POLITECNICO
MILANO 1863

Dipartimento di Elettronica, Informazione e Bioingegneria

Master Degree in Music and Acoustic Engineering

Audio-Video Deepfake Detection through Emotion Recognition

by:
Jacopo Gino

matr.:
921407

Supervisor:
Paolo Bestagini

Co-supervisor:
Davide Salvi

Academic Year
2020-2021

POLITECNICO
MILANO 1863

Dipartimento di Elettronica, Informazione e Bioingegneria

Master Degree in Music and Acoustic Engineering

Rilevamento di deepfake audio-video tramite riconoscimento delle emozioni automatico

Candidato:
Jacopo Gino

matr.:
921407

Relatore:
Paolo Bestagini

Co-relatore:
Davide Salvi

Anno Accademico
2020-2021

Abstract

In recent years, techniques for synthetic media generation have seen huge advancements. With the powerful tools provided by state-of-the-art artificial intelligence approaches, it is now possible to generate audio and visual content so accurately as to be able to deceive human sight and hearing.

These new machine-generated media are known as deepfakes. Although benign and harmless applications can not be overlooked, they immediately raised ethical and legal concerns. As they allow to alter both voice and visual identities of portrayed subjects, some of their malevolent uses may lead to severe consequences such as fake news spreading, falsifying legal proofs, or new forms of blackmail and fraud. Therefore, developing robust and reliable deepfake detection systems is compelling and essential for both the individual and society.

In this thesis, we propose a method for deepfake detection using both audio and video signals. The underlying assumption of the present work is that machines can not recreate emotions in altered or generated subjects as real humans genuinely convey them. For this reason, we adapted neural network-based techniques from the emotion recognition field to this task.

Results show that audio-based techniques detect altered media more accurately than video-based approaches. However, we obtain the best classification results when we adopt a multimodal approach, considering the audio and video modalities together.

Sommario

Negli ultimi anni le tecniche per generare contenuti multimediali sintetici hanno avuto un notevole miglioramento. Con i potenti strumenti forniti da applicazioni di intelligenza artificiale, è ora possibile generare materiali audiovisivi in modo così accurato da poter ingannare i sensi umani di vista e udito.

Questi nuovi media generati da macchine vengono chiamati deepfake. Nonostante i deepfake possano dare vita a nuovi stimolanti scenari futuri, questi media hanno da subito suscitato preoccupazioni sia etiche che legali. Permettendo di alterare le identità vocali e visive delle persone ritratte, alcuni dei utilizzi potrebbero avere gravi conseguenze come la diffusione di fake news, falsificazione di prove legali, nuove forme di frode e ricatto. È quindi indispensabile e urgente sviluppare sistemi di rilevamento dei deepfake che siano attendibili e robusti, per l'individuo e la società.

In questa tesi, proponiamo un metodo multimodale per il rilevamento dei deepfake, basato sull'analisi simultanea di audio e video. L'ipotesi su cui si basa questo lavoro è che l'intelligenza artificiale sia in grado di ricreare nei soggetti rappresentati aspetti di basso livello, ma non riesca a riprodurre aspetti più complessi come le emozioni. Per fare ciò abbiamo adattato a questo obiettivo tecniche di riconoscimento automatico delle emozioni basate su reti neurali.

I risultati mostrano che le tecniche basate sull'audio individuano i media alterati più accuratamente delle tecniche basate sul video. Tuttavia, i migliori risultati nella classificazione vengono ottenuti con un approccio mulimodale, quando consideriamo le modalità audio e video assieme.

Ringraziamenti

Ringrazio innanzitutto Fabio, Clara, Paolo e Davide per l'aiuto e le direzioni fornitemi durante tutti i mesi del lavoro. Paolo e Davide, grazie per la disponibilità e pazienza, letteralmente fino all'ultimo minuto.

Iaia, Nonna, mi avete visto iniziare, ma non finire. Sembra incredibile, ma ho finito ora!

Ringrazio chi ha iniziato l'intero percorso di questi anni con me: Cine, Rav e Pablo. Ringrazio Ema, compagno di studio degli ultimi due anni e con te Nico, grazie del perenne supporto e attenzione. Ringrazio gli amici di una vita: Ale Z., Cami, Saci, Leo, Fra, ciao Ste, Simo. E quelli che mi sono stati più vicino nelle ultime fasi della stesura della tesi: Vali, Edo, Ale V., Vero, Pello, Fede, grazie. Arya, Gabri, ci siete anche voi, come sarei arrivato qui senza aver suonato così tanto. E poi Ema, Ludo, Giulio ti ringrazio qui, befane. Gabbo!

Ringrazio poi chi ha reso questi ultimi mesi così speciali. Berte, Elo, la vostra terra cruda mi ha accompagnato ogni giorno della scrittura e Ade, grazie della vostra arte. Fefè, Caro, Giudi, via Farini. Grazie, artisti, di insegnarmi così tanto e di regalarmi così tanto calore.

Manu, sei il mio riferimento per la programmazione dai tempi del progetto con Anto. Gabriele, quell'incontro in aereoporto, grazie di avermi segnalato l'imminenza delle scadenze, in qualche modo sembra che ci sia arrivato.

Credo di aver citato tutti. Ah, sì. Mamma, papà, la tesi è dedicata a voi.

Ciao!

Nogi

Contents

List of Figures

List of Tables

Glossary

3DCNN 3-Dimensional Convolutional Neural Network. vii, 11, 22, 23, 33, 37, 55, 57–62

AI Artificial Intelligence. 1, 13, 64

ANN Artificial Neural Network. 1, 4

AUC Area Under the Curve. 50

BA Balanced Accuracy. 49, 52

BW Black and White. 33

CGI Computer Generated Images. 14

CNN Convolutional Neural Network. 2, 9, 12, 22–28, 55

DFAE Deepfake Autoencoder. 31, 44

DFDC Deepfake Detection Challenge dataset. 28, 43–46, 52, 56, 58–62

ER Emotion Recognition. 3, 14, 15, 65

FFT Fast Fourier Transform. 19

FPR False Positive Rate. 50

GAN Generative Adversarial Network. 12, 13, 25, 26, 31, 44

HNR Harmonics to Noise Ratio. 21

LGBM Light Gradient Boosting Machine. 42, 58, 59, 61, 62

LLD Low Level Descriptor. 19, 20, 28, 32, 39, 41

logMMSE log-spectral Amplitude MMSE. 19

LPC Linear Prediction Coefficients. 21

LPCC Linear Prediction Cepstral Coefficients. 21

LSTM Long Short-Term Memory Recurrent Neural Network. vii, 7–9, 22–25, 27, 28, 32, 34, 36, 38, 52–54, 58, 65

MFCC Mel Frequency Cepstral Coefficients. 21, 53

ML Machine Learning. 42

MLP Multi-layer Perceptron. vii, 5, 12, 25, 27, 52, 53

MMSE Minimum Mean Square Error. 19

NN Neural Network. 4, 6, 8–10, 23

RGB Red, Green, Blue. 33

RNN Recurrent Neural Network. vii, 6–9, 34

ROC Receiver Operating Characteristic. 50, 61, 62

SER Speech Emotion Recognition. 3, 15, 18–20, 22, 23, 51, 52, 56, 58, 64, 65

SSD Single Shot Detector. 41

SVM Support-Vector Machine. 27, 59

TNR True Negative Rate. 49

TPR True Positive Rate. 49, 50

1

Introduction

In recent years, a new kind of Artificial Intelligence (AI)-generated media have attracted widespread attention. These are called deepfakes. The term comes from a crasis between "deep learning" and "fake" since this fake contents are created by deep learning algorithms, a class of machine learning techniques that relies on Artificial Neural Networks (ANNs). We can use these networks to perform highly realistic audio and video manipulations, altering the visual and speech identities of portrayed subjects. So far, the most targeted categories are celebrities and politicians. This is because deep learning techniques require a large amount of data to perform accurately, and these two are categories with vast visibility. Therefore, content with their representations is easily collectible both for accessibility and quantity. Although deepfake algorithms themselves, like all technologies, have no good or evil attributes, these have been widely used for harmful purposes that can have severe consequences. Some examples concern the spreading of fake news [18], creation of revenge porn videos [19], and fraud cases [20]. This technology is closely related to the ethical, philosophical in a broader sense, comprehensive discussion

on the uses of artificial intelligence. To prevent it from threatening both the individual and society, the research community developed a series of detection methods along with large-scale benchmarks [21].

The visual manipulations of these technologies are typically achieved in two ways. The technique called face-swapping consists in swapping person identities in two videos. The first proposed method was generated by a Reddit user in December of 2017. It is now the most common identity manipulation form, especially for entertainment purposes. The second is referred to as face reenactment. This category of algorithms attempts to control people's expressions in videos, allowing fake creators to generate clips in which someone does something that never happened. On the other hand, synthetic audio is nearer to everyday situations and may be easier to understand both as a tool or a threat. We often overlook the pervasiveness of machine-generated speech. Still, we can find it in audiobooks, in virtual assistants like Siri, Alexa and Google Home. We can hear synthetic announcements in cars, bus stops, train stations, or typical call centers. The outcomes of nowadays techniques can be utterly realistic as to be able to pose evident threats to biometric authentication systems based on voice. In 2019 it was highlighted in [22] how this kind of fake media occasionally contains spatio-temporal glitches. However, it was already clear that these glitches could not be a reliable cue to detect fake media, as new and more accurate deepfake generators were (and are) continuously being developed.

In the past few years, state-of-the-art detection approaches moved toward deep learning to address these problems. As Convolutional Neural Networks (CNNs) are among the best architectures to deal with image data, many approaches to visual manipulation leverage this technique. The authors of [23] implement a CNN model aiming to generalize the semantic content of learned features used by the network to discriminate between real and fake images. To perform the same discrimination in [24] the authors feed their convolutional model with pairs of genuine and altered photos to learn comparison features between the two classes. Recently [25] proposed an ensemble of different trained CNN models to perform face manipulation detection. Some approaches, like the one used by the [26] authors, use semantic features obtained from both audio and video signals to perform the classification. However, current detection

methods are still insufficient to be applied in real scenarios. Further research is needed to increase the generalization and robustness of the developed techniques.

In this work, we propose a method to identify whether a video is genuine or has been altered or generated with a synthetic approach. To perform this classification, we leverage the assumption that synthesis algorithms can reconstruct low-level characteristics of voice and face appearances, but fail to recreate more complex aspects, such as emotions. This approach already proved successful [26]. Therefore, we adapt Emotion Recognition (ER) neural network approaches both for video and audio signals to deepfake detection. Firstly, we train our models on predicting emotions. Then, from the internal layers of the networks, we extract representations of time evolution characteristics of emotions from videos of which we want to infer the nature. These features are then classified as derived by real or fake emotion displays.

The experiments aim to verify the performances of the proposed model. The obtained results show promising performances with a balanced accuracy for detecting genuine or altered content reaching up to a value of 0.9532.

This thesis is organized as follows. In Chapter 2 we provide the reader with the basic knowledge on techniques and methods used to better understand what will be discussed in the following chapters. In Chapter 3 we give an overview of the main state-of-the-art approaches related to this work. We describe some remarkable studies related to Speech Emotion Recognition (SER), video emotion recognition, and deepfake detection. In Chapter 4 we give a formal definition of the problem we tackle in this thesis. We describe the architecture we propose to solve it and provide all its details. In Chapter 5 we describe the metrics used to evaluate our experiments, we provide a detailed description of the experiments we conducted and their results, along with the metrics used to evaluate them. Finally, in Chapter 6 we summarize the work done and suggest possible future improvements.

2

Theoretical Background

This chapter describes the theoretical background of the techniques used in the work and the key ideas behind it, providing the reader the basic knowledge to better understand what we will discuss in the following chapters.

2.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) or simply Neural Networks (NNs) are computational systems used for problem-solving, planning, learning and many more artificial intelligence applications. They consist of a structure of computational blocks called hidden layers, where each of them is composed of a set of so-called artificial neurons we also call perceptrons.

As biological brains, structures which they are remotely inspired on, NNs can learn representations of data, passed to the first net layer, the input layer, as vectors shaped accordingly to the input layer's required shape. In the machine learning field with representation learning, we

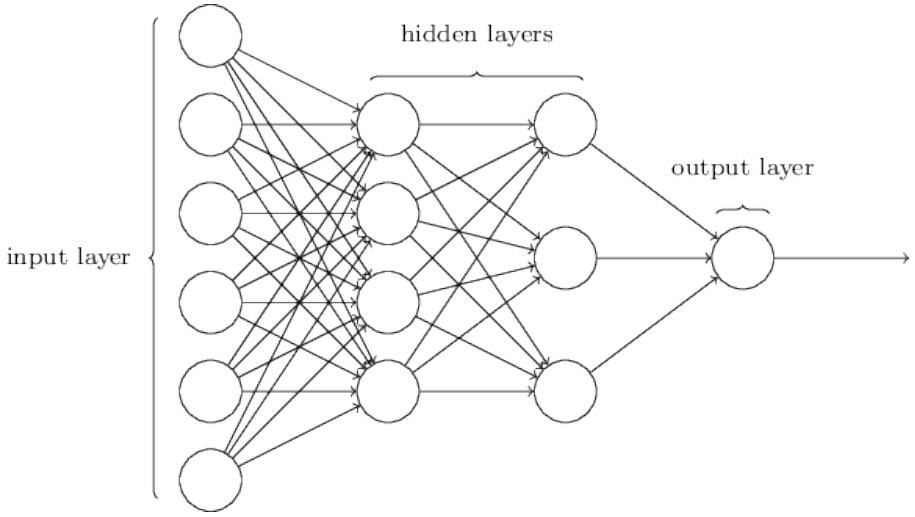


Figure 2.1: A Multi-layer Perceptron (MLP) with two hidden layers and a single perceptron as output layer. Every layer in this scheme is fully connected.

intend a set of techniques that automatically model the representation needed for a given task, classification in our case, from raw data. The inputs of the network travel through the first layer, whose output is the input of the second, and so on until the last one, called the output layer. Every layer performs operations or transformations on the vectors it receives as input depending on its structure and nature. Depending on the needs of the current task, we can process the signal in several manners, exploiting different kinds of layers in the network. In principle, the more the number of layers increases, the more the network learns more abstract features from data.

The most basic network configuration is called a Multi-layer Perceptron (MLP). A Multi-layer Perceptron (MLP) consists of:

- an input layer, which receives data as input vectors;
- an arbitrary number of hidden layers which model the needed data representation as a function between input and output;
- an output layer, which gives the final prediction outcomes. Figure 2.1 shows the scheme of a MLP.

Formally, the output signal y_j of a perceptron with n inputs is defined as

$$y_j = g_j \left(\sum_{i=1}^{n} w_{ij} \cdot x_i \right), \tag{2.1}$$

as it is the weighted sum of all the input signals to the perceptron. g_j is the activation function of the j neuron, which typically is a non-linearity. x_i are the inputs to the neuron and w_{ij} is the weight used by the j-th neuron to scale the i-th input.

All the artificial neurons have updatable weights w_{ij} constituting the parameters of the NNs. The net weights are updated in what is called a learning or training phase. Here, by learning, we intend the ability of a network to progressively improve its efficiency in a given task, such as image recognition or data classification. During the training phase, the input flows through the interconnected layers, and the net weights are updated. The update is the key factor of a NN system. Starting from a random set of weights values, we compute the update by minimizing the error between the desired output of the network and the predicted one. We do so through a specific algorithm called back-propagation [27]. This computes the gradient of a defined cost (or loss) function with respect to the weights and propagates the error from the outputs to the inputs.

NNs have become a standard technique in many fields due to their versatility and efficiency, such as vehicle control, medical diagnosis, cybersecurity, physics, finance, geomorphology, and pattern recognition. In the latter discipline, we can highlight signal classification where the scope of this work fall. We now present some deeper description of the NNs we adopted.

2.1.1 Recurrent neural networks

RNNs [28] are a class of artificial neural networks specialized in dealing with sequential data [29]. Sequential data is everything that we can represent as a sequence. A typical input of such networks can therefore be denoted as $(x_1, x_2, ..., x_T)$ where each data point x_t is a real-valued vector. As sequential data we can refer to both time-dependent and not time-based sequences, such as words in a written text. Thus, RNNs applications are various: natural language processing [30], speech recognition [31], text classification [32] and generation [33]. Figure 2.2 gives an intuitive sense of the recursive nature of the net, where its compressed representation on the left is time-unfolded on the right. In this figure, the network A outputs h_t given the X_t inputs and passes information from

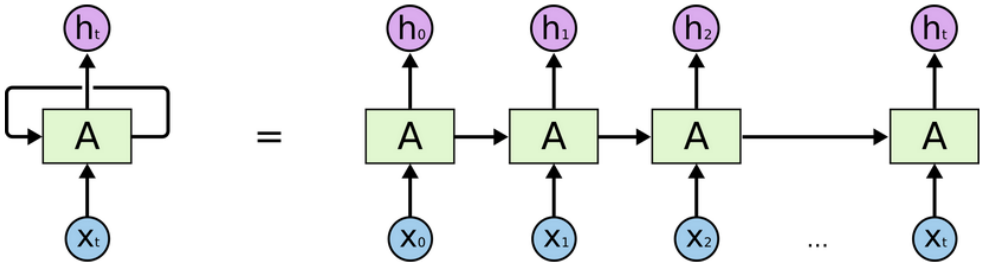


Figure 2.2: Intuitive recursive scheme of RNNs.

one time step to the next. This structure allows the network to model a function based both on present and past. Using an internal memory, RNNs represent information from an arbitrarily long context window. Figure 2.2 also shows the peculiarity of RNNs where the output of a unit is forwarded to the next one, which also loops its output back on itself. Despite this recursive nature, the temporal component is lost after few iterations. We thus say that RNNs have a short memory, and they cannot be used efficiently with long data sequences. The reasons for the long-term dependencies problem were presented in [34]. Error signals flowing backward in time tend to either blow up or vanish [35]. Researchers proposed many different network structures to solve this problem. Above them, we highlight the Long Short-Term Memory Recurrent Neural Network (LSTM) as it is the structure used for the speech classification task in the present work.

