is used when the visual modality is included as the R(2+1)D model is very time-consuming

[2]The maximum number of epochs varies depending on the model

Due to the long training time, after each epoch, a checkpoint of the model with its corresponding weights and the optimizer state is saved. This strategy has been extremely important because on many occasions the models have been trained in two stages. Being able to resume training from where you left off has been a key factor in optimizing time and reuse some trained models to test other configurations.

6.1.2.1 Optimizer and learning rate

The optimizer is the method used to minimize the error function by updating the model's parameters. We use an optimizer that belongs to the family of mini-batch gradient descent algorithms. Mini-batch gradient descent algorithms perform an update of the model's parameters every n training examples (mini-batch) instead of using all training samples. Among the many methods that belong to this family we select **Adam** [58] as our optimizer using the default decay rate coefficients $\beta_1 = 0.9$, $\beta_2 = 0.9$ and $\epsilon =$ 1e-8. Adam has been widely adopted as the "default" optimizer due to its favorable performance compared to other optimizers and its robustness to the selection of the hyperparameters [10].

The learning rate is another key hyperparameter to select and it is one of the most difficult hyperparameters to set due to its strong relation to the model performance. A whole analysis to select the best learning rate could be done but this is out of the scope of this project. The learning rate is set to the default value of Adam's optimizer, 1e-3, for all models. A learning rate scheduler is applied to adjust the learning rate by a factor of 0.1 based on the validation metric after $n = 2$ epochs. If the model does not reduce the validation loss after 2 epochs the learning rate will be reduced multiplying it by a factor of 0.1.

6.1.2.2 Batch size selection

Batch size is one of the most important hyper-parameters when training a deep learning model. It defines the number of samples used to train a single forward and backward pass. There have been multiple studies analyzing how this parameter affects the generalization of the model. Small batches tend to produce noisy gradients, but this behavior has been shown to serve as a regularizing effect [59] while larger batches provide a more accurate estimate of the gradient. However, this holds up to a certain point as studies like [60] have demonstrated that using larger batches may produce a degradation in the generalization ability of the model. As the batch size increases, the amount of

memory required increases with it. This is our limiting factor due to limited computing resources. The batch size selected for all the models[3] is set to 16 which corresponds to the highest value that fits our GPU's memory and it is also a very common value when training large models [10].

6.1.3 Environment and parallelization strategy

The environment used to train our models consists of 4 GeForce GTX 1080 Ti 11GB. Given the model and data size, the training has been distributed among all the GPU's devices. The strategy selected is known as **distributed data parallelism**. It consists of distributing the batches among the devices. This requires that each GPU has to replicate the whole model. Each GPU computes the forward pass using its portion of the input and during the backward pass the gradients from each node are averaged and the parameters are updated and synchronized among the devices. In our case, this means that each GPU will be processing only 4 instances.

6.1.4 Evaluation strategy

Models are evaluated using the Mean Squared Error between the predicted personality trait score and the ground truth labels (see Equation 6.2). Due to computational resources and time-constraint limitations, the evaluation of the models is done using 50% of the test set. The results are also compared with a baseline value computed as the mean of the per-trait ground truth labels of the training set.

$$\frac{1}{5N}\sum_{j=1}^{5}\sum_{i=1}^{N}(\mathrm{y}_{i,j} - \hat{y}_{\ i,j}) \tag{6.2}$$

Where:

N: is the number of instances

$y_{i,j}$: is the self-reported personality trait (groud truth label)

$\hat{y}_{i,j}$: is the predicted personality trait

Apart from the loss metric used to train and evaluate the model, other metrics have been computed to extract better insights of our model performance. The following metrics will only be analyzed for the best-performing model configuration.

[3]Except for the preliminary model in which the batch size number was defined using grid search strategy

- **Participant error**: the participant error is computed as the mean square error of the median of the participant's chunk predictions.
- **Session error**: the session error is computed as the mean square error of the median of the participant's chunk prediction for a particular task session.
- **Task error**: the task error is computed as the mean square error of the median of the participant's chunk prediction for a particular task (e.g Talk).
- **Pearson Correlation Coefficient**: the Pearson Correlation Coefficient (ρ) is used to discover how strongly related are two sets of data. This coefficient is used to discover how our predictions are related to the self-reported personality labels (ground truth) and the peer-reported personality labels (first impression). The Pearson Correlation Coefficient (ρ) is computed using the Equation 6.3, that returns a value in range $[-1, 1]$. The sign of the coefficient ρ indicates the direction of the association between $\mathbf{x}$ and $\mathbf{y}$. The closer ρ is to 1 or -1, the stronger the correlation is between the two sets, whereas a value closer to 0 denotes poor correlation (no tendency between sets).

$$\rho_{x,y} = \frac{\text{cov}(x, y)}{\sigma_x \sigma_y} \tag{6.3}$$

Where:

cov: is the covariance

σ_x: is the standard deviation of x

σ_y: is the standard deviation of y

6.2 Preliminary experiment

This experiment consisted of using the pre-computed features of the FC layer (red box in Figure 6.1) before the output layer of the UDIVA's model [1] and add on top of that a sequential encoder block. The objective of this experiment is to observe if by increasing the time window the model could be improved by learning temporal patterns.

Two different encoder blocks were analyzed: a recurrent method and a self-attention-based method. The recurrent method selected is LSTM, using only

one layer with 128 hidden units. The self-attention-based method selected is the Transformer network with 2 Encoder layers, 4 attention heads and 128 units. For the LSTM method, the context vector is obtained using the last hidden state whereas for the Transformer model it is obtained by placing a Global Average Pooling layer. In both encoder models the output, a 128-dimensional vector is fed into a linear layer to output the OCEAN trait scores.

The window size or the number of time steps is obtained using a grid search strategy to find the best value. The range of search defined is between 2 and 12. As this model does not require a high computational cost, the best hyperparameters(batch size and learning rate) have been retrieved using a grid-search strategy.

