Emotion recognition from biosignals: an analysis of state of the art techniques in EEG, ECG, GSR and speech

Alessandro Calò, Edoardo Procino
Department of Information Technology and Electrical Engineering
University of Oulu
acal22@student.oulu.fi, eprocino22@student.oulu.fi

Abstract

In this report we will explore some state-of-the-art papers in the field of emotion recognition from biosignals. Our aim is to analyze and understand the newest techniques used in the emotion recognition from biosignals, consequently explaining how these concepts can be applied to the everyday life – they can be useful for people –. Specifically, in this paper we will write about three different biosignals and about their related state-of-the-art techniques or applications. In the end, we propose a model of our creation, with the main purpose of exploring this field with a hands-on approach and learning more about the topic. The topics that we will touch are:

- EEG Emotion Recognition using dynamical graph convolutional networks;
- Deepfake speech detection through emotion recognition;
- Emotion Recognition using a combination of biosignals;
- Our experiment on emotion recognition trough EEG analysis.

Contents

1 Introduction to our work

For the course *Affective computing* at the university of Oulu we were supposed to produce a summary about some papers related to the subject. In particular, we chose the topic *emotion recognition from biosignals*, as we were captivated by the depth of the topic and all the possible applications. First of all, the concept of extracting and processing emotions from biosignals is a very fascinating one by itself, since it actively links two worlds that seem apparently far apart: the one of machines and the one of emotions. As mentioned before, it is a huge field with incredible depth and this feature allowed us to study and analyze several topics that we thought would be interesting to explore further.

The report is then organized in four main chapters, in the first one we will talk about the emotion theory and the emotion recognition, with the purpose of introducing the topic. In the following sections, we chose to explore three different topics related to the main one, which are:

- EEG Emotion Recognition using dynamical graph convolutional networks (§ 2);
- Deepfake speech detection through emotion recognition (§ 3);
- Emotion Recognition using a combination of biosignals (§ 4)
- Our experiment on emotion recognition trough EEG analysis (§ 5). .

1.1 Emotions - An Overview

In this section we will talk a little bit about emotion theory and about Emotion recognitions to introduce the topic.

An agreed-upon definition of emotion says that it is a response to an event, which is generally consistent and discrete. Moreover, it can be said that it is a psychological state and it involves an experience which is subjective, a psychological feedback and a behavioral expressive feedback.

In the field of emotion recognition there are two ways of representing emotions:

- Categorical: emotions are divided in fear, anger, surprise, sadness, disgust, curiosity, acceptance and joy;
- Dimensional: emotions are mapped in three dimensions, which are Valence, Arousal and Dominance (VAD). Valence represents the spectrum of feelings from very negative to very positive. Arousal determines how sleepy or excited the subject is. Dominance reflects how strong a felling is in that moment.

1.2 Introduction to EEG in emotion recognition

This research topic has seen much attention being drawn to the study of EEG signals, since they seem to be strongly linked to emotions. Thus, steps were developed to be performed in order to correctly recognize emotions:

Starting from the user, he/she must be exposed to a stimulus while the voltage changes in the brain are recorded, the noise is removed as much as possible. This method involves first examining the data obtained, and then extracting the relevant features. After that, a classifier is trained based on a training set using the extracted features, leading to the explanation of the original signal.

As for the actual classification of emotions, many studies were focused on using Machine Learning algorithms rather than applying Deep Learning methods. Specifically, Support Vector Machines were used in almost 60% of the studies ranging from 2009 to 2016, followed by k-Nearest Neighbours, then Naive Bayes and Multi-Layer Perceptron.

2 EEG Emotion Recognition using dynamical graph convolutional networks

Our review will be focused on Deep Learning methods rather than Machine Learning.

2.1 Dynamical graph convolutional networks approach

In this paper [8], the researches developed a new kind of neural network – called Dynamical Graph Convolutional Neural Network – to achieve a new state of the art (sota) score in EEG emotion recognition.