2.1.2 Long short-term memory networks

The introduction of Long Short-Term Memory Recurrent Neural Network (LSTM) aimed to solve the vanishing gradient problem, which emerges in the RNNs training phase. Whenever the gradient of the error function of the neural network is back-propagated through a unit of a neural network, it gets scaled by a factor that can be less or greater than one. RNNs in particular, suffer from this behavior when dealing with long-term time series, as many training steps become needed. To avoid the excessive dominance of the gradient or its negligence LSTM units were designed to implement a scaling factor of one [36].

The central ideas behind the LSTM architecture are: to use a memory cell that can maintain its state over time; to use nonlinear gating

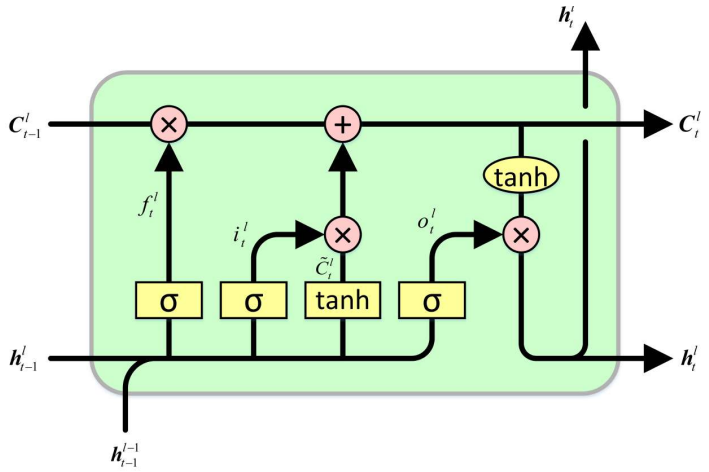


Figure 2.3: Representation of a LSTM cell [1].

units, which regulate the information flow into and out of the cell [31]. Figure 2.3 show the typical structure of a LSTM cell with the following notation: h_{t-1}^l is the input, h_{t-1}^{l-1} is the previous cell output, f_t^l is the forget gate, i_t^l is the input gate, o_t^l is the output gate, C_t^l is the state of the cell, $\tilde{C}_t^l$ is the update state of the cell, C_{t-1}^l is the previous cell memory, h_t^l is the output. By maintaining a unit state unchanged, the cell output becomes independent from its state, allowing the gradient of the error function to flow back to previous units unchanged. The name of this kind of NN follows from the net property of maintaining for longer sequences the short memory of a normal RNN. This long short-term memory makes the LSTM architecture the preferred choice when dealing with long-term time-dependent series [37]. The architecture details can be rather complex. This is also due to the many variants designed in literature [38]. We will now provide some formal definitions of the key LSTM concepts in order to define the critical concept of memory, providing a better understanding of the work we are presenting. In the following equations we have lighten the previous notation. We will describe variables and parameters for each definition separately

The hidden state of a RNN is computed as:

$$s_t = \tanh(Ux_t + Ws_{t-1}) \tag{2.2}$$

where x_t is the input to the unit, t the current time step, s_{t-1} the previous hidden state. U and W being parameters. A LSTM does the same but in a different way as it uses gates to compute its peculiar cell memory.

From now on x_t will represent the unit input, t the current time step, s_{t-1} the previous hidden state and U, W will represent parameters of the cell.

The input gate defines how much of the newly computed state, at the current time step t, we want to let through by computing

$$i = \sigma(U^i x_t + W^i s_{t-1}) \tag{2.3}$$

with σ being the sigmoid function implemented in the LSTM cell as shown in Figure 2.3. The forget gate defines how much of the previous state we need to let through by computing

$$f = \sigma(U^f x_t + W^f s_{t-1}). \tag{2.4}$$

The output gate defines how much of the internal state we need the external network to get by computing

$$o = \sigma(U^o x_t + W^o s_{t-1}) \tag{2.5}$$

We can finally give a formal definition of the cell new state C_t:

$$C_t = C_{t-1} \circ f + g \circ i. \tag{2.6}$$

C_t is a combination of the previously computed memory C_{t-1} multiplied element-wise ($\circ$) by the forget gate f and the newly computed hidden state g, multiplied by the input gate i giving the idea of a cell memory. The so called "candidate" hidden state g is defined as

$$g = \tanh(U^g x_t + W^g s_{t-1}) \tag{2.7}$$

and depends on the current input and previous hidden state. It has the same equation of the vanilla RNN hidden state. However in LSTM, we use the previously defined input gate to get just a part of this g state, controlling the cell state.

2.1.3 Convolutional neural networks

CNNs are a common NN architecture that was inspired by the visual cortex of animals [39]. They excel in machine learning problems, especially those requiring image data, computer vision, and natural language

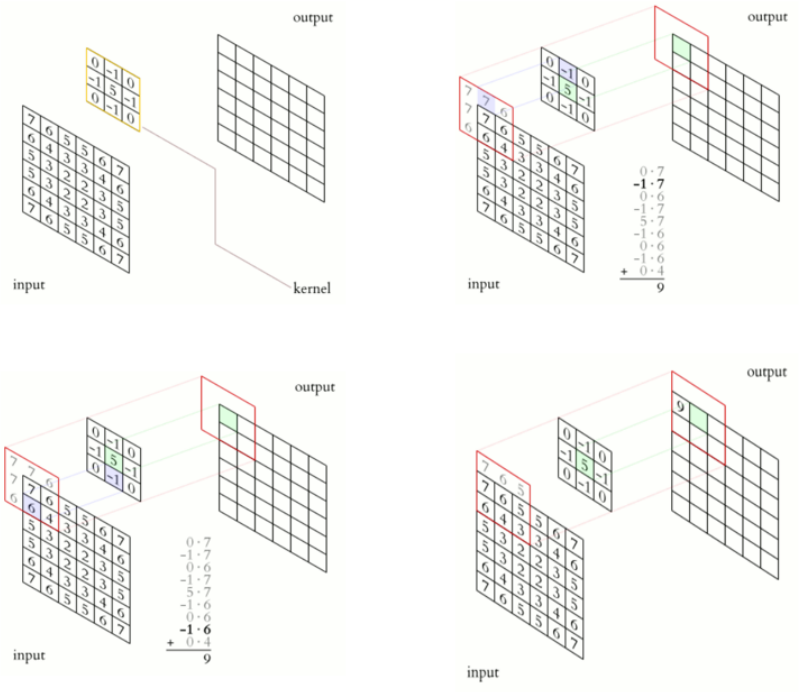


Figure 2.4: Visual sequence of operations in 2D convolution performed with a 3x3 kernel [2].

processing. They normally have multiple layers, including convolutional layers, non-linearity layers like pooling layers and fully-connected layers [40]. As pooling and non-linearity layers do not have parameters, one of the main advantages of the architecture is that the total amount of parameters to train is reduced, decreasing computational complexity in the training phase. We will now present those layers in more detail.

2.1.3.1 Convolutional layer

The convolutional layer is where most of the computation is done. It takes this name from the linear mathematical operation between functions, matrices in the NN case, called convolution. The parameters of the layer are a set of learnable filters or kernels [41]. In image processing, a kernel or convolution matrix, is a small matrix with fixed values used to apply filters to images. The idea behind the process is to let the kernel slide on the image computing the new values following bi-dimensional convolution rules. An intuitive idea of the process is given in Figure 2.4. The input represent the matrix containing pixel color values of an image.

The kernel slides on the input so that its center correspond to the pixel of the input matrix to be processed. Each output value is computed as the weighted sum of the product of each element of the kernel matrix with the corresponding pixel of the underlying input matrix

It is possible to extend convolution to 3 dimensions. In this case, the kernel is a 3D matrix that slides on 3-dimensional data. A video is an example of 3D data. We use two dimensions to define its frames and another one for their temporal evolution. This means that the first two dimensions are spatial, while the latter is temporal. Neural networks that implement this kind of computation are called 3-Dimensional Convolutional Neural Network (3DCNN). This work bases its video pipeline on this kind of network. When dealing with this kind of layer, the hyperparameters, the variables which determine the structure of the architecture that we can control are:

- depth, i.e., the number of filters in a layer
- filter size
- stride, i.e., the amount of filter movement
- zero-padding, to adjust the image size to the net requirements.

Here is also where the parameters reduction takes place. It is a well-established method to connect to the following layer regions of the picture and not the entire image pixel-wise [40, 42, 43, 44, 45].

2.1.3.2 Pooling layer

Pooling is a form of down sampling for images. It consists of dividing the original image in pixel regions, or sectors, and operate on each of those through a mathematical operation.

We can name different kinds of pooling depending on the operation we perform in each region. Max-pooling is the most common form. For each sector, the maximum pixel value is chosen and returned. Figure 2.5 shows an example of this. Another pooling operation is average pooling, which consists of outputting the average values from a given sector instead of the maximum. This method allows to smooth out the loss of information between layers. In the present work, we adopt this technique instead of the max-pooling one.

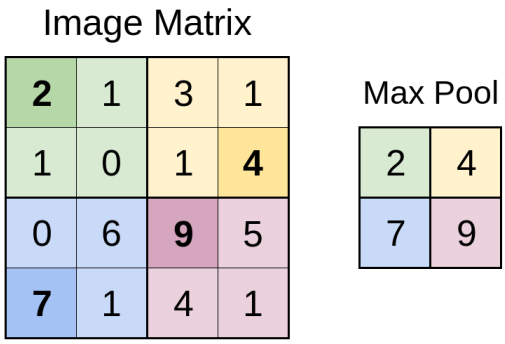


Figure 2.5: An example of a 2x2 max-pooling operation [3].

2.1.3.3 Fully connected layer

In this architecture, all the neurons of a layer are linked with all the neurons of the previous and the following (if present) layers. Here is where the high-level reasoning is done [41], where abstract data features are learned. The structure of the connections resembles the MLP one, as previously shown in Figure 2.1. For many inputs, its formal formulation is the same of (2.1). Typically this is also implemented as the output layer of networks designed for classification tasks. For example, our CNN model used for the emotion recognition and classification task has a 5-unit fully connected layer as the output layer.

2.2 Deepfakes

With the technical progress of recent years, fake multimedia can now provide a very advanced level of realism [46]. The boundary between real and synthetic content is rapidly narrowing as media manipulation techniques and tools are now powerful and easy to use.

When talking about media manipulation tools, we do not simply refer to well-known image and video editing software suites. We also refer to deep learning tools like autoencoders [47] or generative adversarial networks Generative Adversarial Networks (GANs) [48]. Autoencoders are neural networks designed and trained to encode data by learning abstract representations of the inputs efficiently. GANs are a recently proposed method to allow an artificial neural network to generate new data having the same statistical distribution of a given dataset. When considering a large amount of data as input, these tools can output very realistic

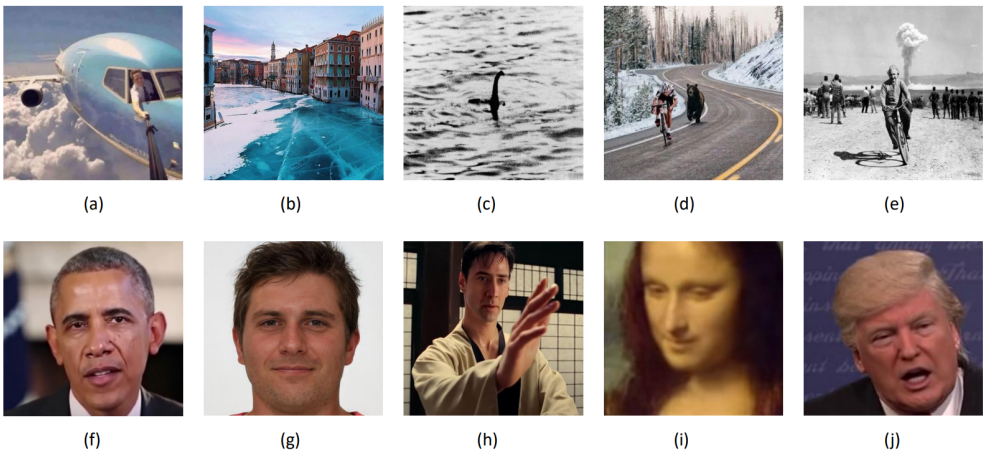


Figure 2.6: Examples of fake contents. [4, 5, 6, 7, 8, 9, 10, 11, 12, 13]

synthetic outcomes. A staggering example can be found in *thispersondoesnotexist.com* [10]. It is a recent site created by the software engineer Phillip Wang which highlights the AI ever-increasing power to present as real completely artificial images. Each time the site's page is refreshed, it returns a new GAN generated human face. The project results are impressive and completely believable. Figure 2.6(g) gives an example.

Deepfakes, a crasis between deep learning and fakes, are manipulated or generated visual and audio content with a high potential to deceive [49]. On the one hand, this opens new possibilities in creative arts, advertising, film production, and video games [46]. On the other hand, it poses threats to democracy by manipulating public opinion during elections [46], discredits politicians [50], creating believable fake news; and to the individual, allowing new forms of fraud or blackmail [49]. The need for automatic tools for deepfake detection is therefore compelling. In a sense, deepfakes are for videos what image editing was for photography. Figure 2.6 portrays some examples. We already have a sensibility to catch fake pictures, that can be manipulated or falsified in many ways. Images like 2.6(a), (b) are created by juxtaposing a photo over the background of another one. Images like 2.6(d), (e) are obtained by inserting in a picture just one external element. Others, like the famous example in 2.6(c) are completely crafted to portray fake or deceptive subjects. Nevertheless, as we tend to place even more trust in the voices we know and the videos we watch [51], the impact of malicious fake videos could be much more profound. Video frames like 2.6(f), (j) and images like

2.6(g) result deeply believable, even if completely synthetic. Other examples reported in the figure come from benign applications of deepfake techniques. In 2.6(h) we find an actor playing in a film he never did and in 2.6(i) we find an animated version of the Mona Lisa. Our trust and perception about video and audio recordings could change, and probably will, as it already was for photography.

Of course, we can not overlook benign opportunities. One leading example is the movie industry, which relies on computer-generated images Computer Generated Images (CGI) for special effects. CGI requires expertise, extensive training, expensive hardware, and special software [49]. With deep learning generated videos, movie production can become more efficient and accessible, and foreign movies may be enjoyed without the need for subtitles [52]. On the other hand, we must detect the illegal use of deepfake videos, and detection policies must be implemented to identify the truth better.

2.3 Emotion Recognition

Emotion Recognition (ER) is the ability to identify human emotions [53]. Precisely, when machines perform identification, we call it automatic emotion recognition. From now on, we are referring to this particular field.

The growing interest in this multidisciplinary area of research is just partly justified by the growing ubiquity of electronic devices. Making human-computer interactions more harmonious is only one of the scopes of this research field but is probably the most natural as we naturally communicate and interact through emotions in the real world and devices in the virtual one. It is interesting to notice that computer games [54], robotics [55] and psychology research [56] benefits from emotion recognition findings and evolution.

The ER task is a challenging one. To implement a solid ER system, we need a solid model to define emotions. However, the consensus on the definition of emotion is far to be found [57]. The beautiful work from Plutchik [57] listed more than ninety definitions of emotion proposed in the last century. So they do not have a clear definition and humans tend to be in mixed emotional states. Even when some of them are

predominant, we are known to misinterpret them at times.

2.3.1 Emotions and their models

Emotions are convoluted states of an individual psyche. They are composed of several elements such as personal experience, physiological state, or present contingencies and can rapidly vary in time [58]. We can use the two models for emotion classification, one is continuous, and one is discrete. Dimensional, or continuous, emotional model uses several quantitative features of emotions such as valence, arousal, control, power to describe them. Those characteristics are treated as dimensions, making emotions dependent on each other in this newly defined space. The most used aspects in the field are three: valence, arousal, and power, also called dominance [58]. Valence describes whether an emotion is positive or negative. Arousal measures the strength of the felt emotion. Dominance refers to the seeming strength of the person that is feeling that particular emotion. Between the disadvantages of the model, the main one could be that it is not intuitive enough and special training may be needed to label each utterance [59]. Other issues may be that in this description of emotions, some become identical, easy examples are fear and anger. Others like surprise cannot be categorized without context as it can be positive or negative depending on the circumstances.

The discrete emotional model is based on a more intuitive understanding of emotions. It treats them as definite and independent categories. As shown by Ekman et al. in [60] and in [61] some emotions like happiness, sadness, anger, fear, disgust and surprise can be identified in many different cultures. Following this research, it is possible to obtain the rest of the emotional spectrum as a combination of these inborn and culturally independent.

In the present work, we use ER techniques to discriminate between real and fake videos. We do so via emotion consistency detected in audio and images coming from considered clips. If portrayed sentiments are detected as consistent, we suppose the audiovisual content to be real. It is supposed fake otherwise. We can divide emotion recognition from video into two research areas. Speech Emotion Recognition (SER) and image emotion recognition. We define Speech Emotion Recognition (SER)

as the collection of techniques and processes that classify speech signals via their emotional content [58]. Hearing is one of our most advanced senses, and it is especially fine-tuned for human speech [62]. Speech is a distinctive trait of humans. It is so important to us to convey emotions through language that we developed tools like emojis to express ourselves better in written text. Moreover, we can somewhat easily infer emotions just by hearing the voice of a speaker, even if it is not from our mother tongue [63]. We are already prone to it as a species and we are exceptionally trained in it by everyday life. It is not a surprising fact that we are trying to extend this understanding to computers [58]. Similar analogies can be done with images and faces.