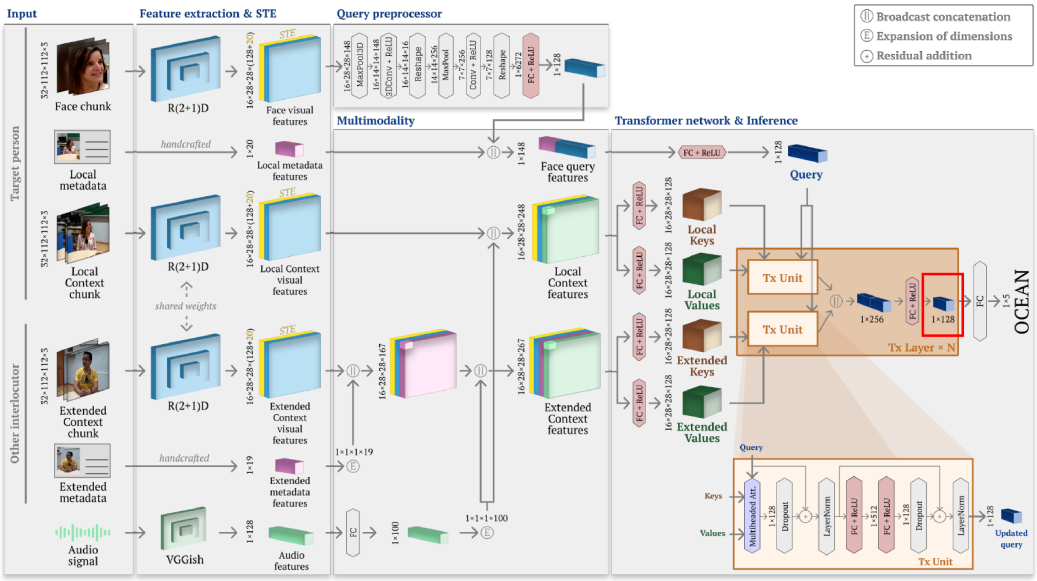


Figure 6.1: Features used for the preliminary experiment (red box)

To assess the performance we compare the evaluation metric (MSE loss) obtained in the validation and test set with the baseline model and the original UDIVA model. The best configuration results are obtained using LSTM as the encoder block, a window size of 12, a learning rate of 1e-3 and a batch size of 2048. The results obtained are shown in Table 6.1.

Although the new model improves the UDIVA's validation MSE, the model is not able to generalize the knowledge to the test error, achieving a worse performance compared to the original model.

Model	Validation MSE	Test MSE
Baseline	1.092	0.991
UDIVA's model	0.985	0.908
UDIVA's features + Sequential	0.928	0.952

Table 6.1: Results of the preliminary experiment

6.3 Visual experiment

In this experiment, we evaluate the performance of the model using only the visual modality and study how the performance varies with the number of timesteps selected.

The number of timesteps evaluated are 3, 6, 9, 12 and using only the visual feature extraction model without the encoder block. This is equivalent to approximately using 9, 18, 27, and 36 seconds for each timestep value and only 3 seconds when using only the visual modal without the encoder. It is important to mention that no timestep value greater than 12 has been analyzed since this is the limit to be able to use all the sessions of the training set. If we had used a larger number we would have lost training data. Furthermore, we must emphasize that the greater the number of timesteps, the greater the training time.

The architecture used for the sequential encoder in this experiment is the LSTM model. In Figure 6.2 the number of training samples is presented according to the number of timesteps selected (L). Obviously, the number of training instances decreases as the window size increase because for each input a greater number of instances are selected. But it is also important to emphasize that as the timestep value increases, the possibility of discarded samples due to temporary discontinuities between chunks also increases.

The results obtained using 50% of the test set are presented in Table 6.2. The worst result is obtained when a single chunk is used as input and the loss decreases as the time window increases, with the value of $L = 12$ being the one that achieves the best performance. This clearly indicates that the model improves by using a larger window as it is able to capture longer-term dependencies.

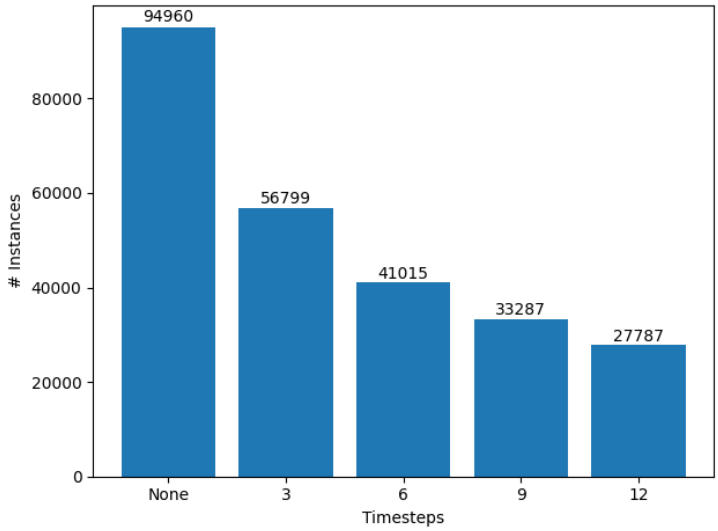


Figure 6.2: Number of training instances w.r.t to L value

Timesteps	Test MSE
None	0.934
3	0.903
6	0.886
9	0.860
12	**0.829**

Table 6.2: Experiment evaluating the visual modality performance using the MSE loss on the 50% of test set

6.4 Audio experiment

In this experiment, we evaluate the performance of the two proposed models for extracting the audio features: ResNet-18 and Vggish. As we explain in Section 5.1.2, the main difference between these models is that ResNet-18 is trained end-to-end using a fine-tuning strategy whereas the Vggish model is used as a feature extractor only as it is already trained on a huge audio dataset.

For this experiment, we only use the audio modality without adding the encoder block on top. The audio features extracted are fed into a FC layer before the last output layer that produces the OCEAN trait scores. The performance of these two models is evaluated using the MSE loss on 50% of

the test set.

The results of this experiment are presented in Table 6.3. The results clearly indicate that Vggish features outperform the ones extracted by the ResNet-18 (0.908 vs 0.942). Normally, we expect that an end-to-end model may outperform a frozen one. However, in this occasion, Vggish model (frozen) outperformed ResNet (end-to-end). A reason for this behavior might be the number of training samples in which the model has been trained. Vggish has been trained on a huge dataset (over 2 million training samples) whereas the ResNet-18 has only been trained using our training set which contains around 95k training samples. The difference is abysmal and in deep learning models, it is known that the amount of data can be a determining factor to obtain good performance.