2.1.1 Graph Neural Network (GNN)

Convolutional Neural Networks (CNN) are very good in the classification task especially with image, video and speech inputs, while they do not perform very well in feature learning problems such as features learning. In these cases graph based methods are the better choice as they use graph theory to work with the data.

From this kind of networks, Graph Convolutional Neural Network (GCNN) borned as an improvement of classic CNN; this new method works better than classical CNN in discriminative feature extraction of signals since they can describe relationship between different nodes in the graph.

The basic idea of the paper under analysis is to use this graph representation to encode the channels of the EEG, but, the problem is being able to predetermine the functional relationship among the EEG channels, thus, the edges between the nodes. To go beyond these limitations, the researchers of this paper proposed a new approach: a Dynamical Graph Convolutional Neural Network (DGCNN). This new method is supposed to learn the adjacent matrix in a dinamical way during the training – instead of requiring that matrix before the model training –.

2.1.2 From GCNN to DGCNN

To well understand this new kind of networks, as the researchers of this paper did, we will go through the graph and spectral graph filtering theory.

Graph representation A directed and connected graph is a tuple of 3 elements: $\{V, \varepsilon, W\}$ where:

- V is the set of nodes with cardinality N;
- ε is the set of edges between nodes;
- $W \in \mathbb{R}^{NxN}$ is a metrix which elements are defined as:

  $$w_{ij} = \begin{cases} exp(-\frac{[dist(i,j)]^2}{2\theta^2}) & \text{if } dist(i,j) \leq \tau \\ 0 & \text{else} \end{cases}$$

  Where τ and θ are fixed parameters and $dist(i,j)$ is the distance between the ith node and the jth node.

Spectral Graph Filtering Spectral graph filtering is a field of mathematics which focuses on the properties of matrices associated with graphs, such as their adjacency matrix or Laplacian matrix, and their characteristic polynomials, eigenvalues, and eigenvectors. [19]. The researchers used this method to try to learn the optimal adjacency matrix using the *K order Chebyshev polynomials* to replace the polynomial expansion of a term in order to seed up and to semplify the calculatios.

2.1.3 The model

Based on that, they built the model that has, as shown in figure 1, the following layers:

- The dynamical graph convolution layer;
- A 1x1 convolutional layer which should learn the discriminative features among the frequency domains;
- A Relu activation layer to have both a not linear mapping and non-negative outputs;

- Some fully connected layers; the last one has a softmax activation function to give in output a probability distribution among all the class labels.

The input of the network is composed by the EEG features extracted from five frequency bands ($\delta, \theta, \alpha, \beta$ and γ).

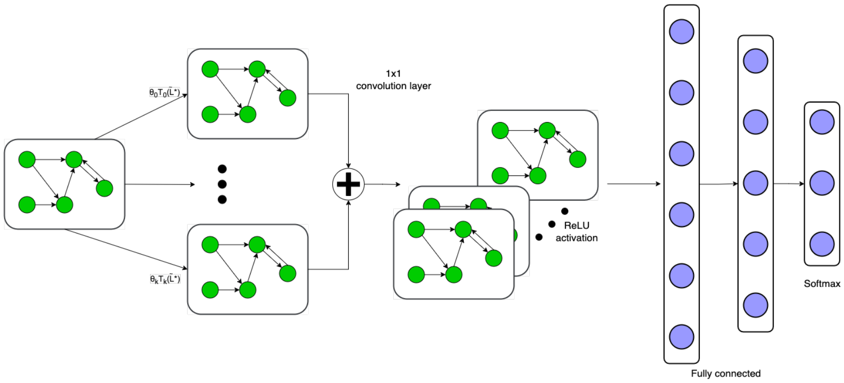

Figure 1: Architecture of the model

Regarding the optimization of the network parameters, they used the back propagation (PB) algorithm with ad hoc loss function based on *cross entropy* [4].