From an image, we can estimate a lot about the emotional state of the subjects in it. We are very experienced in detecting important cues from visual stimuli as a species. We can intuitively identify the context, the actions, gestures and facial expressions. This is not as easy for a computer. Video emotion recognition is the branch of computer vision that tries to make machines deal with visual stimuli correctly or at least as a human would. Face, head and hands movements give precious informative content. The discipline aims to extract and process those features that come from video frames. It is interesting to notice De Silva et al. results [64]. Under the study conditions, subjects recognized happiness, disgust, anger and surprise better from video information, while sadness and fear were better recognized from audio information. In the present work, we focus on facial expressions rather than body movements or hand gestures.

2.3.2 Emotion datasets

Data collection is the very first step of every classification problem. The classification relies on labeled data, that must be correctly collected and must satisfy the needs of the considered problem. Basing our claim on how emotions are collected we can divide emotion databases into three main categories: simulated, induced, natural. In a simulated environment, emotions are performed by semi-professional or professional actors in some cases even with a direction. Performances are collected and labeled according to the performance script. This is the easiest emotion

collection method but also the furthest from a real-world environment. Acted emotions are interpreted and not naturally occurring. The conveying of real-life feelings is not guaranteed and they may even be exaggerated. On the other hand, a controlled environment allows to obtain more features from subjects. While video and audio are present in almost every emotion dataset, body data like heart rate or temperature are impossible to obtain in different scenarios. Elicited or induced emotions are obtained by placing subjects in simulated emotional situations. Films, music, stories commonly elicit subjects' emotional responses. They can be seen as improvised acting performances, narrowing the difference with spontaneous emotions. For these reasons, in recent years, the study of emotion recognition on the expressed stances has gradually moved from posed or induced expressions to more spontaneous expressions [53]. Natural emotion databases are obtained from TV talk shows, radio talks, call-center recordings and similar sources [53]. Emotion labeling criteria are another fundamental aspect of emotion classification and dataset generation. We in fact can define emotions using different models, discrete and continuous as described in 2.3.1. Because of this, emotional datasets are generally provided with exhaustive explanations of their labeling process.

2.4 Conclusive Remarks

This chapter has provided the reader with the main background concepts needed to understand the rest of the work. These range from the set of the used machine learning tools, to the definition of deepfakes, to the problem of emotion recognition from images, audio or video. In the next chapter, we will overview state-of-the-art methods related to both emotion recognition and deepfake detection, the main topics considered in this work.

3

State of the Art

This chapter introduces the state of the art related to this work. We describe some remarkable studies related to SER, video emotion recognition, and deepfake detection, along with some of the fundamental technical aspects we employ in our system.

3.1 Speech Emotion Recognition

Speech is thought to be evolved from early hominids' communication system when they acquired intentionality and cooperation [65, 66]. As humans, we find it to be the most natural way to express ourselves. What separates human language from other animals communicating systems is that human language is open-ended. That means that we can produce a vast range of utterances, create new words and sentences starting from a finite set of elements.

From the audio point of view, speech is a continuous signal of varying length that carries information, and SER aims at detecting the emotional content it expresses automatically. The smallest unit of speech is called

an utterance, generally defined as a brief part of continuous speech ending with a clear pause. Utterances are also the basis of speech emotion analysis techniques. From the literature, we can provide a standard pipeline that is adopted by most of the SER systems [53, 67, 58]. This consists of a pre-processing step, followed by feature extraction, selection, and feature classification.

Pre-processing is the very first step after data collection, used as preparation for feature extraction. It typically starts with signal framing, which is the process of fragmenting the input signal into short partitions. Thanks to this operation it is possible to treat every segment as a quasi-stationary signal and compute audio features from it. After the fragmentation step, windowing functions can be applied to each segment. Window functions are useful to smooth the information loss caused by the Fast Fourier Transform (FFT) of data at the edge of the signals. In fact, framing a signal leads to abrupt edges in the time domain, which results in spectral leakage in the frequency domain. One of the most used window functions is the Hamming window. Formally, a M-sample Hamming window is defined as

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{M-1}\right), \qquad 0 \leq n \leq M. \tag{3.1}$$

SER systems often rely also on speech detection. We can detect fragments of voiced speech by exploiting its periodic nature using techniques such as zero-crossing rate [68] and auto-correlation [69]. Another operation that we can perform is noise reduction, which deletes or attenuates the background noise captured with the speech signal. As shown by Pohjalainen et al. in [70], Minimum Mean Square Error (MMSE) and log-spectral Amplitude MMSE (logMMSE) are the most successfully methods to perform this operations. Minimum Mean Square Error (MMSE) in particular, estimates the clean signal from a given sample function of the noisy one and minimizes the distortions between the two.

Once we have performed the preprocessing step, a set of features is extracted starting from the signal. Audio features, also called Low Level Descriptor (LLD), are one of the crucial aspects of speech emotion recognition. We can divide them into two categories: global and local. Global features are long-term characteristics of the signal like mean, standard deviation, minimum and maximum values. Local features are those that

we can extract by signal fragments assuming the quasi-stationary hypothesis. As emotional features are not uniformly distributed along with speech signals, we can use local features to capture temporal information. This is fundamental in SER systems because, as shown by Rao et al. [71], emotions like anger are predominant at the beginning of utterances, while emotions as surprise lie at the end of it. The LLDs we can extract starting of an audio signal are many. We now present some of the most significant ones, along with those used in the present work following Akçay and Oğuz nomenclature [58].

- *Prosodic features*

  Intonation, stress, and rhythm are features that humans can perceive that convey the most emotional content [59]. Some studies even show that SER systems get similar results or perform better compared to human judges when prosodic features are used [72, 73]. The fundamental frequency of the vocal cords f_0 and its changes over time describe the tone and the rhythm of the voice speaking. Statistics such as the mean, maximum and minimum values and the range of the f_0 are the most salient aspects of f_0 contour. f_0 contour decreases when anger or sadness is expressed and increases while the sentiment expressed is joy. Sadness is also associated with lower f_0 values [74]. The volume or intensity of the voice signal V, correlated with signal energy E with $E \propto V^2$, is associated with different emotions. Research shows that anger, happiness, or surprise yield increased energy while disgust and sadness result in decreasing energy [75].

- *Spectral features*

  Using the Fourier Transform, it is possible to obtain spectral characteristics of the voice signal. There are many of them that we can use. Above all, spectral centroid, spectral roll-off, mean, variance, and other statistical momenta can be extracted and used as features, as we did in the present work. The most widely used spectral-based features are some coming from cepstral analysis. Cepstrum is obtained by applying the Inverse Fourier Transform to the logarithm of the spectrum. We can use it to find periodicities in the frequency domain.

Furthermore, Mel Frequency Cepstral Coefficients (MFCC), which collectively make up a mel-frequency cepstrum, allow good classification results and can be used to surpass or enhance the classification performances based on utterance or prosodic-based features [76, 77, 78]. The mel spectrogram is a representation of the frequency content of a signal as it varies with time, in which the frequencies are converted to the mel scale. The mel scale is a perceptual scale of pitches perceived by listeners to be equal in distance from one another, derived by psychoacoustic studies. Other successful cepstral features are the Linear Prediction Cepstral Coefficients (LPCC), obtained from Linear Prediction Coefficients (LPC), which is a smoothed envelope of the spectrum. They are decisive in speech classification as they yield the distinctive characteristics of the speaker's vocal tract.

- *Voice quality features*

  Voice quality is dependent on the physical conformation of the vocal tract, which can be modified during speech. Involuntary and voluntary changes may produce differentiation in the emotional content of the speech. In [79] Cowie et al. showed a strong correlation between voice quality and emotional content of the speech.

  We define as jitter the measure of the fundamental frequency instability. It measures the changes of f_0 between successive vibratory cycles of the vocal cords. Shimmer is the counterpart of jitter for amplitude as it measures the amplitude variations. Harmonics to Noise Ratio (HNR) is the ratio between periodic and aperiodic components in voice speech signals. It gives the relative level of noise in the frequency spectrum of vowels. These last three features have been shown to improve classification performances when mixed with prosodic features [80, 81].

There are other feature typologies that we do not mention as they were not used in this work. The biggest category we do not describe is Teager Energy Operator [82] based feature, specifically designed to recognize stress and anger.

After features are extracted, it is possible to detect emotion. This can be done in two different ways: following a categorical approach; following

a continuous approach. In the first approach, emotions are considered categories or classes within a list (e.g., happiness, sadness, etc.). Emotion recognition consists in attributing one of these categories to the speech under analysis. In the second approach, emotions are modeled in a continuous space. In this context, it is customary to consider the valence-arousal space, where each emotion is mapped into a point. Valence dimension describing whether an emotion is positive or negative. Arousal dimension defining the strength of the felt emotion.

SER has been around for more than two decades. In the early stages of the field, classifications work was based mainly on classical machine learning techniques. Recently, with the advance of computational power, the community switched its interest to deep learning-based techniques due to their better performances. One of the first significant neural network-based works was [83], performances of recognition were about 50% on eight emotions: joy, teasing, fear, sadness, disgust, anger, surprise, neutral. The data they used was collected for the work from a total of 100 speakers, 50 male and 50 female native Japanese speakers, where each subject uttered a list of 100 Japanese words eight times, one time for each of the eight emotions. On Berlin EmoDB [84] Harár et al. in [85] achieved excellent results describing an architecture made with convolutional, pooling, and fully connected layers with a voice activity detection algorithm to eliminate silent fragments. They achieve excellent classification results on anger, neutrality and sadness. Zhao et al. in [86] applied two CNNs and a LSTM to IEMOCAP [87] speech signals. Raw signals are given to the CNNs that learn local features. Those local features are then fed to the LSTM which learns their long-term dependencies. The data used in this work come from the audio part of Berlin EmoDB and IEMOCAP databases. Using different sets of emotions for different experiments outperforms all traditional approaches proposed for emotion recognition on the databases.

As in [86] many deep learning-based approaches to SER rely on CNNs and at least a LSTM to deal with temporal dependencies and spectral variations [58]. However, this structure increases the complexity of the system. Using 3DCNNs Kim et al. in [88] proposed a modeling of spectral-temporal dynamics based only on the CNN. They focused on seven significative databases: LDC Emotional Prosody [89], eNTER-

FACE [90], Emo-DB, FAU-aibo emotion corpus [91], IEMOCAP, SEMAINE [92], and RECOLA [93]. The emotions considered in this case are four: neutrality, happiness, sadness, and anger. It is interesting to notice that SEMAINE and RECOLA provide only continuous labels such as arousal and valence for emotions. So they mapped continuous labels into the four classes mentioned above using landmarks of valence and arousal dimensions provided by FeelTrace [94]. The latter is an instrument developed to track the emotional content of a stimulus used in the article to switch from the continuous emotional model to the discrete. They claim to have therefore designed the largest corpora for SER experiments with the largest number of speakers and samples. What they obtained is that with the high variance of their data resembling realistically "in-the-wild" scenarios, the 3DCNN approach is simpler and outperforms CNN-LSTM methods.

3.2 Video Emotion Recognition

Human emotions detection has not only been studied at speech level. Indeed, many applications build upon the possibility of detecting emotions from video analysis. The possibility of detecting emotions from video has been made clear in 1974 Ekman's extensive studies on facial expressions [95], which provided two essential results: universal and cross-cultural facial expressions convey some emotions that provide sufficient clues to detect them automatically. This was done by following a typical video processing chain that is still used in most of the researches, and that is shown in Figure 3.1.

Nowadays, most of the video emotion recognition systems exploit the use of convolutional networks. As CNNs are one of the highest performing NNs for computer vision tasks, they are one of the preferred approaches for video and image analysis. Like in the SER field, they are often combined with LSTMs to capture temporal cues. In [96] Khorrami et al. show this procedure to be superior to other approaches on AudioVisual+Emotion Challenge (AV+EC2015) dataset [97]. However, this is not always the case as Hu et al. in [98] show how their two stages spatio-temporal attention CNN model gives better results on RECOLA and AFEW-VA [99] datasets than previously proposed methods. Using

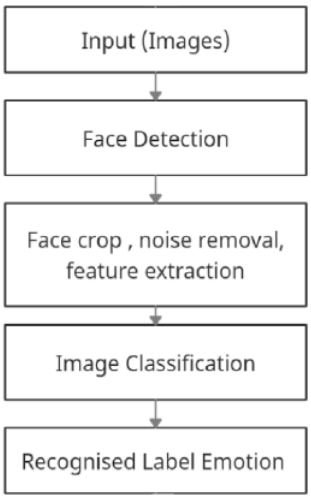


Figure 3.1: Typical pipeline used to perform video emotion recognition [14].

the attention mechanism, they focus their CNN training on informative spatial-temporal features for continuous emotions. The first stage produces a first recognition result that is fed into the second for correction. Both stages implement attention layers to catch the most relevant spatial-temporal features. They also propose a loss function for the two stages combined to improve the overall prediction results.

For the 2015 Emotion Recognition in the Wild (EmotiW) Challenge dataset [100] Kahou et al. propose in [101] a CNN-LSTM architecture for facial expression analysis. Their experiments with three different CNN architectures show that the spatio-temporal evolution of facial features is one of the strongest cues for emotion recognition. The convolutional networks are trained to classify the static images and the LSTM is fed with the features coming from internal layers of the CNN to model the temporal evolution of frames and predict a single emotion for the entire video.

Recently, Li et al. in [102] proposed a neural network with a video frame weight vector approach. They highlight that video sentiment analysis methods only obtain features from the spatial and temporal components of videos. This means they lack of a deeper understanding of emotions from typical video emotion recognition systems. Emotions are not expressed constantly, and it is not easy to understand which of the showed emotions contributes the most to the overall sentiment analysis

of the video. To solve this problem, they extract features from video frames, weighting them accordingly to their CNN emotion detection on the frame itself. These weighted features are then passed to a LSTM to obtain a video sentiment analysis model. On BAUM-1 [103] this model performs better than existing methods. Kahou et al. in [104] conclude that assigning one label for video length introduces noise to the training set. Therefore they tried to correct their CNN classification predictions by training them with still images. Their results are promising and we assume that this could be a good direction to follow.

3.3 Deepfake Detection

Deepfakes are manipulated or generated visual and audio content. Their creation can follow benign or malevolent intentions. Because of their high potential to deceive automatic detection techniques to recognize illegal uses of these synthetic media,must be put in place. Deepfake detection is a binary classification problem. It consists of determining if an audio-visual representation of a subject is genuine or synthetically modified or generated. Both classical machine learning classifiers and deep learning-based ones discriminate between authentic videos and synthetically generated ones. These methods require a large amount of real and fake videos to train models. As this kind of synthetic media is still relatively recent, the number of fake videos available is still rapidly growing, and new approaches to the issue are still being proposed.

According to [15], we can group deepfake detection techniques into methods that perform image analysis and video analysis (see Figure 3.2)

With tools such as CNNs and GANs, image deepfake detection task is challenging as it is possible to preserve the pose, the facial expression, and the lightning of the original photo when creating a deepfake [105].

In [106] Zhang et al. extract a set of features with the bag-of-words method and fed them to different classifiers: SVM, random forest and MLP to discriminate between altered faces and genuine ones obtaining good results. However, image synthesis with GAN models is still challenging to detect. GANs are in fact, very efficient in learning and reproducing the input data distribution. Their development is still ongoing and many new extensions are frequently introduced [15, 23] hardening

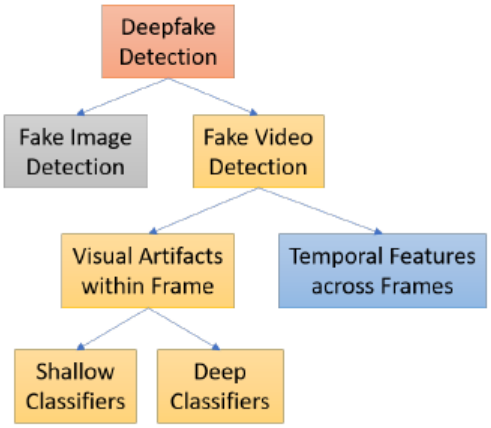


Figure 3.2: Overview of the main categories of deepfake detection techniques [15].

the detection task.

In [23] Xuan et al. removed low-level high-frequency clues of GAN generated images to increase the pixel-level statistical similarity between authentic images and fake ones. This is based on requiring the CNN classifier to learn more meaningful and generalized features than image distortions in order to create a more robust against GANs forensic model. Their experimental results prove the effectiveness of the proposed method.

Hsu et al. in [24] proposed an above state of the art performing system for GAN generated images detection. Firstly they generate a large corpus of fake images using five different GAN networks. These images are then used paired with real ones to make a CNN based architecture learn discriminative features, which are then extracted from internal layers of the network through the training process. These features are then fed to another small CNN to distinguish deceptive images from genuine. Their method significantly outperforms other state of the art fake image detectors. As shown in Figure 3.2 other deepfake detection techniques can be grouped in the fake video category. The latter is the category in which this work falls for the deepfake recognition part. More precisely, we apply techniques that look for temporal features across video frames.

We can also notice from the literature that early works based their discrimination methods on identifying unnatural physical behavior patterns while recently deepfake detection is trending towards deep learning-based

techniques and automatic feature extraction [15].