Model	**Test MSE**
Vggish	**0.908**
Resnet18	0.942

Table 6.3: Audio models comparison using the MSE loss on the 50% of test set

After evaluating these two models, it is clear that the best audio feature extraction model is Vggish. Not only has greater performance but also requires much less training time as the model does not upload the weights.

6.5 Multimodal experiment

In this experiment, we evaluate the performance of all the models implemented by combining them to obtain the best multimodal configuration. The variations that are evaluated are the use of the Vggish and ResNet-18 model for the extraction of audio characteristics, as well as LSTM and Transformer for the sequential encoder block.

All configurations are evaluated with the same hyperparameters. The number of time steps selected is $L = 12$, since it is the value that achieved the highest performance in the previous experiment. Instead of retraining again from scratch the ResNet-18, for this experiment we take advantage of the checkpoints stored from the previous experiment to load the best ResNet-18 weights. This serves as a warm-up for the model, as it will still be trained in an end-to-end fashion without freezing any layer of the architecture (no fine-tuning).

Model	Test MSE
Audio(ResNet-18) + Visual + Encoder(Transformer)	0.856
Audio(ResNet-18) + Visual + Encoder(LSTM)	0.834
Audio(Vggish) + Visual + Encoder(Transformer)	0.838
Audio(Vggish) + Visual + Encoder(LSTM)	**0.784**

Table 6.4: Multimodal configurations comparison using the MSE loss on 50% of the test set

Observing the results presented in Table 6.4 we can corroborate that, even introducing visual modality and the sequential block, Vggish still outperforms ResNet-18 performance. On the other hand, the best-performing architecture for learning temporal patterns is LSTM. The difference between the performance of LSTM and Transformer is quite high, especially when using the Vggish model (0.784 vs 0.838).

This result may be surprising as Transformers have shown higher performance than LSTMs in fields such as NLP or action recognition. However, we must emphasize that the number of timesteps is quite low $L = 12$, so LSTM still can model these range-level dependencies. In addition, another key factor is the number of training samples as transformer-based architectures require a large amount of data to be trained. However, due to limited resources and time-constraints, the number of training samples is only 8336. Finally, the amount of configurable parameters that the Transformer network has is much larger than LSTM (i.e. positional encoding, number of heads, etc). For all these reasons Transformer architecture could not reach LSTM performance, but we strongly believe the performance gap between these two architectures could be reduced by a proper parameter selection and using a larger training set.

The best multimodal configuration is the one that uses the Vggish model to extract the audio characteristics and the LSTM to model the temporal patterns, using a window size of $\mathbf{L = 12}$.

6.6 Discussion

In this section, we discuss the results obtained using the best configuration for each modality and evaluate how the performance has evolved after each modality has been introduced.

We follow an incremental approach starting from the visual model only

and finishing with the multimodal model. As can be seen from Table 6.4, both the inclusion of the audio modality and the increase of the time window (sequential model) significantly improve the performance of the model. When only visual data is used and the model only captures the information within a 3 second time window, the model performs worse than if predicting the mean value (baseline). However, when we increase the time window from 3 seconds to 36 seconds the improvement is very noticeable, reducing the MSE from 0.926 to 0.829. This obviously indicates that the model benefits from capturing a larger range of visual cues. Lastly, the addition of sound modality generates a significant improvement by reducing the loss from 0.829 to 0.784. The improvement corroborates what we have seen in Section 3.2, audio features can accurately predict personality [29]. Especially if we remember the nature of the tasks, tasks such as Talk, Animals, or Ghost have verbal communication as their main component. In this type of task, the influence of the audio modality should be high due to the importance of capturing the verbal cues to infer the personality traits.

Model	Test MSE
Baseline	0.889
Visual features only	0.926
Visual features + Sequential	0.829
AudioVisual features + Sequential	**0.784**

Table 6.5: Modalities comparison using the MSE loss on the 50% of test set

In summary, the combination of visual and audio modalities allows us to capture the verbal and non-verbal cues that are essential to predict personality. Furthermore, by capturing a broader range of information, we can detect longer interaction patterns that could define the participant's personality.

6.6.1 Metrics analysis

In this section, we compare the performance obtained by our best model with UDIVA's model (theirs) and the Baseline model for all metrics described in Section 6.1.4. To fairly compare the models' performance the whole test set is used to evaluate the models.

The results obtained demonstrate that our model can improve the performance of a much complex model like UDIVA's [1] in terms of chunk error, session error, and task error while obtaining almost the same error on the participant

metric[4]. The results obtained are quite impressive if we recall that our model is trained using only the 30% of the training set and that the model is trained for all tasks, not per task like UDIVA's model does. This shows the influence of capturing longer patterns has on the performance of the model. Moreover, another key factor of the success of the model might be the end-to-end learning as the visual features can be fine-tuned during learning, thus increasing the adaptability of the visual feature extraction model.

	Chunk error	**Session Error**	**Task error**	**Participant Error**
Baseline	0.991	0.979	0.889	0.889
Theirs	0.908	0.920	0.833	0.812
Our	**0.853**	**0.909**	**0.808**	0.812

Table 6.6: Metrics analysis

6.6.2 Task error analysis

In this section we analyze the performance per task since each of the four tasks can show more remarkably some traits or specific behavior of the participant that can help infer its personality.

In Table 6.7 the test MSE loss of our best model, UDIVA's model (theirs), and the baseline (B) is shown per task. If we look at the overall performance, our model is able to outperform both in all tasks except for the *Lego* task. The poor performance in *Lego* task might be caused by the noisy sound of the Lego pieces while being moved. Another explanation could be that for this particular task, where movements and decision-making are fast, using a longer time window does not adequately capture fast-paced actions and decisions.