2.2 Experiments

They conduct experiment on two dataset: the DREAMER dataset [7] and the SEED dataset [18]. These are common datasets to work on in this research field [10]

2.2.1 Experiment on SEED dataset

This dataset contains the EEG data of 15 subjects collected with 62 electrodes while subjets were watching film clips with three kind of emotions: negative, positive and neutral. In the dataset there are 45 trials of EEG data for each subject.

On this dataset, they did two experiments:

Subject-Dependent For all the 15 trials of EEG data associated to one session of one subject (43.5 percent of studies analyzed in [10] use user-dependent data), they used the first 9 as training set and the other one as test set and then they computed the accuracy on each subject and the mean of them all. They also tested 5 features (extracted from the frequency bands). The results of this experiment are here summarized:

- The Differential asymmetry feature (DE) was the better one on the accuracy;
- In all tested cases the best accuracy was achieved when all the frequency bands were used together;
- DGCNN and GCNN are better that the other two methods used for comparison – Support Vector Machines (SVM) and Deep Belief Networks (DBN) –, in general DGCNN demontrated to be better then classic GCNN in this task.

The best result achieved is 90.4% accuracy which was achieved using the DGCNN network, the DE feature and all the frequency bands.

Subject-Independent In contrast to the previous experiment, here they used the leave-one-subject-out (LOSO) cross-validation strategy, in particular they used 14 subjects for the training and one for the test, they repeated the experiment such that each subject was used once as testing data and finally they computed the mean accuracy using the accuracies of all the tests.

The results confirm the above explained experiment, infact the best recognition accuracy was reached by the DGCNN model combining the feature extracted from the five frequency bands together, the DE feature also in this case was the better one, but in this experiment the accuracy dropped to 79.95%.

2.2.2 Experiment on DREAMER dataset

The Dreamer dataset is organized as follows:

- It has data on both electrocardiogram (ECG) and electroencephalogram (EEG), which is the signal on which we focused on, not considering the ECG;
- They recorded them during affect elicitation by means of audio-visual stimuli;
- The partecipants was 23 and they were supposed to go trough 18 video clips;
- They used 14 electrodes to register the EEG;
- In the dataset there is also the baseline for each registered signal;
- The label for each data is a vector with 3 numbers –from 1 up to 5– which represent *valence*, *arousal* and *dominance*.

To deal with the dimensional representation of emotion used in this dataset, the researchers transformed them in binary states using the labels ’low’ and ’high’.

For this experiment they manipulated the data to obtain 14 features, one for each channel of the EEG, then they concatenated that features to obtain a 14-dimensional vector that represents a single EEG data sample. Only the subject dependent experiment was performed.

They compared the DGCNN with SVM, FraphSLDA and GSCCA. The better accuracy for all Valence, Arousal and Dominance was reached by the DGCNN network, even though the accuracy scores are slightly less than the one obtained with the seed dataset, infact the network reached the following accuracies:

- Valence: 86.23%
- Arousal: 84.54%
- Dominance: 85.02%

2.2.3 Results

The results are very clear, the DGCNN model is a way better than the other state of the art methods, the same thing can be said comparing them to the results found in [10]. The proposed model performs like that probably due to the use of a nonlinear neural network, the fact that the graph representation can characterize the intrinsic relationships between the channels of the EEG and of course for the optimization of the adjacency matrix.

3 Deepfake speech detection through emotion recognition

3.1 What is a deepfake

A deepfake is a generated media – like images, videos or audios – that represents a real person. In particular, the target person face and/or voice replace an existing one in a pre-existing media.