Yang et al. [107] exposed some 3-dimensional inconsistencies of fake videos. Considered deepfakes are created by merging the synthesized central face region into the original image. This introduces errors that can be revealed when estimating 3-dimensional head poses from the face images. They base their technique on supposing the synthesis algorithm not to guarantee that the original face and the synthesized face have consistent facial landmarks. Those are locations on human faces corresponding to essential structures such as eyes, nose, and mouth tips. The work compares head poses estimated using all facial landmarks and those estimated using only the central region of the face. For real faces, the two estimated head positions will be close. For deepfakes, since the central region is from the synthesized face, the mismatch of landmark locations between original and generated images will lead to more significant differences between the two estimated head poses. The proposed classifier is a Support-Vector Machine (SVM), a supervised classifier. Those calculated differences in estimated head poses are used as a feature vector to feed the model providing successful results.

In 2018, Li et al. [108] revealed unnatural eye blinking patterns in synthetically generated videos and exploited these findings to propose a discriminative method. Landmarks were used in this work as well to identify and align faces during data preprocessing. Faces alignment allows extracting rectangular portions from the images containing the eyes without distortions. These framed image sequences are fed to a CNN based on VGG pre-trained model to extract image features. The proposed system also learns the temporal features of the sequences with an LSTM layer. A fully connected MLP closes the classification pipeline providing the final binary classification with good results.

In a 2019 work funded by Google, Microsoft, and the Defense Advanced Research Projects Agency, Agarwal and Farid [22] highlighted that this kind of fake media occasionally contains spatio-temporal glitches. However, these glitches are not a reliable method to detect fake media as they are continually being reduced, and it is reasonable to expect that with the methods and technique evolution, those glitches will be eliminated. They base their work on politicians' images as they were one of the first categories to suffer from malicious deepfake forgery due to

their exposition and accessible collectible video data. This simplified the training accuracy of deep learning tools and thus creating believable fake videos. Their work uses handcrafted facial-head features extracted with OpenFace [109, 110, 111] a well-known computer vision tool. They extract 190 features every 10 seconds of video for each subject, real or fake, and use them as vectors to feed an SVM classifier. The results are interesting for multiple reasons. They showed how correlations between facial expressions and head movements could distinguish a person from other people and deepfake videos of the person itself. They showed robustness to video compression compared to pixel-based detection methods. They found their low-level descriptors approach is vulnerable to contexts in which the person is speaking. Their dataset is prepared, with speakers filmed in formal environments with the subject looking directly into the camera. Live interviews are much more difficult to generalize on as subjects can look off-camera, they move more, and lightning changes fast.

Recently, Bonettini et al., in [25] studied the ensembling of different trained CNNs to detect facial manipulations. They train four CNN models to detect manipulated videos over Faceforensics++[112] and Deepfake Detection Challenge dataset (DFDC) datasets and compare results with a baseline network. The article show that the ensemble of networks which make use of attention layers and siamese training leads to promising detection results. They report superior average scores compared to the baseline on test sets of both dataset. The performances of the four ensembled nets led the work results to top 3% on the leaderboard computed against the public test set of the DFDC challenge.

We find in [26] by Hosler et al. a similar to our work approach to deepfake detection. They use emotion predictions to detect inconsistencies in emotion conveyance on both speech and faces of the DFDC [113] dataset. While we use the discrete model to define emotions, the work is bases its definitions on valence and arousal dimensions. Using LSTM networks they predict emotions from audio and video LLDs of the SEMAINE [92] dataset. Predicted emotion in time is used to classify videos as authentic or deepfakes. They show experimentally that the proposed system is able to discriminate between real and deepfake videos with accuracy values of up to 99.5%.

Another way to correctly spot deepfakes is watermarking. The cur-

rent ideas are to integrate watermarking and hashing tools into devices that people use to make digital content. This way, multimedia content could contain immutable metadata like time and location of creation. This integration is difficult to implement, but one of the most exciting ways could be blockchain technology. In [114] Hasan et al. provided some promising results for this direction exploiting the technology property of creating unchangeable blocks of metadata.

3.4 Conclusive Remarks

In this chapter we provided an overview of state-of-the-art methods related to the problems of emotion recognition from speech and video, and deepfake detection. In the next chapter we will formally define the problem tackled in this work, and we will provide all the technical details behind the proposed solution.

4
Method

In this chapter, we formulate the problem we tackle in this work. This consists of a deepfake detection system based on well-established concepts inherited from the automatic emotion recognition field adapted to the problem at hand. Here we describe the proposed architecture and provide all the details about it.

4.1 Problem Formulation

Given a target video signal y our goal is to estimate whether it is real or fake. Therefore, we want to assign y a class C with

$$C \in \{\text{Real}, \text{Fake}\}. \tag{4.1}$$

We consider as *Real* those videos that recorded real humans acting or speaking and have not been altered to change the identity of the depicted person in terms of visual appearance or voice. We consider as *Fake* the videos that have been generated or altered with some synthesis technique. As explained in Section 3.3, the main methods used for syn-

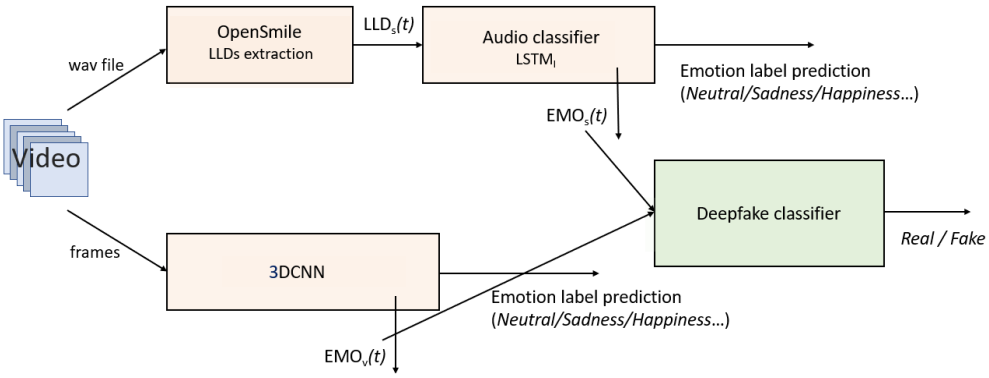


Figure 4.1: The architecture of the proposed system.

thesis are GANs and Deepfake Autoencoders (DFAEs). In this work, we consider fake videos generated with both techniques. In fact, our method estimates the class to which y belongs based on semantic characteristics of both its acoustic and visual components, which have to be intrinsically different between the two classes. Therefore, our approach aims to be sufficiently general to provide a correct class estimation for each type of deepfake synthesis algorithm.

4.2 Proposed System

Figure 4.1 shows the architecture of the proposed system. It is composed of two pipelines, one for each of the signal components, namely the audio track and the visual component. Both pipelines exploit findings from the automatic emotion recognition field. In particular, the system is based on the idea that emotional features can be extracted from both the audio and visual components of an input video. Then, we can use these features alone or combined together to detect whether a video is fake or not. The whole system is based on the idea that deepfake techniques can hardly synthesize high-level aspects such as human emotions in a natural way.

In the following, we provide more details about each block of the proposed method. Whenever some empirical choices have been made to select some proposed pipeline parameters, these are justified by a series of tests, as shall be clear from the forthcoming experimental chapter.

4.3 Audio Pipeline Description

Regarding the audio analysis, we consider short time windows of the input speech signal $s(t)$. From each window, we extract a set of acoustic LLDs (details about the specifically used ones shall be described in Section 4.7.1). Given a time window $s_{win}(t)$ of the speech signal $s(t)$, we call SMILE$(\cdot)$ the feature extraction process and we define the LLD speech features vector as

$$\text{LLD} = \text{SMILE}(s_{win}(t)). \tag{4.2}$$

Considering multiple time windows extracted from the speech time-series $s(t)$, we obtain the time-series $\text{LLD}_s(t)$ by stacking single LLD vectors, whose length depends on how many windows are selected. Formally,

$$\text{LLD}_s \in \mathbb{R}^{M \times K} \tag{4.3}$$

with M being the number of LLD vectors for each time segment and K being the number of extracted features. For speech emotion recognition, we implement an LSTM-based model. The LSTM architecture we found having the best performances on the emotion recognition task is made of two stacked long short-term memory layers with dropouts connected to two stacked dense layers. We call it LSTM_I. We feed the LSTM_I with labeled LLD_s matrices and we train it on five emotion classes. We discuss the labeling problem more in detail in Section 4.9.

From the trained network we extract vectors that we call speech emotional features, EMO_s, from the videos of which we want to infer the class C. The dimensions of each emotional vector are

$$\text{EMO}_s \in \mathbb{R}^{M \times N}, \tag{4.4}$$

where N is a parameter that depends on the shape of the neural network we want to use, as described in Section 4.6, and M is defined as previously in equation 4.3. More formally, we can define EMO_s vectors as

$$\text{EMO}_s(t) = \text{LSTM}_\text{I}(\text{LLD}_s(t)) \tag{4.5}$$

where $\text{LSTM}_\text{I}(\cdot)$ is the trained speech emotion recognition model. Each video is labeled according to the class from which it came from. Finally, they are passed to the deepfake classifier, that is the last stage of the system described in Section 4.5.

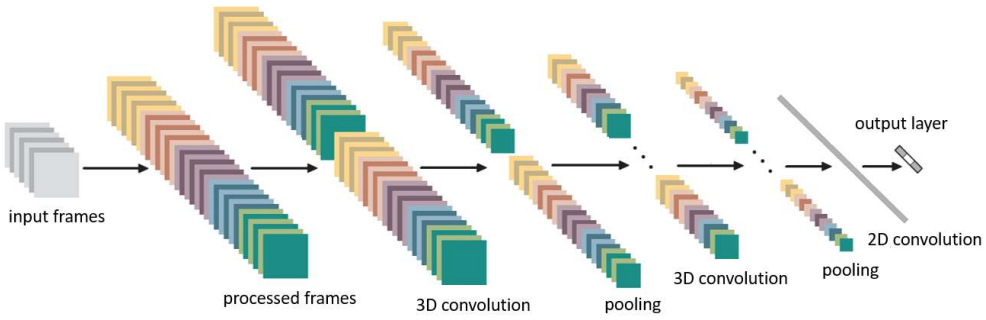


Figure 4.2: 3DCNN model implemented in [16].

4.4 Video Pipeline Description

The first steps of the video pipeline are frame acquisition and processing. For every input utterance (refer to Section 4.9 for a deeper explanation of the method) we extract the video frames and perform preprocessing to make them fit the dimensions requested by the network. Depending on the dataset we are working on, this procedure could be slightly different, as we will see in Section 4.6. Now we present the network architecture we implemented to let the reader understand the further steps of the pipeline. We implemented the same 3-Dimensional Convolutional Neural Network (3DCNN) structure presented by Ji et al. in [16] and we optimized it for our task parameters. This model extracts features from both the spatial and the temporal dimensions by performing 3D convolutions, thereby capturing the motion information encoded in adjacent frames. The structure of the original system can be seen in Figure 4.2.

As they propose, we pass from three color Red, Green, Blue (RGB) channels to one Black and White (BW) and we reduce the dimensions of each frame. Following the paper, for all the frames we compute the gradients along x and y dimensions, and x and y optical flows. Then, we create the map vectors we feed the 3DCNN with the frames concatenated to their gradients and opt-flows. We refer to these as MAP vectors. A formal description of their shape is

$$\text{MAP} \in \mathbb{R}^{H \times W \times N}, \tag{4.6}$$

where H and W are the height and width of each map respectively and N is the number of feature maps per vector. We feed our 3DCNN model with MAP vectors and we train it on five emotion classes.

From the trained video emotion recognition model, called 3DCNN($\cdot$), we extract vectors of what we call video emotional features, EMO_v, from videos of which we want to infer the class C. Formally

$$\text{EMO}_v(t) = \text{3DCNN}(\text{MAP}(t)). \tag{4.7}$$

where $\text{EMO}_v(t)$ is the images-derived time-series and

$$\text{EMO}_v \in \mathbb{R}^{M \times N}, \tag{4.8}$$

where M and N are defined as in equation 4.5. We label these video emotion features accordingly to the class of the video they were taken from and pass them to the last stage of the system.

4.5 Deepfake Detection Stage

Following the problem formulation we gave in Section 4.1 we need our last classifier to discriminate correctly between two classes. In this section, we present the two different classification models we experimented with and the two different modality fusion methods we used for deepfake detection.

4.5.1 Classification models

The first classification model we present is a LSTM neural network we refer to as LSTM_{II}. From both pipeline in fact we receive time-series describing emotion evolution in a considered video, being its speech part $\text{EMO}_s(t)$ or the visual part $\text{EMO}_v(t)$. For reasons we gave in Section 2.1.1, this kind of RNN is the preferred architecture when dealing with this kind of data. Therefore, we receive vectors in the correct format and there is no need for further processing. We feed them to the LSTM_{II} and train it to discriminate between the two labels in C: *Real* and *Fake*, on both modalities separately.

The second model we present consists of several classifiers. As we will later describe in Sections 4.6 and 4.7 and, we train multiple classifiers simultaneously on data received from pipelines. Then, for a later fusion stage, we consider the one that performed better in the classification task. As exhaustively described in Section 4.7 for this second approach, we are required to reduce the dimensionality of our data. For

each emotion time-series we receive, being it EMO_s or EMO_v, we compute 3 statistical moments: mean, variance, kurtosis and combine them in a one-dimensional vector we refer to as EMO_{stat}, with

$$\text{EMO}_{\text{stat}} \in \mathbb{R}^Q \tag{4.9}$$

where $Q = M \cdot 3$ and M is the same of equation 4.5.

4.5.2 Types of fusion

We can unify the two pipelines with two different criteria. One method is to combine $\text{EMO}_s(t)$ and $\text{EMO}_v(t)$ time-series corresponding to the same video. Starting from these data, we label the new vector accordingly to the time-series label and feed it to our final classifier, which predicts the class C. We call this feature-level fusion.

The second strategy is to predict the class for each $\text{EMO}_s(t)$ and $\text{EMO}_v(t)$ separately and then combine the prediction results. Predictions are probability values returned by the last stage classifier, of a video y being *Real* or *Fake*. The final score associated with each video is simply computed as the average between the scores given by models on the two modalities separately. We call this decision-level fusion.

4.6 Method Implementation

In this section we provide the implementation details and used parameters of the techniques described in Sections 4.3, 4.4 and 4.5. For the sake of clarity, we will exhaustively describe the tools and datasets used for the work in separate sections later in this chapter.

4.6.1 Audio pipeline setup

We now refer to the setup we used to implement the methodology described in Section 4.3.

The first step is to acquire audio descriptors from the considered speech signal $s(t)$. To do this, we use OpenSmile extraction toolkit (see Section 4.7.1) [115]. We extract 130 audio descriptors for each time window $s_{win}(t) = 10ms$ of the $s(t)$ signal, thus generating $(100, 130)$ vectors per second. For our experiments, we considered a time context of 3

Figure 4.3: A typical IEMOCAP frame.

seconds. Therefore, our LLD_s matrices have dimensions $(300, 130)$. To enable the feature-level fusion in the last stage of our system, we select the 300 LLD vectors from time regions individuated by timestamps returned by the video pipeline frame extraction. OpenSmile time resolution is higher than the frames per second rates of clips on which we performed emotion recognition and deepfake detection. This sub-sampling allows a consistent extraction of emotional features between modalities. The sampled LLD_s matrices are passed to the $LSTM_I$ network. As described in detail in the experiment chapter, we find that the best architecture comprises 6 layers. The first LSTM layer is made of 64 units and is followed by a dropout layer. The second has 32 units and is followed by a dropout layer as well. 32 is also the dimensional input space of the first fully connected layer. The final one is a 5 neurons dense layer, allowing the 5 class discrete emotion classification. Speech emotion features defined in equation 4.5 are extracted by the first fully connected layer, thus generating (300, 32) dimensional EMO_s vectors, which we pass to the last stage of the system.

4.6.2 Video pipeline setup

We are now referring to Section 4.4 implementation setup.

The two used datasets (see Section 4.8 for further details) contain videos with different frame dimensions. IEMOCAP [87], the dataset used to perform emotion recognition, has frames which dimensions are 480x720, width and height respectively, and display both actors speaking in black-contoured windows. An unprocessed IEMOCAP frame can be seen in Figure 4.3. The first step we need to implement is to acquire

Figure 4.4: A typical DFDC frame.

video information just from the most relevant frame section. By dataset design the subject on the left is considered as the main speaker and the one monitored by IEMOCAP's instrumentation setup [87]. Therefore, we crop the frame to consider just the visual information from the left window. Then, we extract the subject's face from this cropped frame using BlazeFace [116], a tool for face detection we describe in Section 4.7.2. From our experimental campaign we found the optimal frame reduction dimensions to be 80x60. On the other hand, DFDC clips, the dataset we used for deepfake detection [113], display one actor per frame facing the camera in a 512x512 window. A typical DFDC frame can be seen in Figure 4.4. We directly apply BlazeFace to these frames and perform the aforementioned dimensional reduction on bounding boxes that BlazeFace selects for each face.

We now proceed to describe the operations performed by the 3DCNN model we implemented. It is based on the one proposed by Ji et al. in [16]. For the sake of clarity, we advise to refer to Figure 4.2 as it helps to understand the architecture we will now describe. They propose 7 as the optimal number of the frame to consider per input vector. Our experimentation found 7 to be the best performance, allowing the number of adjacent frames to consider for our emotion recognition task. Therefore, every 7 frames, we compute their gradients along x and y dimensions obtaining 14 additional feature maps, and their x and y dimensions optical flows obtaining 12 more. Thus, independently of the dataset, MAP

vectors defined in equation 4.6 in Section 4.4 have dimension (80, 60, 33).