Analyzing the *Animals* task performance, we observe that our model achieves extremely good performance in the openness and agreeableness trait. The error obtained for the agreeableness trait is almost negligible (0.094). Recalling the theory, people that have a low value in this trait tend to be more competitive. One of the main objectives behind the design of this task was to foster competitiveness to see who guesses first the animal that is on his forehead. Our model seems to capture perfectly this trait, identifying how competitive the participants are. Openness trait is related to characteristics like imagination or creativity but we do not have a clear reason why in this particular task our model performs much better than the other models. Neuroticism has also

[4]The UDIVA's participant error is slightly better (0.8122) versus our model error (0.8125)

been significantly improved indicating that our model can capture behaviours like anxiety, stress. These patterns might be captured from the intonation of the participant, the pace of speech, or by detecting some gestures that might indicate nervousness or on the contrary the absence of movements indicating that the participant is more stable.

Analyzing the *Ghost* task performance, we do not observe any significant gains in any of the traits, and the overall score obtained is pretty similar to those obtained by the Baseline or UDIVA's model. The *Ghost* task was also designed to foster competitiveness but in this case, our model just slightly improves the agreeableness trait. One reason why our model does not significantly improve the performance on this task may be because the movements are at a very high pace and by using a large window size we fail to capture some fast-paced actions. Moreover, audio was not relevant either, as in this task there was no verbal communication.

Analyzing the *Lego* task performance, we can observe that our model performs even worse than the baseline. Possible reasons behind this poor performance are the fast-paced patterns might not be captured and the most likely reason is that for this particular task the audio modality might be unhelpful as probably the audio features are capturing the Lego pieces' sounds, which obviously do not have any relation with the personality of the participant and just add noise to the problem.

Lastly, for the *Talk* we can observe that our model achieves the best overall performance. The improvement in the extraversion trait error is quite noticeable. If we recall the theory, extraversion is characterized by talkativeness and sociability among many others. It is pretty reasonable that our model is able to capture this trait for this particular task as by using the audio features and visual features it might capture how extroverted the participant is by analyzing who prefers to start the conversations, who talks during more time, etc. Another feature that our model might be capturing is the response time, as it is known that extroverted people tend to respond faster [33] while introverted people carefully think things before speaking [31, 32]. Using a larger time window might also be beneficial for capturing these characteristics.

To conclude, if we look at the average error per trait among all the tasks the trait that our model captures the worst is extraversion. This is quite surprising as there are shreds of evidence in the literature that show that extraversion is typically the easiest trait to infer[61]. However, some studies have observed that the extraversion trait accuracy is affected by the window size when inferring the real personality recognition[62]. The window size selected for our

	Animals						Ghost					
	O	**C**	**E**	**A**	**N**	**Avg**	**O**	**C**	**E**	**A**	**N**	**Avg**
B	0.725	0.877	0.991	0.673	1.179	0.889	**0.725**	0.877	0.991	0.673	1.179	0.889
Theirs	0.747	**0.756**	**0.891**	0.579	1.021	0.799	0.742	0.894	**0.845**	0.667	1.139	0.857
Our	**0.208**	1.263	1.193	**0.094**	**0.727**	**0.697**	0.759	**0.804**	0.927	**0.627**	**1.09**	**0.841**

	Lego						Talk					
	O	**C**	**E**	**A**	**N**	**Avg**	**O**	**C**	**E**	**A**	**N**	**Avg**
B	**0.725**	0.877	0.991	0.673	1.179	0.889	**0.725**	0.877	0.991	0.673	1.179	0.889
Theirs	0.745	0.839	**0.953**	**0.66**	**1.099**	**0.859**	0.774	0.791	0.87	**0.669**	**0.986**	0.818
Our	0.887	**0.834**	0.992	0.738	1.139	0.918	0.809	**0.68**	**0.695**	0.681	1.015	**0.776**

Table 6.7: Task error analysis

model might not be the best for predicting this trait so further analysis could be done to select an appropriate time window. Finally, another differential point is that extraversion has strongly been associated with a facial expression like a smile. This could be a reason why UDIVA's model, which uses face visual features, obtains the best extraversion score in all tasks.

6.6.3 Correlation analysis

In this section, we analyze and discuss the per-trait correlation between the predictions and the ground truth labels. We compare the results obtained by our model with UDIVA's predictions and the peer-reported personality labels (FI, first impression). This allows us to assess whether AI can achieve human-level perception for personality recognition.

The correlation between sets is obtained using Pearson Correlation Coefficient (see Equation 6.3). Values close to ± 1 indicate that exists a strong correlation between the sets whereas values close to 0 indicate poor correlation. In the psychological field, when the coefficient is greater than 0.3 or less than -0.3, it is considered that there is a high correlation, especially when internal states such as personality are measured [63, 64].

As can be seen from Table 6.8, the highest coefficients are those obtained with the peer-reported values, except for the agreeableness trait in which our model obtains a higher correlation (0.45). Agreeableness seems to be the trait that our model captures best, not only in terms of MSE error but also if we consider the correlation compared to human performance. As mentioned previously, the extraversion trait is usually the easiest trait to infer and the results obtained by the peer-reported labels corroborate this claim, achieving the highest correlation value (0.74).

Looking at the coefficients obtained by our model, we can say that,exception

for the Openness trait (-0.27), all values indicate a strong correlation with the ground truth labels. Lastly, if we compare the results obtained by the AI models(our and theirs), our model achieves higher correlation coefficients in all traits.

	O	**C**	**E**	**A**	**N**
FI vs GT	**0.44**	**0.59**	**0.74**	0.37	**0.61**
Our AI vs GT	-0.27	0.5	0.41	**0.45**	0.53
Their AI vs GT	-0.08	0.4	0.39	0.22	0.44

Table 6.8: Correlation analysis with ground truth

Before ending the correlation analysis we evaluate how correlated are the AI predictions with the peer-reported labels. Even though the models are trained using the ground truth labels, in the literature, we have seen that apparent personality recognition is easier to capture. In Table 6.9, we show the per-trait correlation coefficients between the AI predictions(ours and theirs) and the peer-reported labels. The results clearly indicate that our model's predictions are more correlated with the FI labels compared to the UDIVA's predictions. However, the negative sign in the Openness coefficient indicates that our predictions are strongly related but the variables move in the opposite direction, which is not the expected behavior as both should move in the same direction (positive sign). On the other hand, if we compare the coefficients with those obtained in Table 6.8 we can see that for all traits our predictions are more correlated with the ground truth labels, except for the openness and extraversion trait. Nonetheless, in the case of the openness correlation coefficient, even having a stronger correlation, due to the opposite direction it is better the result obtained in Table 6.8 even if it is less correlated.