Nowadays, deepfakes, due to their high quality, can represent a problem for the society since they can be used to represent people in compromising behaviors. An example is the work done by Jordan Peele in 2018 [13]. He made a realistic video representing the ex president Barack Obama only to show the power of AI techniques combined with Adobe After Effects. Obviously videos like this one can be very dangerous if used with bad intentions, for this reason researches are studying ways to detect deepfakes. An example of that is the work done by E. Conti et al. [3] where they tryes to detect video deepfakes throgh state-of-the-art speech emotion recognition (SER). The point of their research is that audio deepfake techniques cannot correctly sintesize natural emotional behavior. In the following sections we will discuss the work of E. Conti et al. comparing it with another approach discussed in a paper done by a subset of the authors.

3.2 The Model

The proposed approach is based on transfer-learning: they used semantic features extracted from a SER network as input of a deepfake classifier. The work focused on text-to-speech (TTS) and mixture TTS/Voice Conversion (VC) deepfakes because pure VC deepfakes do not have speech semantic information that are used for the detection.

As stated before, the aim of the paper is to detect if a speech recording belongs to a real person or not. So the goal is map a speech audio signal called x to its class y: $x \to y \in \{REAL, DEEPFAKE\}$.

As shown in figure 3, the precess is divided in two blocks: the first one is the network proposed in [2], starting from the input x, it detects the associated emotion E_x and extracts some features F_x (see the following section for details), while, in the second block, the Synthetic Spech Detector (SSD), associates a class – real or deepfake – to F_x.

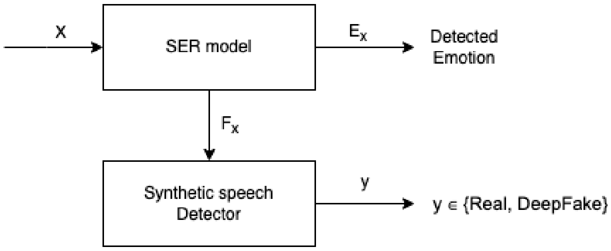

Figure 2: Scheme of the architecture proposed by E. Conti et al.

3.2.1 The network: 3-D Convolutional Recurrent Neural Networks With Attention Model

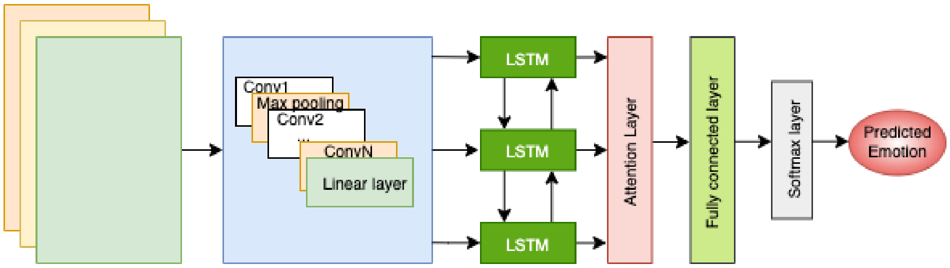

Figure 3: Architecture of the model

The network used by [2] is quite complex and works in 4 steps:

Preprocessing Startig from the initial signals the differences between the different speakers are lowered computing the zero mean and unit variance of the signal, then using an Hamming windows the signal is splitted in frames. After that, the log-Mels m_i is calculated in the following way:

- They calculate the power spectrum of of each frame using discrete Fourier transformation;
- The power spectum is passed through the Mel-filter bank i to produce P_i;
- $m_i = log(P_i)$

At this point, in order to obtain a 3-D feature map they calculated the delta features (Δ) and the delta-delta features ($\Delta\Delta$). Generally speaking, for a feature f_k at time instant k, its delta is defined as $\Delta_k = f_k - f_{k-1}$ while its delta-delta is defined as $\Delta\Delta_k = \Delta_k - \Delta_{k-1}$ [5].

The final preprocessed input for the network is the composition of m_i, Δ and $\Delta\Delta$ as 3-D feature representation.

CRNN model The second step consists in feed the 3-D input to a convolutional recurrent neural networrk (CRNN) composed with 3-D convolution layers, one 3-D max-pooling, one linear layer and one Long Short Term Memory (LSTM) layer for temporal summarization –LSTMs have feedback connections so they can process not only single data points (such as images), but also entire sequences of data (such as speech or video) [14]–.