For each MAP vector, we apply 3D convolutions with a kernel size of 7x7x3. 7x7 in the spatial dimensions and 3 in the temporal dimension on each channel separately: frames, x-gradients, y-gradients, x-optflows, y-optflows. Two sets of different convolutions are applied at each location, obtaining two 23 feature maps vectors as in the considered article. Our spatial dimensions are now 74x54. In the following layer, we apply 2x2 average 3D pooling on each feature map, leading to the same number of feature maps with a reduced spatial resolution of 37x27. Next, we apply another 3D convolution layer with a kernel size of 7x6x3 on each of the five channels in the two sets of feature maps separately. We apply six different filters, leading to six distinct sets of feature maps, each containing 13 feature maps. The next layer is a 3x3 average pooling. We apply it on each feature map, from which we obtain the same number of feature maps with a reduced spatial resolution of 10x7. We perform convolution only in the spatial dimensions at this point. The convolution kernel size used is 10x7, so that the sizes of the output feature maps are reduced to 1x1. Then, we apply 32 filters to obtain the same output dimensional space of the feature extraction layer of the LSTM_I architecture described in Section 4.6.1.

Video emotional features are extracted from this convolutional layer, obtaining (300, 32) dimensional EMO_v vectors. The architecture is closed with a 5 neuron dense layer for the discrete emotional recognition task on IEMOCAP dataset.

4.6.3 Fusion stage setup

The first model we presented in Section 4.5.2 is LSTM_II. The architecture that provides the best classification results on our input data is composed of four layers. The first is a 32-units LSTM one, as we want the model to learn temporal cues from the time evolution of features $\text{EMO}_s(t)$ and $\text{EMO}_v(t)$ we are passing to it. After dropout, we applied two fully connected layers. The first is composed of 8 units and the second has 2, because of the binary nature of the deepfake detection task. The second approach is based on classical machine learning models, built and trained with Lazypredict [117]. A powerful tool was chosen for its ease of use

and versatility. See Section 4.7.3 for further details. As mentioned in Section 4.5.1, we need to reduce the dimensionality of our data to work with this tool. We perform this by computing mean, variance and kurtosis for each of the 32 columns of emotional features matrices, obtaining 96-dimensional EMO_{stat} vectors. On our data, we found the best performing model being LGBM classifiers (details about the model shall be described in Section 4.7.3). When we fuse modalities feature-level, we combine EMO_s and EMO_v matrices horizontally, passing from (300, 32) to (300, 64) vectors. This way, each row contains both acoustic and visual emotional information extracted from the same signal time-segment. We can now directly feed these (300, 64) vectors to our LSTM_{II} deepfake classifier, or process them as explained and feed Lazypredict's models. When fusing the two pipelines decision-level, we compute the average between the scores returned by models on the two modalities separately. This way, we can combine LSTM_{II} predictions with Lazypredict's and choose the best performing combination of the two, as we will describe in detail in the forthcoming experimental chapter.

4.7 Used Toolkits

In this section, we give a description of tools we used for the implementation of our method, and the motivations for their usage. First, we describe the toolkit we used in the audio preprocessing step of the pipeline. Later, we describe the tool we used in last stages of our video experimental campaign, we analyze this experiment in Section 5.4.4. Finally, we illustrate a tool we implemented for the last stage of our system.

4.7.1 OpenSmile

Frequency domain information, which is very important for human perception of sound and speech, is not readily available from raw, time-domain audio signals. Consequently, the signal processing community has crafted many low-level features useful for speech processing. We describe these in detail in Section 3.1. We remind we refer to these as Low Level Descriptors (LLDs). In this work, for audio analysis, we extract LLDs using OpenSmile [115]. It is an open-source, C++ implemented

software, developed by audEERING [118], for automatic extraction of features from audio signals. "SMILE" stands for "Speech and Music Interpretation by Large-space Extraction". OpenSmile gathers a vast pool of feature extraction algorithms from the speech processing and music information retrieval communities. Because of this, it is one of the most used toolkits in both sectors. The software can extract different features with a time resolution of 10 milliseconds, allowing real-time and commercial applications. We exploited this capability as we described in Section 4.6.1.

Table 4.1: List of features extracted by OpenSmile with the ComParE2016 configuration [17].

4 energy related LLD	**Group**
Sum of auditory spectrum (loudness)	prosodic
Sum of RASTA-filtered auditory spectrum	prosodic
RMS Energy, Zero-Crossing Rate	prosodic
55 spectral LLD	**Group**
RASTA-filt. aud. spect. bds. 1-26 (0-8 kHz)	spectral
MFCC 1-14	spectral
Spectral energy 250-650Hz, 1k-4kHz	spectral
Spectral Roll-Off Pt. 0.25, 0.5, 0.75, 0.9	spectral
Spectral Flux, Centroid, Entropy, Slope	spectral
Psychoacoustic Sharpness, Harmonicity	spectral
Spectral Variance, Skewness, Kurtosis	spectral
6 voicing related LLD	**Group**
F_0 (SHS & Viterbi smoothing)	prosodic
Prob. of voicing	voice qual.
log. HNR, Jitter (local & δ, Shimmer (local)	voice qual

In this work, we use the set of descriptors provided by the so-called *ComParE2016* configuration [119]. OpenSmile can operate in different configurations, developed for the many research competitions where the toolkit serves as a benchmark. This configuration in particular, was developed for the Interspeech 2016 Computational Paralinguistics Challenge, an open challenge dealing with states and traits of speakers as manifested in their speech signal's properties. We extract features from

three different LLD categories, described in Section 3.1, for a total of 130 descriptors: 65 descriptor and their 65 first temporal derivatives. Table 4.1 shows the complete feature list in detail.

4.7.2 BlazeFace

Since we focused on facial features for both our video emotion recognition architecture and deepfake detection task, we needed to cancel out as much environmental noise as possible from frame images. Namely, we had to process every frame to extract faces. A first attempt was made to implement a hand-crafted, static crop on the considered images. This technique was prone to misalignments and noise-capture as subjects in frames move concerning the camera shot, exposing the background of the environment in which the recording took place. Therefore, we needed to accurately and dynamically focus our feature extraction process on faces contained in frames. To do this, we used BlazeFace [116]. It is a face detector recently developed by Google. It was designed for efficient face detection via smartphone cameras. Because of this, it is fast and computationally light. The tool is based on a Single Shot Detector (SSD) architecture, specifically modified and optimized to exploit small GPUs capabilities. BlazeFace is at the present moment the best performing tool for real-world applications, namely between the frameworks that run on everyday devices [120]. It predicts the face bounding box from camera shots with an average accuracy of 98.61% and runs very fast: up to 1000+ frames per second on flagship devices [116]. We applied the tool to every IEMOCAP and DFDC clip, extracting and saving bounded face images detected in frames, before processing them as described in Section 4.6.2. Two bounding box predicted by BlazeFace can be seen in Figure 4.5.

4.7.3 Lazypredict

As previously mentioned in Section 4.5.1, one of the approaches we used for the last stage of our system comes from classical machine learning. We exploit Lazypredict [117] capabilities. This python package allows building many classical machine learning models at once. From its results, it is possible to understand which model works better for the input data it is fed with. The data vectors we receive from both pipelines need

(a) An interpolated bounding box from IEMOCAP.

(b) An interpolated bounding box from DFDC.

Figure 4.5: Bounding boxes predicted by BlazeFace.

another step of processing to work rightly with it. Now we are going to describe it. In both pipelines we generate a (300, 32) vector for every audio and video frame as previously mentioned in Section 4.6.3. To correctly pass data to Lazypredict, we need a dimensionality reduction. To do this without loss of data information, we exploit our vectors' statistical characteristics. For every column of EMO_s (defined in equation 4.5) or EMO_v (defined in equation 4.7) matrices we receive, we compute three statistical moments: mean, variance and kurtosis. Obtaining for every emotion feature matrix a 96-dimensional vector we can feed to Lazypredict coupled with the corresponding label. The best performing model Lazypredict returns for EMO_s matrices classification is a Light Gradient Boosting Machine (LGBM) classifier. We will use its predictions in the decision-level fusion approach. It was created by Guolin Ke at Microsoft in 2016 [121]. It is designed to be light, to use low memory resources and to be capable of handling large-scala data. As all gradient boosting techniques, it produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. Firstly, builds a model with the boosting method, i.e. generates models consecutively giving more and more weight to the errors made in previous models. Then, it generalizes them by allowing optimization of an arbitrary differentiable loss function. It is now one the most successful Machine Learning (ML)

algorithms.

4.8 Used Datasets

This section presents the datasets that we used for the training and evaluation stages of both the discrete model emotion recognition and deepfake detection tasks.

4.8.1 IEMOCAP

In this work, we trained emotion classification models on IEMOCAP, "interactive emotional dyadic motion capture database" [87]. Dyadic stands for the characteristic of the dataset of being recorded with couples dialogues. The database contains performances of ten actors: 5 males and 5 females. Age and genders are significant aspects of databases, as the quality of data affects the recognition performances. The actors were asked to perform selected scripts with explicit emotional content. Subjects were also asked to improvise dialogues in hypothetical scenarios for a total of approximately 12 hours of content. We focused on the improvised dialogues. We considered them to elicit more genuine emotions, allowing a better generalization capability of our video and audio models, although making the emotion recognition task more difficult. This leverages the underlying idea of this thesis that emotion inconsistencies in videos are correlated with synthetic manipulations. We planned to apply trained emotional models to DFDC dataset.

IEMOCAP performances, both scripted and spontaneous, were recorded with cameras, microphones and markers on face, head and hands. Initially designed to target anger, sadness, happiness, frustration and neutral state, IEMOCAP labels also include disgust, fear and surprise. During the dataset creation, it became clear that those 5 initial classes were too poorly descriptive of the sentimental states elicited by the actors from a human point of view. The most present emotions are, however: happiness, sadness, neutrality, anger and frustration. In the present work, we trained the system on the following classes: neutrality, anger, happiness, sadness and a fifth category grouping all other emotions in the dataset. This is because the first four emotions are among the most common emo-

tional descriptors found in literature [122]. We also followed previous works of IEMOCAP's designers, Busso and Narayan [123, 124, 125, 126], and remained consistent with prior IEMOCAP research [127]. The latter-is the article from which we started building our speech emotion recognition models.

IEMOCAP, along with audiovisual and gesture content for each performance, contains labeled dialogues transcriptions. These are divided in utterances, which were evaluated in their emotional content by six selected evaluators [87]. Its content is divided into five folders, one for a dyadic session. For our experiments, we used improvised content from every folder.

4.8.2 DFDC

For our deepfake detection work, we used a subset of the Deepfake Detection Challenge dataset (DFDC) [113]. The full dataset contains approximately $120,000$ clips, of which, $100,000$ are labeled as *Fake*, and the rest as *Real*. Videos are sourced from $3,426$ paid actors and actresses speaking in various settings and lightning for roughly 15 minutes each.

Fake videos in the dataset were created with different approaches, most of them with DFAE and three different GANs. The set of models selected was chosen to cover some of the most popular video faking systems when the dataset was created. In addition, some approaches with less realistic outcomes were included in order to represent low-effort deepfakes. These refer to techniques that compute facial landmarks on the source and target images, then morph pixels from the source image to match the landmarks in the target one. However, the number of videos per faking method is not equal. The majority of face-swapped videos were created with the DFAE architecture. This choice was made when creating the dataset to reflect the distribution of public deepfake videos. Several of these varieties, on genuine and synthetic clips, were designed explicitly by the database builders. They aimed to make it possible for a detection model trained only on DFDC to generalize on real "in-the-wild" deepfakes.

Coming to DFDC structure, clips are divided into 50 folders, numbered from 0 to 49. Each comprehends a set of *Real* videos, along with

all derivative *Fakes*. While videos are largely visual-based fakes, some of them in divisions 45 to 49 contain falsified audio in addition to possible falsified video frames. To create a dataset for our experiments, we considered the 10 seconds-long clips within folders 45 to 49 that contained both faked video and audio, for a total of $6,848$ videos.

4.9 Signal Segmentation and Labeling

This section describes the labeling and segmentation criteria we have mentioned in this chapter during the explanation of the implemented method.

We remind that we performed emotion recognition and deepfake detection on two different dataset: IEMOCAP and DFDC respectively. IEMOCAP's video and utterance lengths vary widely in the time range, while our DFDC subset has fixed signals durations of 10 seconds. In order to allow consistent applications of our models to both dataset, we have to set labeling criteria valid for the two different data formats. We implemented two ways of extracting 10 seconds long time fragments from IEMOCAP and two ways to label these. We found one of the combinations to give better classification results as described in Section 5.4

IEMOCAP is provided with dialogues transcriptions indexed by utterance names and timestamps. Each utterance is associated with evaluating its emotional content under the hypothesis that the expressed emotion is stable for the length of the utterance itself. The labeling is done by six different evaluators who operated independently. Each utterance emotional content is estimated by three of them to add consistency to the dataset. To extract audio and video frameset from each video segment, we require a match between at least two of these evaluations. We extract the correspondent labels well. A visual representation of the process is given in Figure 4.6.

The first proposed audiovisual vectors extraction is to only consider those utterances that are at least 10 seconds long. This led to a scarcity of data, as these required utterances occur rarely in IEMOCAP, which has an average utterance duration of only $4,5$ seconds. The classification trials with these inputs give in fact poor perfomance for every proposed model both video and audio as shown by the experiments in Section5.3.1

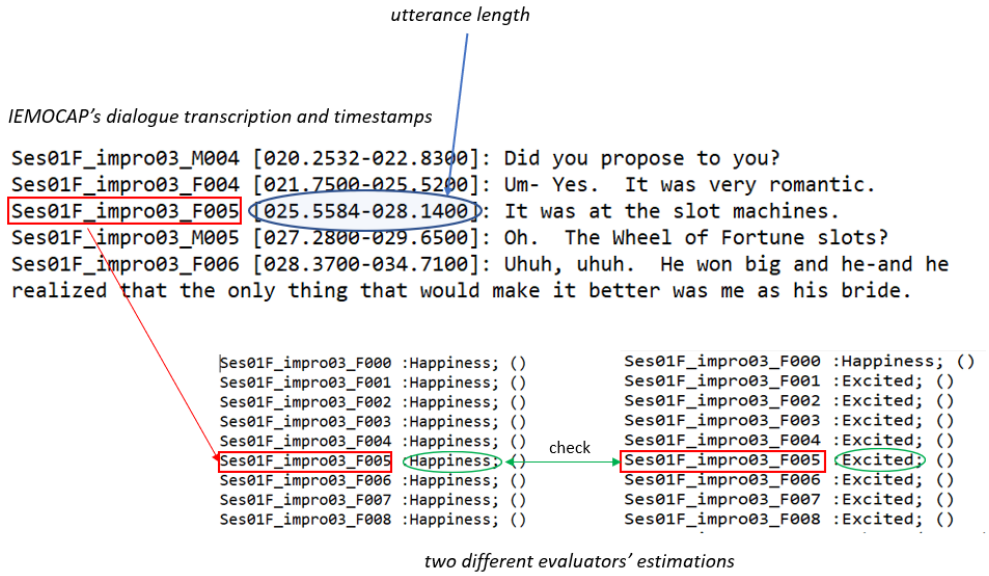


Figure 4.6: Visual representation of IEMOCAP labeling process.

and 5.4.1. We then relaxed the time constraint to one second. Shorter utterances in IEMOCAP contain single words or exclamations. We consider these occurrences as non-optimal and noise-inducing for our chosen emotional classes. We propose to: extract vectors with this new time constraint from every utterance labeled as we currently require, per video; subsequently, for every actor, aggregate all extracted vectors with the same label to 10 seconds long clusters of audiovisual samples. Our goal is to extract emotional information from signals. Basing ourselves on an intuitive understanding of emotions and supporting our claim with Ekman's et al. studies [60, 128, 95, 61], Frick's [74] and other recent findings resumed by Vankudre et al. in [14] and by Sidorov et al. in [129], we can sustain that the same cross-cultural emotions are characterized by and are detectable with the same audio features and visual cues. On the other hand, DFDC contains real and face-swapped clips labeled as *Real* or *Fake* respectively. There are no multiple evaluations, so there is no need to check if the labeling is consistent. Extracted acoustics and visual vectors are labeled consequently. There is also no need to process its clips further, as all the signal segmentation we previously described for IEMOCAP was designed to respect our DFDC's subset data format.

4.10 Conclusive Remarks

In this chapter, we have formally addressed the problem tackled in this thesis and reported the proposed scheme for its solution. Then, we gave the implementation details of the system we adopted and provided descriptions of the tools we utilized. In the following chapter, we define metrics we used for our experiments evaluations then we show their results.

5

Metrics, Experiments and Results

In this chapter, we provide all the details related to our experiments. We present metrics and training parameters we used for experimental validation. We introduce all the performed experiments, describing each setup in detail, dividing between emotion recognition and deepfake detection tasks.

5.1 Metrics

To evaluate the performances of our experiments, we adopt different metrics. The emotion recognition task is a multiclass classification problem, while deepfake detection has a binary nature. They differ conceptually as binary classification problems refer to those situations where just two classes are present: True and False. On the other hand, multiclass classification problems describe situations with more than two labels. In both cases, the task is to classify data labels correctly. Balanced accuracy is a metric valid for both problems, as we will describe, there is no conceptual difference for the two balanced accuracy definitions apart from

a slightly difference in the formal formulation. We now introduce the concepts of True Positive Rate (TPR) and True Negative Rate (TNR) to define it. TPR, commonly called recall or sensitivity, is defined as the ratio between the true positive and the total amount of positive samples, i.e.