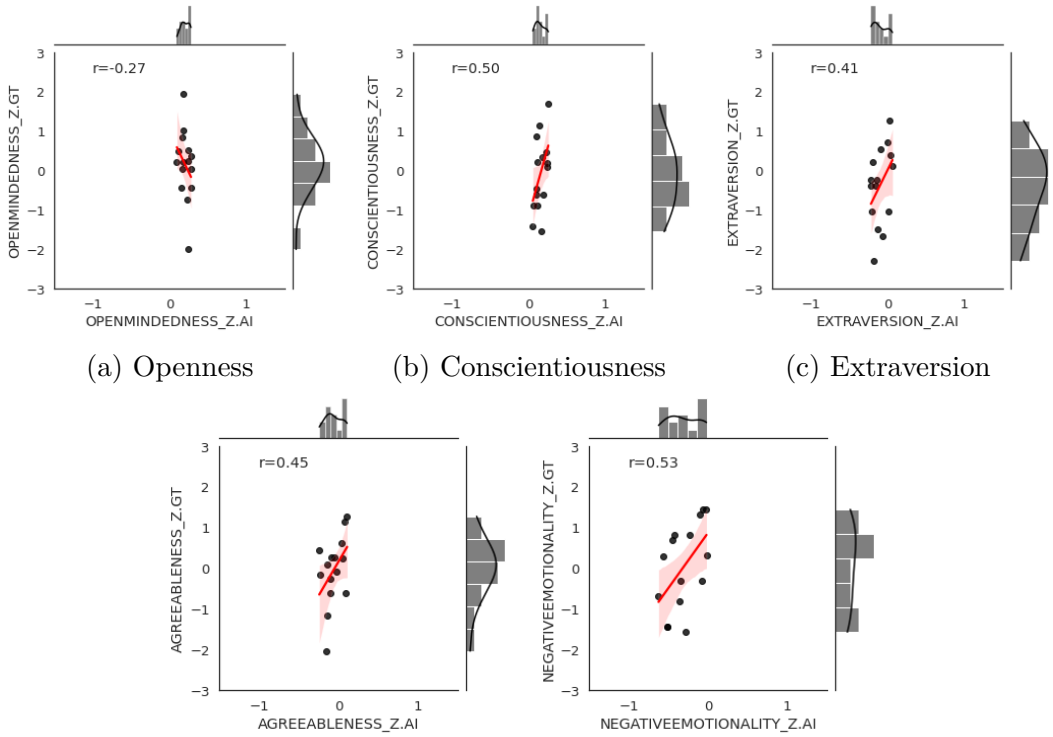
	O	**C**	**E**	**A**	**N**
Our AI vs FI	-0.71	**0.38**	**0.47**	**0.41**	**0.51**
Theirs AI vs FI	**0.17**	0.33	0.34	0.02	0.15

Table 6.9: Correlation analysis with first impression

Correlation is another metric to assess the performance of our models, but in order to get a better idea of what our model predicts in Figure 6.3 we show the per-trait correlations between our predictions and the ground truth labels, as well as the per-trait distribution of both sets.

Looking at Figure 6.3 we can conclude that our model suffers from the **regression toward the mean** problem. This can be easily appreciated by

comparing the distribution of our predictions with the ground truth labels. In all traits, our predictions are densely concentrated around the mean ground truth trait value. If we observe the values of the X-axis, which are the AI trait scores, we can see that all predictions reside in a really small range [-0.5, 0.5] as opposed to the real scores which have a much wider range [-2.5, 2]. This behavior shows that our model could benefit by using some kind of constraint or regularization in the definition of the training loss so that it tries to use a greater range of values and move away from the mean, as was done in [65].



(a) Openness (b) Conscientiousness (c) Extraversion

(d) Agreeableness (e) Neuroticism

Figure 6.3: Per-trait correlations between the ground truth labels(Y-axis) and the AI predictions (X-axis)

Chapter 7

Conclusions

In this Master's thesis, we have presented a multimodal model that extracts audiovisual features using state-of-the-art methods to infer the personality of a target person in a dyadic scenario. Our model is trained on the UDIVA dataset, a multimodal dataset of face-to-face dyadic interactions recorded in different contexts, each of them related to a different collaborative/competitive task.

We have gradually studied how the inclusion of each modality affects the performance of the model. Starting only by using the visual features in a short time interval(3 seconds), then increasing the interval up to 36 seconds, and finally adding the audio modality. We have found consistent improvements in each of the modifications, obtaining the best model when combining the audio and video modality using a time window of 36 seconds($L = 12$). The results obtained have corroborated our two main hypotheses:

- Performing the predictions using a larger range of visual and acoustic cues makes the model capable of capturing behaviors of longer duration and therefore obtain a performance improvement.
- The combination of audio and visual modality makes a stronger personality regressor as it is able to capture verbal (i.e. intonation, speed, volume) and non-verbal (i.e. gestures, facial expressions, body movement) behaviors that are strongly related to the personality.

The best model has been able to improve the performance of UDIVA's model in most of the metrics while being trained with only 30 percent of the training set, in a multi-task manner and using fewer modalities, which indicates that there is still room for improvement. We must emphasize the

importance of using a broader range of visual and acoustic cues, as it has undoubtedly been the contribution that has brought the greatest performance improvement.

Nonetheless, even though the model performance has been pretty good, we have found that most of the predictions are highly concentrated around the ground truth mean trait value, which indicates that our model suffers from what is known as regression to the mean problem. Different proposals to mitigate this problem are presented as future work in Chapter 8.

Regarding the implementation of the model, we would like to emphasize the importance of the transfer learning strategy for this project. Using models that have been previously trained with a dataset in a domain similar to ours has allowed us to obtain great results taking into account the limited computational resources and the size of the training set that we had for models of such complexity.