Attention layer After the high feature extraction performed by the first module, an attention layers is used to focus on emotion relevant parts. An attention layer in a particular kind of layer that can be used to highlights some parts of the data at the expense of other parts.

Fully connected layer and softmax classification Finally, a fully connected layer is used to obtain higher level representations that help the softmax classifier which produce a probability distibution on N classes from which we can extract the prediction E_x, in this case N represents the number of recognizable emotions.

3.2.2 The syntetic speech detector

To extract F_x from the above described network a transfer-learning approach was used. In particular F_x is the output of the attention layer.

This vector has discriminative power for synthetic speech detection so we can use it as input for a binary classifier – a decision tree forest was used – which takes in input F_x and produces in output $y \in \{REAL, DEEPFAKE\}$, in particular, they used a Random Forest Classifier – the best found hyperparameters are *NumberOfLearners=300* and *information gain* as quality criterion function –.

3.3 The used data

E. Conti et al. used 5 different datasets to train the SER block and to train and test the deepfake detector. This choice was made to avoid overfitting and to make to make the proposed techniques suitable for real world conditions.

The used datasets are:

- *ASVspoof2019* which contains both real and deepfake data, it was created to develop antispoofing techniques for automatic speaker verification;
- LibriSpeech that contains 1000 hours of authentic speech (they consider only the subset called *train-clear-100*);
- *LJSpeech* that contains audioclip from a single person;
- *Cloud2019* which contains tracks from different TTS cloud services (as Amazon AWS Polly and Google Cloud Standard);
- *Interactive Emotional Dyadic Motion Capture* which contains video and audio annotated with speaker's emotions, this is the only one dataset used to train the SER block

3.3.1 Input preprocessing

In order to avoid dataset-specific results, they pre-preocess all the tracks to make them more uniform, the steps was:

- All the tracks were converted to mono;
- They were downsapled to $F_s = 16kHz$;
- The tracks were filtered using a Butterworth band-pass digital filter (the Butterworth filter is a type of signal processing filter designed to have a frequency response that is as flat as possible in the passband [1]) with order 6, considering a lowcut frequency $F_l = 250Hz$ and a highcut frequency $F_h = 3600Hz$;
- Each track was normalized using the infinity norm;
- Each track was reduced to have a common length $L_{cut} = 3s$ (using 0 padding if necessary);
- The STFT was computed using an hamming window of length $L_w = 0.025s$ and a hopsize $L_h = 0.01s$;
- The spectrum and the deltas were calculated.

3.3.2 Dataset augmentation

In order to test the system against audio degradation, they created a second version of the final dataset. They did that adding white noise to the speech tracks following two approaches:

1. For the train and validation sets they injected noise according to a double-layer probability distribution, in particular the first layer injected white noise between 30 and 15 dB with probability 0.8, while the second one injected noise between 15 and 10 dB with probability 0.3;
2. For the test set, the power of the noise was fixed to [25, 20, 15, 10] dB.

3.4 Results

To test the model the researchers did two tests, the former using the clear dataset while the latter using the dataset augmented with the noise injection.

Experiment with the model trained on the clear dataset In this first experiment, we showed that the proposed approach reaches higher discrimination performance than classic CNN methods, and that training the used network directly for SSD leads to worse performance than training it for SER and then using it as feature extractor for the SSD task. This classifier, trained and tested on the dataset without noise injection, performs a way better than the 3 models – well-established state of the art methods – used as comparison since it reaches AUC (Area under the ROC Curve) = 0.98, while the others reach AUCs between 0.86 and 0.89. On the other hand, as the level of noise in the test set increases, the performance degradates more and more tending to label all samples as authentic (increasing in false-negative rates).