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{5.1}$$

where TP are the true positives and FN are the false negatives (i.e., positive samples detected as negatives). The TNR, or specificity, is defined as the ratio between the true negatives and the total amount of negative samples, i.e.

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}, \tag{5.2}$$

where TN are the true negatives and FP are the false positives (i.e., negative samples detected as positives). For a binary classification problem we can now define Balanced Accuracy (BA) as the average between the TPR and TNR, namely

$$\mathrm{BA} = \frac{\mathrm{TPR} + \mathrm{TNR}}{2}. \tag{5.3}$$

To give a simple multiclass BA definition, we can call m_n the number of samples belonging to the class n and r_n the number of correctly predicted samples belonging to the same n class. Having this, with N number of classes, we can write

$$\mathrm{BA} = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{r_n}{m_n} \right), \tag{5.4}$$

giving this metric a sort of generalized formulation. In our experiments, $N = 5$ for the emotion recognition task and $N = 2$ for deepfake detection. In the last case the two BA definitions coincide. The balanced accuracy metric is helpful for concisely evaluating the system's performance with a single scalar value. The definitions of TPR and TNR rates can also be visualized and described with another metric we use, the confusion matrix. For N classes, the confusion matrix is a table with N rows and N columns. Binary and multiclass classifications confusion matrix structure differ as multiclass is a generalized form of the binary one. For the sake of clarity here we describe both structures. In the binary classification

context, for deepfake detection, the structure we adopt is

$$\begin{bmatrix} \text{TN} & \text{FP} \\ \text{FN} & \text{TP} \end{bmatrix}, \tag{5.5}$$

where we consider as negatives, clips that are labeled as *Real* and positives those labeled as *Fake*. In a multiclass context, the confusion matrix structure we adopt is more complex. The table is a square matrix where the correct predictions are shown on the main diagonal. All the errors are outside it. We compute each class's TP, TN, FP and FP values separately, following the main diagonal. A visual example is given below, considering $N = 3$ classes, where each TP value indicates the considered class.

$$\begin{pmatrix} \text{TP} & \text{FN} & \text{FN} \\ \text{FP} & \text{TN} & // \\ \text{FP} & // & \text{TN} \end{pmatrix} \begin{pmatrix} \text{TN} & \text{FN} & // \\ \text{FP} & \text{TP} & \text{FP} \\ // & \text{FN} & \text{TN} \end{pmatrix} \begin{pmatrix} \text{TN} & // & \text{FN} \\ // & \text{TN} & \text{FN} \\ \text{FP} & \text{FP} & \text{TP} \end{pmatrix} \tag{5.6}$$

We compute the TN, FP and FN values of each class by summing the corresponding TN, FP and FP values related to the other classes.

Another metric we adopt for binary classification performance evaluation is the Receiver Operating Characteristic (ROC) curve [130]. The ROC curve is created by plotting the TPR against the False Positive Rate (FPR) at various probability threshold settings. The FPR is defined as the ratio between the false positive and the total amount of negative samples, having

$$\text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}}. \tag{5.7}$$

Or, in a more immediate formulation

$$\text{FPR} = 1 - \text{TNR}. \tag{5.8}$$

ROC curves allow visualizing relevant characteristics for classification tasks, being the comparison of the two aforementioned important operating characteristics. A key scalar parameter to correctly read a ROC curve is the Area Under the Curve (AUC). It is equal to the probability that a classifier will label a randomly chosen positive instance higher than a randomly chosen negative one [131]. An AUC equal to 0.5 corresponds to random guesses, whereas an AUC equal to one corresponds to the perfect classification model. The ROC curve is a standard metric in machine

learning for evaluating the performance of binary classifiers. Moreover, plotting multiple curves on the same graph provides an immediate graphical comparison between different evaluation setups or techniques.

5.2 Training Parameters

Both feature extraction and classification stages are data-driven. Hence, training stages are required for every implemented model. We have trained the Speech Emotion Recognition (SER) system following the specifics proposed in [127]. This means that, for the SER training alone, we performed feature extraction on audio tracks. We normalized each signal between -1 and 1. Specifically, we have used the improvised dialogues from the IEMOCAP dataset of which we considered the classes of *Anger*, *Happiness*, *Sadness*, *Neutral*. We then considered a last class we called *Other* for every other label in the dataset. Called N the number of classes, we have $N = 5$. Since IEMOCAP is divided into 5 dialogue sessions containing different actors, we have selected different sets of sessions for training and testing phases of our models to let our results remain speaker-independent. Following the specifics in [127], we generally trained our models on the first four sessions and tested them on the fifth session. We have used Adam optimizer with learning rate $l_r = 3 \cdot 10^{-4}$ and categorical cross-entropy as loss function.

In our system, the SER network acts both as an emotion estimator and a feature extractor for subsequent tasks. We present the input vectors and feature vectors dimensions in each experiment separately. As we will show in Section 5.3 those were occasionally changed empirically, according to the best-obtained performances. The train sets have been balanced with the repetition of samples belonging to classes with lower cardinalities. This produces an increase in classification performance at the inherent expense of generalization. Despite this, we decided to carry out this process as the classification results obtained on the test sets are still good. Referring now to the video emotion recognition task, we remained consistent with the training criteria we used for the previous model. We implemented the network for the task subsequently to the SER one. We have used the improvised dialogues from the IEMOCAP dataset considering the same classes mentioned above. We obtained the

best results for emotion prediction using Adam optimizer with a learning rate $l_r = 2 \cdot 10^{-4}$ and categorical cross-entropy as loss function. Specularly to our SER network, our video model acts both as an emotion estimator and a feature extractor. We will present input vectors and feature vectors dimensions while describing the relative experiments.

For deepfake detection task we used a different dataset. We considered a subset of the DFDC, composed by its folders from 45 to 49. For every experiment we performed for this task, we divided it as follows: sessions 45, 46, 48 combined as the training set, session 47 to validate our model and session 49 to test it. When we refer to it simply as DFDC we are still implying this implemented split. For the LSTM classifier we implemented for classification we used Adam optimizer with a learning rate of $l_r = 4 \cdot 10^{-4}$ and categorical cross-entropy as function. From now on, when it is not specified otherwise, we will refer to Balanced Accuracy (BA) simply as accuracy.

5.3 Speech Emotion Recognition Results

In this section, we describe in detail the SER experiments we conducted and provide their results. To be consistent with prior research on IEMOCAP we begin implementing Tripathi et al. models [127].

5.3.1 Experiment I

The first model proposed in [127] is a fully connected MLP with 1024, 512, 256 hidden neural units. It is fed with feature vectors of size (100, 34) for each 10 seconds utterance. With ReLU as activation function and 4 output neurons with softmax they achieve 0.506 accuracy on the four classes of: *Anger*, *Happiness*, *Sadness* and *Neutral*. They use the first 4 IEMOCAP sessions as their train set and the last one as test set.

We implemented the same network architecture and setup to replicate the article performances and have a first speech emotion classifier. With the same time constraint of 10 seconds, we extract (100, 39) feature vectors per utterance. We obtain features using OpenSmile in the so-called EmoBase2010 configuration. With this setup, we achieved 0.334 balanced accuracy on the same 4 classes. It is worth noticing that the

Table 5.1: List of features extracted by OpenSmile with the EmoBase2010 configuration [17].

1 energy related LLD	**Group**
Loudness as normalized intensity	prosodic
32 spectral related LLD	**Group**
MFCCs (0-15)	spectral
logpower of 8 Mel-frequency bands (0-8kHz)	spectral
8 LSP frequencies derived by 8 LPC coeffs.	spectral
6 voicing related LLD	**Group**
F_0 (SHS & Viterbi smoothing)	prosodic
Jitter (local & δ), Shimmer (local)	voice qual.
Prob. of voicing	voice qual

two sets of extracted features differ, especially as we do not focus on the chromagram-based ones. The paper mentions 13 Mel Frequency Cepstral Coefficients (MFCC), 13 chromagram-based features and 8 spectral features: zero-crossing rate, short-term energy, short-term entropy of energy, spectral centroid and spread, spectral entropy, spectral flux, spectral roll-off. Table 5.1 provides the complete list of features we extracted in our implementation. We point out that, at this stage, we do not apply any normalization process. We also highlight that the implemented model is computationally heavy.

5.3.2 Experiment II

In this experiment we want to achieve better emotion estimation accuracy on IEMOCAP's audio tracks. To do this, we refer to the second speech emotion recognition model proposed in [127] which was reported as having higher performances of 0.5132. We implement a LSTM network based on the one described in the article. Our goal with this architecture is to infer time dependencies in the signal that a MLP architecture would not have learned. We already provided a description of the implemented model in Section 4.6.1. It is the one we refer to as LSTM_I. Compared to the architecture we used in the experiment 5.3.1, we also reduce the number of trainable parameters by more than a factor of 10, thus reducing

computational complexity. To better exploit the LSTM layers capability of learning time-dependent characteristics, we collect more descriptor vectors per second. We now extract 100 vectors per second. This being allowed by OpenSmile time resolution. We also shortened the time context to 3 seconds. The choice is reasonable as it takes into account the mean IEMOCAP utterance duration of 4.5 seconds. We then switched to the previously described ComParE2016 configuration (see Section 4.7.1 for details) to augment the amount of extracted audio descriptors, having more informative data inputs. We therefore fed our LSTM model with (300, 130) LLD_s vectors. We obtained comparable to the paper results with 0.54 accuracy on improvised data.

5.3.3 Experiment III

Maintaining the same model of the previous experiment 5.3.2, we added another emotional class to the problem. A label for every other emotion recorded in IEMOCAP. The goal is now to augment our model's emotion description capabilities to provide more representative characteristics in time for the later deepfake detection stage. We perform this experiment to show the performances we obtain with the implemented model on $N = 5$ classes.

First implementation.

To obtain more training data, we implemented overlap between time windows we extract features from. From now on, we also scale our features as mentioned in 5.2 between -1 and 1 values. With a 67% overlap, namely 2 seconds overlap, we obtained 0.557 accuracy.

Second implementation

With the same setup, we implemented a control on provided emotion evaluations. Each IEMOCAP utterance is labeled by three different evaluators. To consider an utterance as valid, we now require at least two evaluations to be identical. From now on, we implemented this check step for every experiment conducted on IEMOCAP. We obtained 0.44 accuracy on improvised data, interpreting this fall in recognition rate with data reduction following the addition of the new constraint.

We now introduce the experiments on video emotion recognition and

their results for further explanation on the experimental campaign.

5.4 Video Emotion Recognition Results

Here we provide descriptions and results of our experiments on emotion recognition performed on IEMOCAP's video tracks. We based our models on [16]. As we built our image-based system specularly to the speech one, we needed a single model to act both as a classifier and a feature extractor. Since we want to collect time-dependent cues, we also need a model to capture the time dependencies between video frames. 3DCNN typology was the choice for this task network.

3DCNN based on the work of Ji et al. [16]. CNNs are among the best architectures for image analysis and though 3DCNNs we can also capture the temporal information encoded in adjacent frames. For the first experiments, we tried a more straightforward implementation of the model. We experimented with 2 layers of convolution and tried to pass 300 adjacent frames. We used the model to extract visual features directly from the internal layers of the net.

5.4.1 Experiment I

This experiment aims to show the performances of an architecture proposed for a visual but not emotion-based task to video emotion recognition. The model presented in [16] performed human action recognition with good results, 90.2 accuracy on average over 6 classes of actions. For the first experiment, we implemented a simplified version of the proposed model. Our 3DCNN was composed of a first 3D convolutional layer with 2 filters, a 3D max-pooling layer for feature dimensions reduction, a dropout layer, a second 3D convolutional layer with 32 filters and an output dense layer with 5 neurons. Moreover, at this stage we do not process frames as described in [16]. We proceeded to the audio pipeline. We perform classification on 5 classes with the same division of folders. We check for emotion evaluation consistencies and initially consider a time constraint for utterances of 10 seconds. Given IEMOCAP's frames per second rate of $fps_{iemocap} = 30Hz$, with this time context, we feed the network with vectors of 300 frames from which we obtain promising

results with over 0.64 accuracy on average. We point out that this result was obtained with a manual crop on IEMOCAP frames. We pass from 470x720 dimensions to 100x93, basing on the required actor position for IEMOCAP instrumentation correct recording. We center our crop window on the center of the left speaker window. It is now essential to focus on the labeling method, on which we gave a detailed description in Section 4.9. We label each frame in input vectors individually. By doing this, we discard time information contained in adjacent frames. For our deepfake detection task, however, we need to correctly model visual temporal cues of displayed emotions.

5.4.2 Experiment II

For the following experimental setup we changed our labeling method and maintained unchanged our network model. The goal of this experiment is to correctly address temporal features of emotional evolutions in time. We gave a single label for each of the 300 frames vectors we extracted from 10 seconds time clusters created as described in Section 4.9. With the usual data division, we obtain 0.52 accuracy on the five classes.

It is now crucial to highlight some experimental passages to provide further descriptions of the experiments. We extracted speech emotion descriptors applying the SER model of the third experiment 5.3.3 on DFDC audio tracks. Its performances were not excellent: 0.54 accuracy on available data and 0.44 on the restricted subset. However, binary classification on the so extracted $\mathrm{EMO}_s(t)$ time-series in the last stage obtained good scores with Lazypredict: over 0.88 accuracy on the two *Real* and *Fake* classes. Being the accuracies of this experiment architecture and the one considered for speech tracks comparable on the same dataset, we extracted $\mathrm{EMO}_v(t)$ time-series from this model. This is based on the hypothesis that similar behavior was possible for the video pipeline, being the two pipelines constructed specularly. Given this, we extract visual emotion time-series from DFDC video tracks and classify them. We obtain low classification results. The best Lazypredict classifier performs to 0.52 accuracy. Implementing an LSTM model to better address temporal information, the one we refer to as $\mathrm{LSTM}_{\mathrm{II}}$, we obtain a flat 0.5.

At this point, we tried to extract $\text{EMO}_v(t)$ time-series from the same 3DCNN model. Still, we changed the activation function for the convolutional layer from which we extract feature vectors, moving from a softmax function to the tanh function used in the article. With this new configuration, performances of both emotion recognition and deepfake detection tasks raise. We obtain 0.64 accuracy for emotion recognition and over 0.52 accuracy for both deepfake classification approaches. Encouraged by the new results on the emotion recognition task, we implemented the exact 3DCNN structure proposed in [16].

5.4.3 Experiment III

This experiment aims to adapt the exact 3DCNN model for human action recognition to video emotion recognition and show its performances. We give a detailed description of the model used for this experiment and its frame processing method in Section 4.6.2. With the new frame processing required for the network, we remove the time constraint and start labeling each MAP vector, defined as in 4.6, with the utterance label the frames to process are taken from. From our experimentation, we find 7 to be the optimal number of adjacent frames to consider. We manually crop IEMOCAP frames from 470x720 to 110x95, centering our crop window on the center of the left speaker window. We obtain the best results by reducing cropped frames to 80x60 dimensions with bicubic interpolation. We report that this configuration is memory demanding so, we reduced our training set to the first two IEMOCAP sessions. We test our model on the fifth.

5.4.4 Experiment IV

This experiment aimed to show the performance of video emotion recognition with our implemented model and less memory demanding setup. Our approach to frame extraction changes as we apply BlazeFace face detection on each IEMOCAP frame. From now on, for every video-related task, we will use this face detector toolkit. With the software, we extract and save the face bounding boxes the tool predicts. We reduce them to the same 80x60 dimensions of the previous experiment with bicubic interpolation. We group and process 7 frames at a time, labeling them as in

the previous experiment. With this configuration, we train our 3DCNN model. We obtained good classification results using the first three sessions as the training set and the fifth as the test set. We provided some examples of BlazeFace-extracted faces in Section 4.7.2, Figure 4.5.

5.5 Deepfake Detection Results

Here we give results and details of the experiments we conducted for deepfake detection. To discriminate between altered and genuine videos, we use both their audio and visual components. Separately as a first approach, combined as a second. $\text{EMO}_s(t)$ and $\text{EMO}_v(t)$ are the time-series representing the evolution in time of the emotional content of considered videos. We label both emotional components in every (300, 32) matrix we extract from emotion recognition networks.

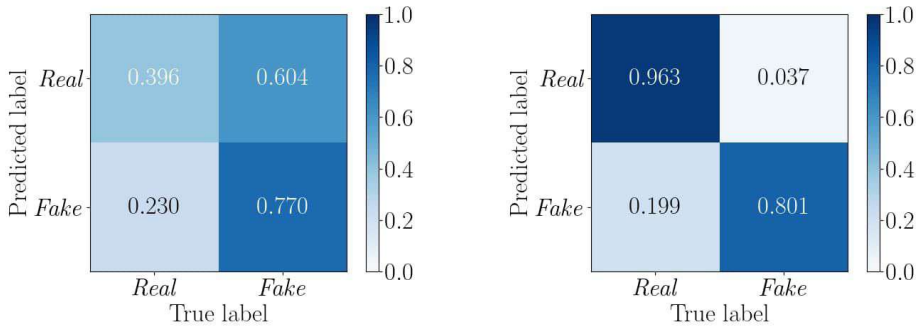
5.5.1 Audio Deepfake Detection

The following experiments have been performed to show our performances on deepfake detection considering only the emotional features obtained by audio tracks. To classify DFDC's $\text{EMO}_s(t)$, we use a LSTM resembling the one implemented for the SER task. We refer to it as LSTM_{II}. The architecture that gives the best performances is composed of: a 32 filters LSTM layer, a dropout and two dense layers. The first with 8 neurons and ReLU activation function and the last with 2 and softmax for its binary classification task. Referring to the division of the folder we mentioned in Section 5.2, this model discriminate between *Real* and *Fake* videos with accuracies up to 0.5838. We compared this result with the Lazypredict multiclassification approach, obtaining the best accuracy of over 0.882 with its best model, an LGBM classifier.