To conclude, I believe this thesis shows many of the skills learned during the Master in Artificial Intelligence. Concepts like data preprocessing, implementation of deep learning models, or data visualization have been essential to carry out this project successfully.

Chapter 8

Future work

The results of this thesis together with the corresponding analysis have opened new lines of work to improve the performance of our model. As future work, we suggest the following lines of work:

- Regarding the modalities used, we suggest adding the transcriptions which are currently being annotated as well as the metadata from both participants and the context data from the other participant. An interesting and more computationally efficient approach could be to use as context only the personality traits of the other participant, since studies show that our behavior can be influenced by the actions of the person with whom we are interacting.

- The redesign of the loss function to tackle the regression to the mean problem. One approach that could be worth trying is use to the Bell loss[65]. Furthermore, literature studies suggest that there are small and moderate correlations between traits [66]. The weighting of the correlation between traits could also be introduced in combination with the Mean Squared error to reduce the regression to the mean problem.

- Increase the features extracted from each modality. For the visual modality, we suggest extracting facial features as they have proved to provide relevant information for inferring personality. For the audio modality, we suggest combining the audio feature learning model with hand-crafted audio features that are widely used in the literature like Zero Crossing Rate, Energy, MFCC among others.

- Train with 100 percent of training data, it has not been done due to time restrictions and because with a proper sampling of data the results

obtained were already good but for architectures like transformers that are data-hungry this could improve the performance and even outperform LSTM. Another strategy that could be studied is to use self-supervised learning to pre-train the transformer layers.

- Weighting the traits per modality and task. Some studies have shown that some modalities are better at predicting some traits than others [40]. During the fusion mechanism, we could give more importance to one modality or another depending on the task or the trait.

Automatic personality recognition is an area that is still in its early stages and there are endless lines of work to improve the performance of these models. For this reason, we would like to encourage anyone to continue investigating this challenging area with so many possibilities for improvement.

Bibliography

[1] C. Palmero, J. Selva, S. Smeureanu, J. C. S. Jacques Junior, A. Clapés, A. Moseguí, Z. Zhang, D. Gallardo, G. Guilera, D. Leiva, and S. Escalera, "Context-Aware Personality Inference in Dyadic Scenarios: Introducing the UDIVA Dataset," Tech. Rep. [Online]. Available: https://ec. iii, iii, 2, 18, 21, 22, 23, 24, 25, 39, 45

[2] J. S. Wiggins, *The five-factor model of personality: Theoretical perspectives*. Guilford Press, 1996. 1

[3] G. W. Allport, "Personality: A psychological interpretation." 1937. 1

[4] R. B. Cattell, *Description and measurement of personality*, 1946. [Online]. Available: https://psycnet.apa.org/record/1947-00501-000 1

[5] H. J. Eysenck, *Dimensions of personality*. Transaction Publishers, 1950, vol. 5. 1

[6] K. Cherry, "What Are the Big 5 Personality Traits?" [Online]. Available: https://www.verywellmind.com/the-big-five-personality-dimensions-2795422 1, 2

[7] A. Subramaniam, V. Patel, A. Mishra, P. Balasubramanian, and A. Mittal, "Bi-modal First Impressions Recognition using Temporally Ordered Deep Audio and Stochastic Visual Features," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9915 LNCS, pp. 337–348, 10 2016. [Online]. Available: http://arxiv.org/abs/1610.10048 2, 17, 18

[8] C. L. Zhang, H. Zhang, X. S. Wei, and J. Wu, "Deep bimodal regression for apparent personality analysis," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9915 LNCS.

Springer Verlag, 2016, pp. 311–324. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-49409-8_25 2, 17

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998. 5

[10] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," pp. 436–444, 5 2015. [Online]. Available: https://www.nature.com/articles/nature14539 6, 37, 38

[11] "CS231n Convolutional Neural Networks for Visual Recognition." [Online]. Available: https://cs231n.github.io/convolutional-networks/ 7

[12] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986. [Online]. Available: https://www.nature.com/articles/323533a0 8, 9

[13] Y. Bengio, P. Simard, and P. Frasconi, "Learning Long-Term Dependencies with Gradient Descent is Difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994. 8

[14] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 11 1997. [Online]. Available: http://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf 9

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, [U+FFFD] Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 2017-December. Neural information processing systems foundation, 6 2017, pp. 5999–6009. [Online]. Available: https://arxiv.org/abs/1706.03762v5 10, 18, 31

[16] 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," 10 2020. [Online]. Available: http://arxiv.org/abs/2010.11929 10

[17] G. Bertasius, H. Wang, and L. Torresani, "Is Space-Time Attention All You Need for Video Understanding?" 2 2021. [Online]. Available: http://arxiv.org/abs/2102.05095 10

[18] J. Willis and A. Todorov, "First impressions: Making up your mind after a 100-ms exposure to a face," *Psychological Science*, vol. 17, no. 7, pp. 592–598, 7 2006. [Online]. Available: https://journals.sagepub.com/doi/10.1111/j.1467-9280.2006.01750.x 15

[19] V. Bruce and A. Young, "Understanding face recognition," *British Journal of Psychology*, vol. 77, no. 3, pp. 305–327, 1986. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/3756376/ 15

[20] C. Ventura, D. Masip, and A. Lapedriza, "Interpreting CNN Models for Apparent Personality Trait Regression," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2017-July. IEEE Computer Society, 8 2017, pp. 1705–1713. 15

[21] Y. Yan, J. Nie, L. Huang, Z. Li, Q. Cao, and Z. Wei, "Exploring relationship between face and trustworthy impression using mid-level facial features," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9516. Springer Verlag, 2016, pp. 540–549. 15

[22] F. Gürpınar, H. Kaya, and A. A. Salah, "Combining deep facial and ambient features for first impression estimation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9915 LNCS. Springer Verlag, 2016, pp. 372–385. 15

[23] F. De La Torre, W. S. Chu, X. Xiong, F. Vicente, X. Ding, and J. Cohn, "IntraFace," in *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, FG 2015*, vol. 1. Institute of Electrical and Electronics Engineers Inc., 7 2015. [Online]. Available: /pmc/articles/PMC4918819//pmc/articles/PMC4918819/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC4918819/ 15