Experiment with the model trained on the noised dataset The results show that the system trained on clean data is better than the one trained also with noised data, but the latter is a way better than the former on classifing noised data preserving very good performances on the clear data. Infact, analysing the test set with 10dB noise, the second model reaches (almost always) a balanced accuracy greater than 0.80 with both real and deepfeak videos, while the model trained only with the clear data, always on the 10dB test set, reaches a balanced accuracy greater than 0.93 for the real videos, but less than 0.36 for deepfake videos.

3.5 A comparison with a speech and video architecture

To explore a little bit over our topic and to see how biosignals emotion analysis can be integrated also with other techniques, we decided to write about this [12] paper which uses emotion analysis on both video and audio to assert if a video is a deepfake or not.

In this paper the researchers proposed a hybrid approach assuming that deepfake videos of human speaker display inconsistencies between the emotion in the face and the one in the voice. Thus, as can be seen in figure 4, the aim of the paper was to detect the above mentioned inconsistencies using the valence-arousal model of emotion. To detect deepfakes, audio and facial Low Level Descriptors (LLDs) are first used to estimate how valence and arousal change over time, then this estimation is fed to a supervised classifier.

As we will see better later, the obtained results show that deepfake speech generation methods are not able to well synthesize natural emotions.

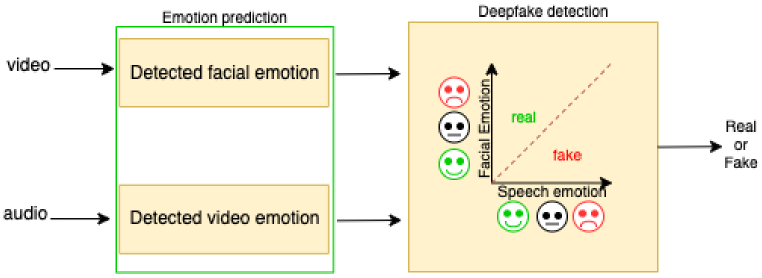

Figure 4: Representation of the proposed system

3.5.1 The method

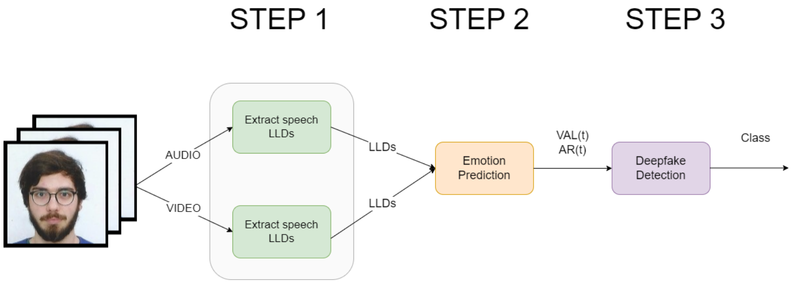

Figure 5: Representation of the proposed system

As image 5 resumes, the proposed method works in 3 stages:

Low-Level feature extraction In this first step, the system extracts LLDs that describe the face and the voice.

- **Facial features**: To extract the LLDs for the facial part, the researchers of this paper used the *Facial Action Coding System (FACS)* which is a system to taxonomize human facial movements by their appearance on the face [11];
- **Speech features**: To extract features from the audio, the researchers adopted a combination of three feature sets described on OpenSmile (MFCC, GeMAPS and eGeMAPS) [17]. They did not use all the features, but only subsets of them.

Emotion recognition Using the extracted LLDs sequences as input, the second stage is in charge of estimate the speaker's emotion over time both from audio and video source. To represent the evolution of the motions over time, the researchers used an LSTM neural network trained on the SEMAINE dataset [16] which outputs for time-series:

- $VAL_s(t)$ and $AR_s(t)$ are the speech based valence and arousal time-series;
- $VAL_f(t)$ and $AR_f(t)$ are the video based valence and arousal time-series.