As we can see in Figure 5.1 LSTM_{II} recognize more accurately fake videos than real ones using speech emotion features. However, the LGBM classifier overall performance is better.

5.5.2 Video Deepfake Detection

We conducted this set of experiments to obtain the performances on deepfake detection considering only video tracks. We obtained the best



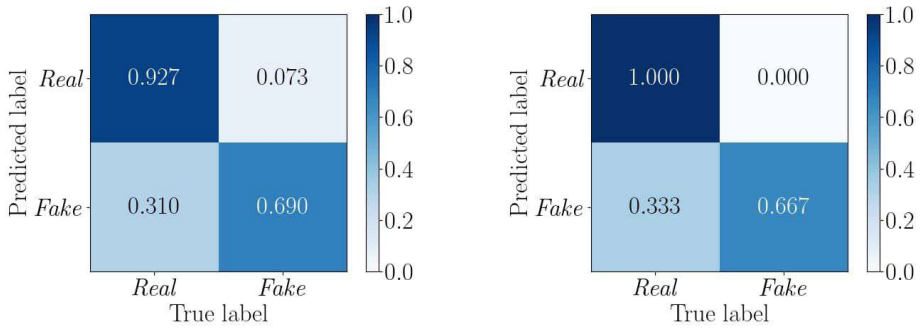
(a) Discrimination performed with $LSTM_{II}$ classifier.

(b) Discrimination performed with LGBM classifier.

Figure 5.1: Confusion matrices comparison for deepfake detection based on audio.

classification results using $EMO_v(t)$ as descriptors, was 0.57 accuracy for both $LSTM_{II}$ model and the best performing Lazypredict classifier, which was a Support-Vector Machine (SVM) for this modality. We followed a new approach to tackle this problem since the recognition rates with video emotion features were low even when extracting them from our best video emotion recognition network. We applied the 3DCNN model directly to DFDC videos without generating any feature. We utilized it as a mere binary classifier. To do this, we substituted the output layer with 5 neurons and softmax of the 3DCNN video emotion recognition network with a layer of 2 neurons layers with softmax. We trained it directly on DFDC clips using the same folder division we used for the audio experiments. With this approach, we achieved average accuracies slightly below 0.83, with a minimum of 0.8242 and a maximum of 0.8304 By doing this, we are no longer considering emotions.

To take emotion classification into account, and therefore the emotion characteristics of DFDC clips, we tackled the problem with a new approach. We used the IEMOCAP trained 3DCNN model and re-trained it on DFDC. We took the trained model and substituted its 5 neuron output layer with a 2 neuron dense layer. We processed DFDC frames as described in Section 5.4.4. By doing this, we are no longer extracting speech and visual cues from the same time regions. We have feature-level decoupled audio and video discrimination systems. With these double training stages, we say the network is taking emotions into account. This



(a) Discrimination performed with 3DCNN model directly applied to DFDC.

(b) Discrimination performed with 3DCNN model with re-train stage.

Figure 5.2: Confusion matrices comparison for deepfake detection based on video.

configuration obtains accuracies slightly above 0.83, with a minimum value of 0.8296 and a maximum of 0.8341.

As we can see from Figure 5.2 the two models perform very similarly on the same data. However, the double-trained 3DCNN model has a moderate edge over the other approach.

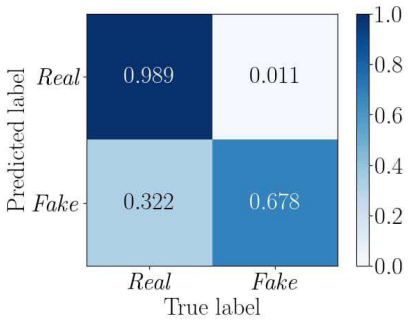
5.5.3 Bimodal Deepfake Detection

This set of experiments was conducted to show our performances when combining the two classification modalities in a bimodal approach. At first, we implemented a feature-level fusion with both classification models. We combine each (300, 32) EMO_s and EMO_v matrix horizontally, into (300, 64) feature matrices. For both classification approaches, we obtain that one of the feature sets dominates the other. With Lazypredict classifiers, we obtain the exact results of the audio modality alone, 0.88. With the $LSTM_{II}$ model we obtain the same classification results of the video modality alone 0.57.

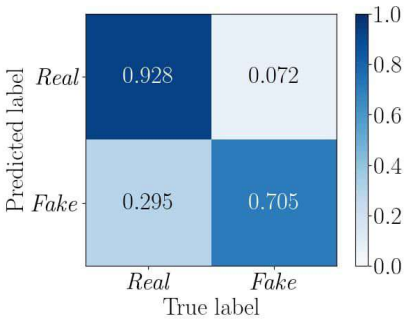
We implemented the decision-level fusion by computing the average between the classification scores given by models on the two modalities separately. We can do this with different model combinations. For a visual representation of the results, we refer to bimodal confusion matrices given in Figure 5.3.

We achieve accuracies of:

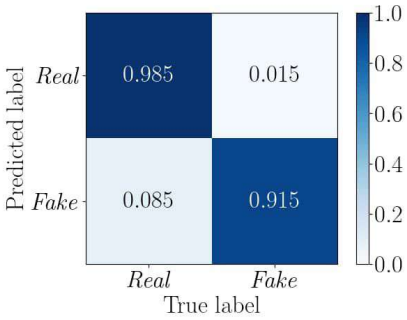
- 0.8284 with $LSTM_{II}$ and emotion-considering 3DCNN;



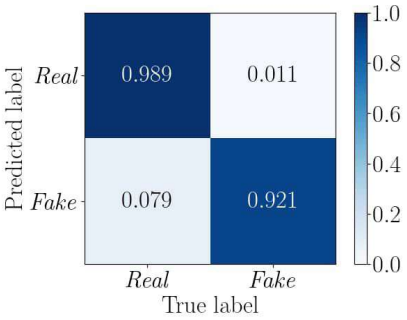
(a) Bimodal detection performed with $LSTM_{II}$ and emotion-considering 3DCNN model.



(b) Bimodal detection performed with $LSTM_{II}$ and 3DCNN model trained only on DFDC.



(c) Bimodal detection performed with LGBM classifier and 3DCNN model trained only on DFDC.

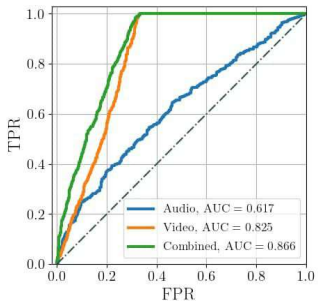


(d) Bimodal detection performed with LGBM classifier and emotion-considering 3DCNN model.

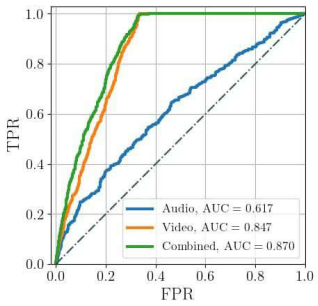
Figure 5.3: Confusion matrices comparisons for different decision-level fusion approaches.

- 0.8304 with $LSTM_{II}$ and emotion-independent 3DCNN;
- 0.9503 with LGBM classifier and emotion-independent 3DCNN;
- 0.9532 with LGBM classifier and emotion-considering 3DCNN;

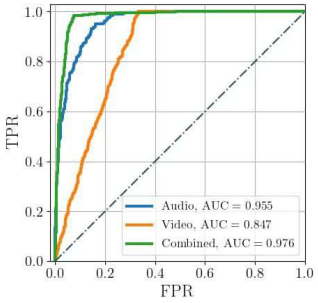
These results shows how combining modalities is a valid approach to deepfake detection with our system. Bimodality allows achieving better performances compared to single modalities. In particular, the approach with LGBM classifier and the double trained 3DCNN model combined give, on average, our best results with a reasonable margin over that obtained with LGBM combined to the emotion-independent 3DCNN. Plots in Figure5.4 of the respective ROC curves gives some more insights on these decision-level fusions. They show how well the LGBM classifier performs compared to the $LSTM_{II}$ model in classifying $EMO_s(t)$ time-



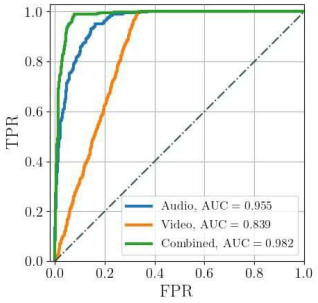
(a) Bimodal detection performed with $LSTM_{II}$ and emotion-considering 3DCNN model.



(b) Bimodal detection performed with $LSTM_{II}$ and 3DCNN model trained only on DFDC.



(c) Bimodal detection performed with LGBM and 3DCNN model trained only on DFDC.



(d) Bimodal detection performed with LGBM and emotion-considering 3DCNN model.

Figure 5.4: ROC curve comparisons for different decision-level fusion approaches.

series. They also show how the bimodal approach is better compared to single modalities. The ROC curves of the best unimodal approaches are always very similar to the bimodality one. This is expected, as it indicates that the best performing unimodal approach gives a significant contribution to the overall performance. We can also see that in the worst case bimodality enhances the probability of rightly predicting a class, being it *Real* or *Fake*, by 2.3%.

5.6 Conclusive Remarks

In this chapter, we analyzed the results of the set of conducted experiments to evaluate the proposed system's different components. While some of the experiments did not excel in their task, we remained consis-

tent with performances mentioned by the paper we followed. Moreover, the results of the experiments gave an overall positive response to the proposed approach. In particular, the last bimodal combination evaluated in 5.5.2 performs with reasonable accuracies.

The next chapter summarizes the work done and provides some possible future works to expand it further.

6

Conclusions and Future Works

The advancements of AI-based technologies allow believable multimedia manipulations. In a society as connected as ours, with the pervasivity of connected devices and social media importance in communication nowadays, the need for controls over shared media entity and origin is as delicate as necessary. Diffusion of realistic altered contents and fake news pose severe threats to both society and the individual. Moreover, these AI-based techniques evolve as rapidly as they spread. In this scenario, it is no surprise that the number of deepfake related publications raised almost 6 times between the years 2018-2019 and again by more than 3 times between 2019 and 2020.

In this work of thesis, we proposed a system to discriminate between genuine and synthetic media. We tackled our goal as a binary classification problem. The system is based on two main pipelines based on audio and video components of multimedia content. The first one exploits state-of-the-art Speech Emotion Recognition (SER) techniques to perform emotion recognition on speech signals. It later extracts emotional features from clips of which we want to infer the nature. The second one

is a video emotion recognition-based pipeline, built specularly to the first one, from which we obtain both emotion predictions based on visual cues and video emotion features for further deepfake detection. The pipelines are fused in a further classification stage as we use extracted emotional features to discriminate between real and fake videos. We experimented with two kinds of fusion: feature and decision-level. The latter giving promising results achieving balanced accuracy up to 0.9532. However, many improvements may still be implemented to improve the proposed methodology further.

Classification performed with speech content alone gives better results than the video modality. We can read this result in two ways. The first conclusion we can make is that audio techniques are more effective for this task. The second could be that when creating a deepfake, more effort is put on creating the video more than its audio content. Thus visual content could be more accurate and refined, making detection based on visual manipulation more difficult. Given this, the first suggestion goes in the direction of strengthening the emotion recognition models. The networks could be upgraded with attention layers for both modalities to enhance the semantic quality of learned features. The video model could individuate, for example, critical facial regions for emotion conveyance and focus on them in the training phase.

Another improvement on the SER model could be implementing bidirectional LSTM layers to obtain a better understanding of the considered time context. By this, extracted speech features could be more informative. Another way to improve the informative content of extracted characteristics could be to consider additional classes in the Emotion Recognition (ER) task. This could be done by leveraging the idea of generalizing the semantic content of collected emotion features.

The network models could also be strengthened by working on their parameters. It could be implemented automatic research of the hyperparameters. At this point, the task could be optimized by analyzing the sensitivity of the models as the latter varies.

Moreover, based on the proposed system, an interesting experiment could be to implement a fusion stage for the two emotion recognition pipelines. This would not influence the deepfake recognition performances but could provide good results for the ER task. As we did for

deepfake detection, this fusion could be performed both feature-level and decision-level.

Bibliography

[1] C. Olah, "Understanding LSTM Networks." http://colah.github.io/posts/2015-08-Understanding-LSTMs, 2015. Accessed: 2021-09-03.

[2] M. Plotke, "2D Image-Kernel Convolution Animation." https://it.wikipedia.org/wiki/Matrice_di_convoluzione#/media/File:2D_Convolution_Animation.gif, 2013. Accessed: 2021-09-08.

[3] A. G. Walters, "Convolutional Neural Networks (CNN) to Classify Sentences." https://austingwalters.com/convolutional-neural-networks-cnn-to-classify-sentences/max-pooling/, 2019. Accessed: 2021-08-26.

[4] "Thats what we call a selfie." https://aviationhumor.net/thats-what-we-call-a-selfie/, 2019. Accessed: 2021-09-08.

[5] R. Jahns, "Iced Venice." https://www.instagram.com/nois7/, 2014. Accessed: 2021-09-08.

[6] R. K. Wilson, "Surgeon's photograph," 1934.

[7] R. Kerckhoffs, "Cycling stimulants." https://twitter.com/raykerckhoffs. Accessed: 2021-09-08.

[8] "Atomic winter." https://twitter.com/oldpicsarchive/status/549538555262156800. Available: 2015-01-14.

[9] J. Peele, "You wont believe what Obama says in this video!." https://www.youtube.com/watch?v=cQ54GDm1eL0, 2018. Accessed: 2021-09-08.

[10] P. Wang, "thispersondoesnotexist." https://thispersondoesnotexist.com/, 2019. Accessed: 2021-08-26.

[11] Stryderstormcrow, "Nicolas Cage is Neo trailer [Deepfake]." https://www.youtube.com/watch?v=G6vi9XoMEZc&t=2s, 2020. Accessed: 2021-09-08.

[12] R. S. artificial intelligence lab, "AI brings Mona Lisa to life." https://www.youtube.com/watch?v=P2uZF-5F1wI, 2019. Accessed: 2021-09-08.

[13] CVPR, "Trump's deepfake." https://www.technologyreview.it/la-sfida-dei-deepfake, 2019. Accessed: 2021-09-08.

[14] G. Vankudre, V. Ghulaxe, A. Dhomane, S. Badlani, and T. Rane, "A survey on infant emotion recognition through video clips," in *2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, pp. 296–300, IEEE, 2021.

[15] T. T. Nguyen, C. M. Nguyen, D. T. Nguyen, D. T. Nguyen, and S. Nahavandi, "Deep learning for deepfakes creation and detection: A survey," *arXiv preprint arXiv:1909.11573*, 2019.

[16] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2012.

[17] audEERING, "opensmile." https://www.audeering.com/opensmile/. Accessed: 2021-09-03.

[18] E. Howcroft, "How faking videos became easy and why that's so scary," *Bloomberg: New York, NY, USA*, 2018.

[19] C. Wang, "Deepfakes, revenge porn, and the impact on women," *Forbes*, Nov. 2019.

[20] BBC News, "'deepfake' app causes fraud and privacy fears in china." https://www.bbc.com/news/technology-49570418, 2019.

[21] P. Yu, Z. Xia, J. Fei, and Y. Lu, "A survey on deepfake video detection," *IET Biometrics*, 2021.

[22] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li, "Protecting world leaders against deep fakes.," in *CVPR workshops*, vol. 1, 2019.

[23] X. Xuan, B. Peng, W. Wang, and J. Dong, "On the generalization of gan image forensics," in *Chinese conference on biometric recognition*, pp. 134–141, Springer, 2019.

[24] C.-C. Hsu, Y.-X. Zhuang, and C.-Y. Lee, "Deep fake image detection based on pairwise learning," *Applied Sciences*, vol. 10, no. 1, p. 370, 2020.

[25] N. Bonettini, E. D. Cannas, S. Mandelli, L. Bondi, P. Bestagini, and S. Tubaro, "Video face manipulation detection through ensemble of cnns," in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 5012–5019, 2021.

[26] B. Hosler, D. Salvi, A. Murray, F. Antonacci, P. Bestagini, S. Tubaro, and M. C. Stamm, "Do deepfakes feel emotions? a semantic approach to detecting deepfakes via emotional inconsistencies," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1013–1022, 2021.

[27] D. E. Rumelhart, R. Durbin, R. Golden, and Y. Chauvin, "Backpropagation: The basic theory," *Backpropagation: Theory, architectures and applications*, pp. 1–34, 1995.

[28] B. Hammer, A. Micheli, and A. Sperduti, "Adaptive contextual processing of structured data by recursive neural networks: A survey of computational properties," in *Perspectives of Neural-Symbolic Integration*, pp. 67–94, Springer, 2007.

[29] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019*, 2015.

[30] Y. Goldberg, "Neural network methods for natural language processing," *Synthesis lectures on human language technologies*, vol. 10, no. 1, pp. 1–309, 2017.

[31] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2017.

[32] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," *arXiv preprint arXiv:1605.05101*, 2016.

[33] I. Sutskever, J. Martens, and G. E. Hinton, "Generating text with recurrent neural networks," in *ICML*, 2011.

[34] 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.

[35] Y. Yu, X. Si, C. Hu, and J. Zhang, "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures," *Neural Computation*, vol. 31, pp. 1235–1270, 07 2019.

[36] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling," in *Thirteenth annual conference of the international speech communication association*, 2012.

[37] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[38] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2222–2232, 2016.