[24] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," 2015. 16

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on*

Learning Representations, ICLR 2015 - Conference Track Proceedings. International Conference on Learning Representations, ICLR, 9 2015. [Online]. Available: http://www.robots.ox.ac.uk/ 16, 29

[26] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," in *IEEE International Conference on Neural Networks - Conference Proceedings*, vol. 2, 2004, pp. 985–990. 16

[27] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A Closer Look at Spatiotemporal Convolutions for Action Recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 6450–6459, 11 2017. [Online]. Available: http://arxiv.org/abs/1711.11248 16, 18, 27

[28] 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*, 2017, pp. 6299–6308. 16, 27

[29] T. Polzehl, S. Möller, and F. Metze, "Automatically assessing personality from speech," in *Proceedings - 2010 IEEE 4th International Conference on Semantic Computing, ICSC 2010*, 2010, pp. 134–140. 16, 45

[30] J. Park, S. Lee, K. Brotherton, D. Um, and J. Park, "Identification of speech characteristics to distinguish human personality of introversive and extroversive male groups," *International Journal of Environmental Research and Public Health*, vol. 17, no. 6, 3 2020. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/32210035/ 16

[31] A. W. Siegman and B. Pope, "Effects of question specificity and anxiety-producing messages on verbal fluency in the initial interview," *Journal of Personality and Social Psychology*, vol. 2, no. 4, pp. 522–530, 10 1965. [Online]. Available: /record/1966-00509-001 16, 47

[32] R. W. Ramsay, "Speech Patterns and Personality," *Language and Speech*, vol. 11, no. 1, pp. 54–63, 8 1968. [Online]. Available: https://journals.sagepub.com/doi/10.1177/002383096801100108 16, 47

[33] J. Brebner and C. Cooper, "The effect of a low rate of regular signals upon the reaction times of introverts and extraverts," *Journal of Research in Personality*, vol. 8, no. 3, pp. 263–276, 10 1974. 16, 47

[34] J. Stahl and T. Rammsayer, "Extroversion-related differences in speed of premotor and motor processing as revealed by lateralized readiness potentials," *Journal of Motor Behavior*, vol. 40, no. 2, pp. 143–154, 3 2008. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/18400680/ 16

[35] M. A. Carbonneau, E. Granger, Y. Attabi, and G. Gagnon, "Feature Learning from Spectrograms for Assessment of Personality Traits," *IEEE Transactions on Affective Computing*, vol. 11, no. 1, pp. 25–31, 1 2020. [Online]. Available: http://arxiv.org/abs/1610.01223http://dx.doi.org/10.1109/TAFFC.2017.2763132 17

[36] G. Mohammadi and A. Vinciarelli, "Automatic personality perception: Prediction of trait attribution based on prosodic features," *IEEE Transactions on Affective Computing*, vol. 3, no. 3, pp. 273–284, 2012. 17

[37] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. Weiss, and K. Wilson, "CNN Architectures for Large-Scale Audio Classification," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017. [Online]. Available: https://arxiv.org/abs/1609.09430 17, 18, 28

[38] S. Aslan and U. Güdükbay, "Multimodal Video-based Apparent Personality Recognition Using Long Short-Term Memory and Convolutional Neural Networks," 11 2019. [Online]. Available: http://arxiv.org/abs/1911.00381 17

[39] "ChaLearn First impressions (ECCV '16, ICPR '16)," 2016. [Online]. Available: https://chalearnlap.cvc.uab.cat/dataset/20/description/ 17

[40] O. Kampman, E. J. Barezi, D. Bertero, and P. Fung, "Investigating Audio, Visual, and Text Fusion Methods for End-to-End Automatic Personality Prediction," *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, vol. 2, pp. 606–611, 5 2018. [Online]. Available: http://arxiv.org/abs/1805.00705 17, 54

[41] X.-S. Wei, C.-W. Xie, and J. Wu, "Mask-CNN: Localizing Parts and Selecting Descriptors for Fine-Grained Image Recognition," 5 2016. [Online]. Available: http://arxiv.org/abs/1605.06878 17

[42] X. S. Wei, J. H. Luo, J. Wu, and Z. H. Zhou, "Selective Convolutional Descriptor Aggregation for Fine-Grained Image Retrieval," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2868–2881, 6 2017. 17

[43] V. Ponce-López, B. Chen, M. Oliu, C. Corneanu, A. Clapés, I. Guyon, X. Baró, H. J. Escalante, and S. Escalera, "Chalearn LAP 2016: First round challenge on first impressions - Dataset and results," vol. 9915 LNCS. Springer Verlag, 2016, pp. 400–418. 17

[44] D. Ghadiyaram, M. Feiszli, D. Tran, X. Yan, H. Wang, and D. Mahajan, "Large-scale weakly-supervised pre-training for video action recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 12 038–12 047, 5 2019. [Online]. Available: http://arxiv.org/abs/1905.00561 18, 27

[45] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "YouTube-8M: A Large-Scale Video Classification Benchmark," 9 2016. [Online]. Available: http://arxiv.org/abs/1609.08675 18, 29

[46] M. K. Rothbart, S. A. Ahadi, K. L. Hershey, and P. Fisher, "Investigations of temperament at three to seven years: The children's behavior questionnaire," pp. 1394–1408, 9 2001. [Online]. Available: https://srcd.onlinelibrary.wiley.com/doi/full/10.1111/1467-8624.00355https://srcd.onlinelibrary.wiley.com/doi/abs/10.1111/1467-8624.00355https://srcd.onlinelibrary.wiley.com/doi/10.1111/1467-8624.00355 21

[47] L. K. Ellis and M. K. Rothbart, "Revision of the early adolescent temperament questionnaire," in *Poster presented at the 2001 biennial meeting of the society for research in child development, Minneapolis, Minnesota*, 2001. 22

[48] C. J. Soto and O. P. John, "The next Big Five Inventory (BFI-2): Developing and assessing a hierarchical model with 15 facets to enhance bandwidth, fidelity, and predictive power," *Journal of Personality and Social Psychology*, vol. 113, no. 1, pp. 117–143, 7 2017. [Online]. Available: /record/2016-17156-001 22, 35