Deepfake detection The last stage is the deepfake detector, trainend on the Deepfake Detection Challenge training dataset [6], which uses the time-series produced by the Emotion recognition step to assert if a given video is fake or not. To do so, two different approaches were used:

- **Statistical Features approach**: They trained a lot of different classifiers – Random forest (RF), XGBoost (XGB), Logistic Regression (LR) and K-Nearest Neighbors (k-NN)– trained on 10 statistical features including the Lin's Concordance Correlation Coefficient and the mean of speech and video arousal;
- **Learned Features approach**: Since temporal evolution of valence and arousal could be very important to detect deepfakes, also a LSTM classifier with VAL_s, VAL_f, AR_s and AR_f was used.

Both approaches were trained on only audio features, only video features and on both video and audio features.

3.5.2 Results

The obtained results are very good. As reported in table 1, the results highlight that:

- The audio features are more useful than video ones to discriminate between real and fake videos;
- Audio plus video features are not always better than only audio features;
- LSTM model is a way better than the other models, especially with audio and video features together it reaches 99.5% balanced accuracy.

The results also highlights the fact that the LSTM model is the only model that performs very well on only facial features supporting the hypothesis that deepfakes are not good at creating semantic consistencies and also that synthetic speech does not achieve the same emotional range as the real one.

Model	Scenario	Balanced accuracy	AUC	TPR@5%
Stat - RF	Audio	87.1%	0.937	83.2%
	Video	50.2%	0.509%	12.0%
	Audio + Video	84.9%	0.945	84.9%
Stat - XGB	Audio	87.8%	0.944	94.7%
	Video	51.1%	0.519%	51.9%
	Audio + Video	87.4%	0.947	89.4%
Stat - LR	Audio	84.7%	0.930	81.8%
	Video	50.4%	0.508%	13.2%
	Audio + Video	85.3%	0.933	82.7%
Stat - k-NN	Audio	84.5%	0.882	86.3%
	Video	51.8%	0.507%	16.8%
	Audio + Video	80.1%	0.921	91.3%
LSTM	Audio	98.9%	1.000	100%
	Video	95.7%	0.973%	94.3%
	Audio + Video	99.5%	1.000	100%

Table 1: Deepfake detection performances: accuracy, AUC and detection rate at 5% false alarm rate

3.6 Conclusion

Not considering the performances achieved by the two examined methods – which are both very good and dataset depending –, it is clear that the speech features have a very high discriminative power on the examined task. In [3] only with the speech the researchers were able to reach incredible results also in context with noise, while [12] underlines the fact that the speech is a way more important than video in detect deepfakes – although combining the two is the solution that gets the best results –.

4 Emotion Recognition using EEG, ECG and face images

4.1 Emotion Classification using a self-supervised multi-task CNN for ECG signals

The paper was captivating because it utilized self-supervised learning for the first time to recognize emotions using ECGs.

4.1.1 ECG for Affective Computing

Electrocardiograms (ECGs) depict the electrical activity of the heart as measured by electrodes attached to the body's surface. An ECG signal is made of different waves: the P wave, the QRS complex, the T wave, and often the U wave. These wave-forms are used to understand the cardiac state of an individual, as the they contain significant information for that task. It is important to note that this kind of signal has been proved to be strongly correlated with emotional states and attributes. These signals can be acquired raw in order to obtain high-level representations, or many feature extraction techniques can be utilized to analyze them.

4.1.2 Self-Supervised Representation Learning for ECG

In self-supervised learning, the structure of unlabeled data is exploited to create learning problems that can be solved with supervised techniques. In this work in particular, this paradigm consists of training a network using labels which are generated automatically, instead of human-annotated ones. Indeed, a big problem of fully-supervised methods is that they usually require a large quantity of human-annotated labels: in this multi-task self-supervised representation learning method, there is no need for that (this way larger datasets can be utilized).