[39] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *The Journal of physiology*, vol. 195, no. 1, pp. 215–243, 1968.

[40] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, 2017.

[41] N. Aloysius and M. Geetha, "A review on deep convolutional neural networks," in *2017 International Conference on Communication and Signal Processing (ICCSP)*, pp. 0588–0592, 2017.

[42] J. Wu, "Introduction to convolutional neural networks," *National Key Lab for Novel Software Technology. Nanjing University. China*, vol. 5, no. 23, p. 495, 2017.

[43] O. Abdel-Hamid, L. Deng, and D. Yu, "Exploring convolutional neural network structures and optimization techniques for speech recognition.," in *Interspeech*, vol. 11, pp. 73–5, Citeseer, 2013.

[44] R. E. Turner, "Lecture 14: Convolutional neural networks for computer vision." `http://learning.eng.cam.ac.uk/pub/Public/Turner/Teaching/ml-lecture-3-slides.pdf`, 2014.

[45] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[46] L. Verdoliva, "Media forensics and deepfakes: An overview," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 910–932, 2020.

[47] J. Chen, B. Xie, H. Zhang, and J. Zhai, "Deep autoencoders in pattern recognition: a survey," in *Bio-inspired computing models and algorithms*, pp. 229–255, World Scientific, 2019.

[48] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[49] J. Kietzmann, L. W. Lee, I. P. McCarthy, and T. C. Kietzmann, "Deepfakes: Trick or treat?," *Business Horizons*, vol. 63, no. 2, pp. 135–146, 2020.

[50] "French charity publishes deepfake of trump saying aids is over.." `https://www.euronews.com/`, 2019. Accesed: 2021-08-30.

[51] B. Brucato, "The new transparency: police violence in the context of ubiquitous surveillance," *Media and Communication*, vol. 3, no. 3, pp. 39–55, 2015.

[52] B. Usukhbayar, "Deepfake videos: The future of entertainment," 03 2020.

[53] C.-H. Wu, J.-C. Lin, and W.-L. Wei, "Survey on audiovisual emotion recognition: databases, features, and data fusion strategies," *APSIPA transactions on signal and information processing*, vol. 3, 2014.

[54] M. Szwoch and W. Szwoch, "Emotion recognition for affect aware video games," in *Image Processing & Communications Challenges 6* (R. S. Choraś, ed.), (Cham), pp. 227–236, Springer International Publishing, 2015.

[55] X. Huahu, G. Jue, and Y. Jian, "Application of speech emotion recognition in intelligent household robot," in *2010 International Conference on Artificial Intelligence and Computational Intelligence*, vol. 1, pp. 537–541, 2010.

[56] L.-S. A. Low, N. C. Maddage, M. Lech, L. B. Sheeber, and N. B. Allen, "Detection of clinical depression in adolescents speech during family interactions," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 3, pp. 574–586, 2010.

[57] R. Plutchik, "The nature of emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice," *American scientist*, vol. 89, no. 4, pp. 344–350, 2001.

[58] M. B. Akçay and K. Oğuz, "Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers," *Speech Communication*, vol. 116, pp. 56–76, 2020.

[59] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang, "A survey of affect recognition methods: Audio, visual, and spontaneous

expressions," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 1, pp. 39–58, 2008.

[60] P. Ekman and H. Oster, "Facial expressions of emotion," *Annual review of psychology*, vol. 30, no. 1, pp. 527–554, 1979.

[61] P. Ekman, W. V. Friesen, and P. Ellsworth, *Emotion in the human face: Guidelines for research and an integration of findings*, vol. 11. Elsevier, 2013.

[62] E. A. Shaw, "Transformation of sound pressure level from the free field to the eardrum in the horizontal plane," *The Journal of the Acoustical Society of America*, vol. 56, no. 6, pp. 1848–1861, 1974.

[63] P. R. Shaver, S. Wu, and J. C. Schwartz, "Cross-cultural similarities and differences in emotion and its representation.," 1992.

[64] L. C. De Silva, T. Miyasato, and R. Nakatsu, "Use of multimodal information in facial emotion recognition," *IEICE TRANSACTIONS on Information and Systems*, vol. 81, no. 1, pp. 105–114, 1998.

[65] A. A. Ghazanfar and D. Y. Takahashi, "The evolution of speech: vision, rhythm, cooperatio," *Trends in cognitive sciences vol. 18*, pp. 543–553, 2014.

[66] Davidson and Iain, "On the evolution of language," *Current Anthropology*, no. 2, pp. 165–170, 1993.

[67] R. A. Khalil, E. Jones, M. I. Babar, T. Jan, M. H. Zafar, and T. Alhussain, "Speech emotion recognition using deep learning techniques: A review," *IEEE Access*, vol. 7, pp. 117327–117345, 2019.

[68] R. Bachu, S. Kopparthi, B. Adapa, and B. Barkana, "Separation of voiced and unvoiced using zero crossing rate and energy of the speech signal," in *American Society for Engineering Education (ASEE) zone conference proceedings*, pp. 1–7, 2008.

[69] H. Ghaemmaghami, B. Baker, R. Vogt, and S. Sridharan, "Noise robust voice activity detection using features extracted from the

time-domain autocorrelation function," in *Proceedings of the 11th Annual Conference of the International Speech Communication Association*, pp. 3118–3121, International Speech Communication Association, 2010.

[70] J. Pohjalainen, F. Fabien Ringeval, Z. Zhang, and B. Schuller, "Spectral and cepstral audio noise reduction techniques in speech emotion recognition," in *Proceedings of the 24th ACM International Conference on Multimedia*, MM '16, (New York, NY, USA), p. 670674, Association for Computing Machinery, 2016.

[71] Rao, S. K., Koolagudi, S. G., Vempada, and R. Reddy, "Emotion recognition from speech using global and local prosodic features," *International Journal of Speech Technology*, vol. 16, 2013.

[72] A. Nogueiras, A. Moreno, A. Bonafonte, and J. B. Mariño, "Speech emotion recognition using hidden markov models," in *Seventh European conference on speech communication and technology*, 2001.

[73] I. Luengo, E. Navas, I. Hernáez, and J. Sánchez, "Automatic emotion recognition using prosodic parameters," in *Ninth European conference on speech communication and technology*, 2005.

[74] R. W. Frick, "Communicating emotion: The role of prosodic features.," *Psychological bulletin*, vol. 97, no. 3, p. 412, 1985.

[75] W.-J. Yoon, Y.-H. Cho, and K.-S. Park, "A study of speech emotion recognition and its application to mobile services," in *Ubiquitous Intelligence and Computing* (J. Indulska, J. Ma, L. T. Yang, T. Ungerer, and J. Cao, eds.), (Berlin, Heidelberg), pp. 758–766, Springer Berlin Heidelberg, 2007.

[76] Kuchibhotla, Swarna, Vankayalapati, Vaddi, and Anne, "A comparative analysis of classifiers in emotion recognition through acoustic features," 2014.

[77] D. Bitouk, R. Verma, and A. Nenkova, "Class-level spectral features for emotion recognition," *Speech Communication*, vol. 52, no. 7, pp. 613–625, 2010.

[78] N. Sato and Y. Obuchi, "Emotion recognition using mel-frequency cepstral coefficients," *Information and Media Technologies*, vol. 2, no. 3, pp. 835–848, 2007.

[79] R. Cowie and E. Douglas-Cowie, "Automatic statistical analysis of the signal and prosodic signs of emotion in speech," in *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96*, vol. 3, pp. 1989–1992, IEEE, 1996.

[80] M. Borchert and A. Dusterhoft, "Emotions in speech - experiments with prosody and quality features in speech for use in categorical and dimensional emotion recognition environments," in *2005 International Conference on Natural Language Processing and Knowledge Engineering*, pp. 147–151, 2005.

[81] S. Zhang, "Emotion recognition in chinese natural speech by combining prosody and voice quality features," in *International Symposium on Neural Networks*, pp. 457–464, Springer, 2008.

[82] J. Kaiser, "On a simple algorithm to calculate the 'energy' of a signal," in *International Conference on Acoustics, Speech, and Signal Processing*, pp. 381–384 vol.1, 1990.

[83] J. Nicholson, K. Takahashi, and R. Nakatsu, "Emotion recognition in speech using neural networks," *Neural computing & applications*, vol. 9, no. 4, pp. 290–296, 2000.

[84] F. Burkhardt, A. Paeschke, M. Rolfes, W. F. Sendlmeier, and B. Weiss, "A database of german emotional speech," in *Ninth european conference on speech communication and technology*, 2005.

[85] P. Harár, R. Burget, and M. K. Dutta, "Speech emotion recognition with deep learning," in *2017 4th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 137–140, 2017.

[86] J. Zhao, X. Mao, and L. Chen, "Speech emotion recognition using deep 1d and 2d cnn lstm networks," *Biomedical Signal Processing and Control*, vol. 47, pp. 312–323, 2019.

[87] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "Iemocap: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, no. 4, pp. 335–359, 2008.

[88] J. Kim, K. P. Truong, G. Englebienne, and V. Evers, "Learning spectro-temporal features with 3d cnns for speech emotion recognition," in *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, pp. 383–388, IEEE, 2017.

[89] M. Liberman, "Emotional prosody speech and transcripts," *http://www. ldc. upenn. edu/Catalog/CatalogEntry. jsp? catalogId= LDC2002S28*, 2002.

[90] O. Martin, I. Kotsia, B. Macq, and I. Pitas, "The enterface'05 audio-visual emotion database," in *22nd International Conference on Data Engineering Workshops (ICDEW'06)*, pp. 8–8, IEEE, 2006.

[91] A. Batliner, C. Hacker, S. Steidl, E. Nöth, S. D'Arcy, M. J. Russell, and M. Wong, "" you stupid tin box"-children interacting with the aibo robot: A cross-linguistic emotional speech corpus," 2004.

[92] G. McKeown, M. F. Valstar, R. Cowie, and M. Pantic, "The semaine corpus of emotionally coloured character interactions," in *2010 IEEE International Conference on Multimedia and Expo*, pp. 1079–1084, IEEE, 2010.

[93] F. Ringeval, A. Sonderegger, J. Sauer, and D. Lalanne, "Introducing the recola multimodal corpus of remote collaborative and affective interactions," in *2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG)*, pp. 1–8, IEEE, 2013.

[94] R. Cowie, E. Douglas-Cowie, S. Savvidou*, E. McMahon, M. Sawey, and M. Schröder, "'feeltrace': An instrument for recording perceived emotion in real time," in *ISCA tutorial and research workshop (ITRW) on speech and emotion*, 2000.

[95] E. Paul, Keltner, and Dacher, "Universal facial expressions of emotion," *Segerstrale U, P. Molnar P, eds. Nonverbal communication: Where nature meets culture*, vol. 27, p. 46, 1997.

[96] P. Khorrami, T. Le Paine, K. Brady, C. Dagli, and T. S. Huang, "How deep neural networks can improve emotion recognition on video data," in *2016 IEEE international conference on image processing (ICIP)*, pp. 619–623, IEEE, 2016.

[97] F. Ringeval, B. Schuller, M. Valstar, S. Jaiswal, E. Marchi, D. Lalanne, R. Cowie, and M. Pantic, "Av+ ec 2015: The first affect recognition challenge bridging across audio, video, and physiological data," in *Proceedings of the 5th international workshop on audio/visual emotion challenge*, pp. 3–8, 2015.

[98] M. Hu, Q. Chu, X. Wang, L. He, and F. Ren, "A two-stage spatiotemporal attention convolution network for continuous dimensional emotion recognition from facial video," *IEEE Signal Processing Letters*, vol. 28, pp. 698–702, 2021.

[99] J. Kossaifi, G. Tzimiropoulos, S. Todorovic, and M. Pantic, "Afew-va database for valence and arousal estimation in-the-wild," *Image and Vision Computing*, vol. 65, pp. 23–36, 2017.

[100] A. Dhall, O. Ramana Murthy, R. Goecke, J. Joshi, and T. Gedeon, "Video and image based emotion recognition challenges in the wild: Emotiw 2015," in *Proceedings of the 2015 ACM on international conference on multimodal interaction*, pp. 423–426, 2015.

[101] S. Ebrahimi Kahou, V. Michalski, K. Konda, R. Memisevic, and C. Pal, "Recurrent neural networks for emotion recognition in video," in *Proceedings of the 2015 ACM on international conference on multimodal interaction*, pp. 467–474, 2015.

[102] C. Li, Y. Shi, and X. Yi, "Video emotion recognition based on convolutional neural networks," in *Journal of Physics: Conference Series*, vol. 1, p. 1738, IOP Publishing, 2021.

[103] S. Zhalehpour, O. Onder, Z. Akhtar, and C. E. Erdem, "Baum-1: A spontaneous audio-visual face database of affective and mental

states," *IEEE Transactions on Affective Computing*, vol. 8, no. 3, pp. 300–313, 2016.

[104] S. E. Kahou, X. Bouthillier, P. Lamblin, C. Gulcehre, V. Michalski, K. Konda, S. Jean, P. Froumenty, Y. Dauphin, N. Boulanger-Lewandowski, *et al.*, "Emonets: Multimodal deep learning approaches for emotion recognition in video," *Journal on Multimodal User Interfaces*, vol. 10, no. 2, pp. 99–111, 2016.

[105] I. Korshunova, W. Shi, J. Dambre, and L. Theis, "Fast face-swap using convolutional neural networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 3677–3685, 2017.

[106] Y. Zhang, L. Zheng, and V. L. Thing, "Automated face swapping and its detection," in *2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP)*, pp. 15–19, IEEE, 2017.

[107] X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent head poses," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8261–8265, IEEE, 2019.

[108] Y. Li, M.-C. Chang, and S. Lyu, "In ictu oculi: Exposing ai created fake videos by detecting eye blinking," in *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–7, IEEE, 2018.

[109] T. Baltrušaitis, P. Robinson, and L.-P. Morency, "Openface: an open source facial behavior analysis toolkit," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1–10, IEEE, 2016.

[110] B. Amos, B. Ludwiczuk, M. Satyanarayanan, *et al.*, "Openface: A general-purpose face recognition library with mobile applications," *CMU School of Computer Science*, vol. 6, no. 2, 2016.

[111] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L.-P. Morency, "Openface 2.0: Facial behavior analysis toolkit," in *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pp. 59–66, IEEE, 2018.

[112] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "Faceforensics++: Learning to detect manipulated facial images," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–11, 2019.

[113] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. Canton Ferrer, "The deepfake detection challenge dataset," *arXiv e-prints*, pp. arXiv–2006, 2020.

[114] H. R. Hasan and K. Salah, "Combating deepfake videos using blockchain and smart contracts," *Ieee Access*, vol. 7, pp. 41596–41606, 2019.

[115] F. Eyben, M. Wöllmer, and B. Schuller, "Opensmile: the munich versatile and fast open-source audio feature extractor," in *Proceedings of the 18th ACM international conference on Multimedia*, pp. 1459–1462, 2010.

[116] V. Bazarevsky, Y. Kartynnik, A. Vakunov, K. Raveendran, and M. Grundmann, "Blazeface: Sub-millisecond neural face detection on mobile gpus," *arXiv preprint arXiv:1907.05047*, 2019.

[117] S. Pandala, "lazypredict." `https://github.com/shankarpandala/lazypredict`. Version used: 0.22.

[118] P. Wang, "Audeering." `https://www.audeering.com/`. Accessed: 2021-09-07.

[119] B. Schuller, S. Steidl, A. Batliner, J. Hirschberg, J. K. Burgoon, A. Baird, A. Elkins, Y. Zhang, E. Coutinho, K. Evanini, *et al.*, "The interspeech 2016 computational paralinguistics challenge: Deception, sincerity & native language," in *17TH Annual Conference of the International Speech Communication Association (Interspeech 2016), Vols 1-5*, pp. 2001–2005, 2016.

[120] D. S. Brar, A. Kumar, Pallavi, U. Mittal, and P. Rana, "Face detection for real world application," in *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)*, pp. 239–242, 2021.

[121] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, pp. 3146–3154, 2017.

[122] R. Picard, "Affective computing-mit media laboratory perceptual computing section," *Technical Report no. 321*, 1995.

[123] C. Busso, Z. Deng, S. Yildirim, M. Bulut, C. M. Lee, A. Kazemzadeh, S. Lee, U. Neumann, and S. Narayanan, "Analysis of emotion recognition using facial expressions, speech and multimodal information," in *Proceedings of the 6th international conference on Multimodal interfaces*, pp. 205–211, 2004.

[124] C. Busso, S. Lee, and S. S. Narayanan, "Using neutral speech models for emotional speech analysis," in *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[125] C. Busso and S. S. Narayanan, "Interrelation between speech and facial gestures in emotional utterances: a single subject study," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2331–2347, 2007.

[126] C. Busso and S. Narayanan, "Interplay between linguistic and affective goals in facial expression during emotional utterances," in *7th International Seminar on Speech Production (ISSP 2006)*, pp. 549–556, 2006.

[127] S. Tripathi, S. Tripathi, and H. Beigi, "Multi-modal emotion recognition on iemocap dataset using deep learning," *arXiv preprint arXiv:1804.05788*, 2018.

[128] P. Ekman, "Facial expression and emotion.," *American psychologist*, vol. 48, no. 4, p. 384, 1993.

[129] M. Sidorov, S. Ultes, and A. Schmitt, "Emotions are a personal thing: Towards speaker-adaptive emotion recognition," in *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 4803–4807, IEEE, 2014.

[130] C. Marzban, "The roc curve and the area under it as performance measures," *Weather and Forecasting*, vol. 19, no. 6, pp. 1106–1114, 2004.

[131] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.