[49] M. C. Ashton and K. Lee, "The HEXACO-60: A short measure of the major dimensions of personality," *Journal of Personality Assessment*, vol. 91, no. 4, pp. 340–345, 2009. 22

[50] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman, "Video Action Transformer Network," 12 2018. [Online]. Available: http://arxiv.org/abs/1812.02707 24

[51] C. Palmero, J. Selva, S. Smeureanu, J. C. S. Jacques Junior, A. Clapés, A. Moseguí, Z. Zhang, D. Gallardo, G. Guilera, D. Leiva, and S. Escalera, "Supplementary Material-Context-Aware Personality Inference in Dyadic Scenarios: Introducing the UDIVA Dataset," Tech. Rep. 25

[52] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks," 2015. 27

[53] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The Kinetics Human Action Video Dataset," 5 2017. [Online]. Available: http://arxiv.org/abs/1705.06950 27

[54] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. F. Li, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 9 2014, pp. 1725–1732. 27

[55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December. IEEE Computer Society, 12 2016, pp. 770–778. [Online]. Available: http://image-net.org/challenges/LSVRC/2015/ 28, 29

[56] G. Ke, D. He, and T.-Y. Liu, "Rethinking Positional Encoding in Language Pre-training," 6 2020. [Online]. Available: http://arxiv.org/abs/2006.15595 31

[57] L. Teijeiro-Mosquera, J. I. Biel, J. L. Alba-Castro, and D. Gatica-Perez, "What your face vlogs about: Expressions of emotion and big-five traits impressions in youtube," *IEEE Transactions on Affective Computing*, vol. 6, no. 2, pp. 193–205, 4 2015. 36

[58] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, 12 2015. [Online]. Available: https://arxiv.org/abs/1412.6980v9 37

[59] D. R. Wilson and T. R. Martinez, "The general inefficiency of batch training for gradient descent learning," *Neural Networks*, vol. 16, no. 10, pp. 1429–1451, 12 2003. 37

[60] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima," *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 9 2016. [Online]. Available: http://arxiv.org/abs/1609.04836 37

[61] J.-I. Biel, O. Aran, and D. Gatica-Perez, "You Are Known by How You Vlog: Personality Impressions and Nonverbal Behavior in YouTube," Tech. Rep. 1, 7 2011. [Online]. Available: www.aaai.org 47

[62] B. Lepri, R. Subramanian, K. Kalimeri, J. Staiano, F. Pianesi, and N. Sebe, "Employing social gaze and speaking activity for automatic determination of the Extraversion trait," in *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction, ICMI-MLMI 2010*, 2010. 47

[63] L. Liu, D. Preotiuc-Pietro, Z. R. Samani, M. Moghaddam, and L. Ungar, "Analyzing Personality through Social Media Profile Picture Choice," *undefined*, 2016. 48

[64] G. J. Meyer, S. E. Finn, L. D. Eyde, G. G. Kay, K. L. Moreland, R. R. Dies, E. J. Eisman, T. W. Kubiszyn, and G. M. Reed, "Psychological testing and psychological assessment: A review of evidence and issues," *American Psychologist*, vol. 56, no. 2, pp. 128–165, 2001. 48

[65] Y. Li, J. Wan, Q. Miao, S. Escalera, H. Fang, H. Chen, X. Qi, and G. Guo, "CR-Net: A Deep Classification-Regression Network for Multimodal Apparent Personality Analysis," *International Journal of Computer Vision*, vol. 128, no. 12, pp. 2763–2780, 12 2020. [Online]. Available: https://link.springer.com/article/10.1007/s11263-020-01309-y 50, 53

[66] R. A. Power and M. Pluess, "Heritability estimates of the Big Five personality traits based on common genetic variants," *Translational Psychiatry*, vol. 5, no. 7, p. e604, 7 2015. [Online]. Available: /pmc/articles/PMC5068715//pmc/articles/PMC5068715/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC5068715/ 53

[67] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio Set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[68] 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 Visual Transformer," 12 2020. [Online]. Available: http://arxiv.org/abs/2012.12556

[69] H. J. Escalante, H. Kaya, A. A. Salah, S. Escalera, Y. Gucluturk, U. Guclu, X. Baro, I. Guyon, J. J. Junior, M. Madadi, S. Ayache, E. Viegas, F. Gurpinar, A. S. Wicaksana, C. C. S. Liem, M. A. J. van Gerven, and R. van Lier, "Explaining First Impressions: Modeling, Recognizing, and Explaining Apparent Personality from Videos," 2 2018. [Online]. Available: http://arxiv.org/abs/1802.00745

[70] J. C. Silveira Jacques Junior, Y. Gucluturk, M. Perez, U. Guclu, C. Andujar, X. Baro, H. J. Escalante, I. Guyon, M. A. J. Van Gerven, R. Van Lier, and S. Escalera, "First Impressions: A Survey on Vision-based Apparent Personality Trait Analysis," *IEEE Transactions on Affective Computing*, pp. 1–1, 7 2019.

[71] J. C. S Jacques Junior, C. Andujar, X. BaróBar, H. Jair Escalante, I. Guyon, M. A. J van Gerven, R. van Lier, S. Escalera, and H. Jair Escalante is, "First Impressions: A Survey on Vision-based Apparent Personality Trait Analysis," Tech. Rep. [Online]. Available: https://www.theguardian.com/technology/2017/apr/13/

[72] L. Cun, J. Henderson, Y. Le Cun, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Handwritten Digit Recognition with a Back-Propagation Network," Tech. Rep.

[73] L. V. Phan and J. F. Rauthmann, "Personality computing: New frontiers in personality assessment," 2021. [Online]. Available: https://doi.org/10.1111/spc3.12624

[74] Y. Mehta, N. Majumder, A. Gelbukh, and E. Cambria, "Recent trends in deep learning based personality detection," *Artificial Intelligence Review*, vol. 53, no. 4, pp. 2313–2339, 4 2020.

[75] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-Alone Self-Attention in Vision Models," 6 2019. [Online]. Available: http://arxiv.org/abs/1906.05909

[76] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in Vision: A Survey," 1 2021. [Online]. Available: http://arxiv.org/abs/2101.01169