4.1.3 Data and Preprocessing

Data The datasets used for this work are all available to the public:

- AMIGOS contains data collected from 40 partakers to study mood, personality, and emotional responses, while iteracting with multimedia content both being alone or in group
- DREAMER: to elicit emotional responses, 23 participants watched video clips and were given audio and video stimuli during the study (see 2.2.2 for details about the dataset's structure)
- WESAD contains ECG data of 17 different subjects, regarding four different emotional states namely amused, neutral, meditated, stressed.
- SWELL: the ECG data was obtained from 25 participants, with the goal of understanding the mental stress and emotional attributes of employees in a typical office environment.

Preprocessing The amount of pre-processing that was done to the signals is very minimal, with the justification being that the authors wanted to better understand the impact of the proposed model on learning important ECG representations based on almost raw input.

SWELL and WESAD ECG signals are downsampled to 256 Hz to be consistent with AMIGOS and DREAMER, and after that a high-pass filter (0.8 Hz) is applied to remove the baseline wander. The last step was to perform z-score normalization.

4.1.4 Model

This work uses a transfer learning technique to discriminate between emotions trough a the classification of valence, arousal, stress and, emotional states. This is achieved by using two networks, one to learn the ECG representations and one for the emotion classification task.

The first network is trained on ECG signals, in order to make it capable of extracting the ECG representations (pretext set): its meaning is to learn generalized features about the ECG data which is automatically labeled. After this first step, the weights are then frozen and transferred to the network whose goal is to perform the discrimination task.

The two CNNs (Figure 6) are almost identical, having 3 convolutional blocks, and 7 branches, with 2 dense layers. After the last convolutional block, there is a max-pooling layer. The difference is that the first network has an output which is fed to 2 parallel fully connected layers (60% dropout to overcome overfitting) and finally to a sigmoid layer, while the other network has one block of fully connected layers (512 nodes). The weights from the first newtork's shared layers are frozen and transferred to the second one. As a result, the emotion recognition network's fully connected layers are trained on data labelled with emotions.

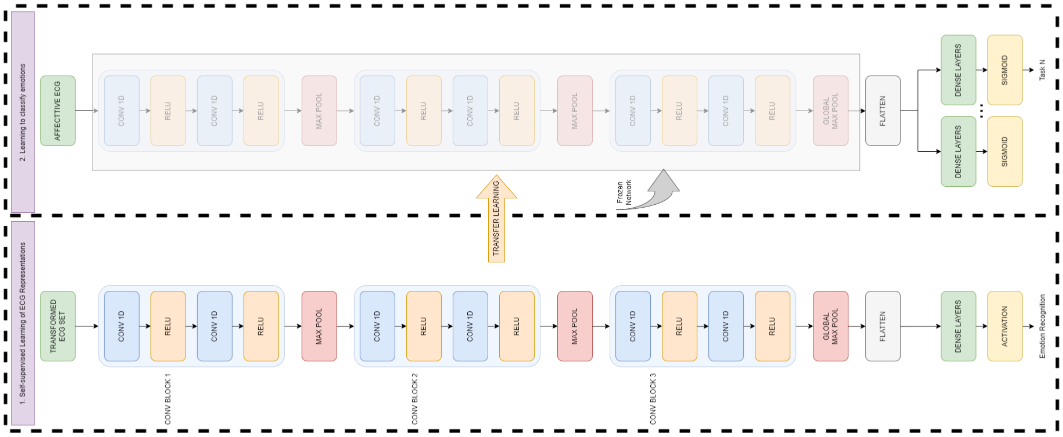

Figure 6: Transfer Learning Networks Overview

ECG Representation Network This network deserves some further analysis: for the purpose of learning the representation of ECG, this network is a self-supervised deep-multitask CNN, which goes trough a process that involves the recognition of different transformations which are applied to the raw ECG signals (no human-labeled data). We will list the six transformations below, without going in depth into them as this is not the purpose of this project:

- Noise addition
- Scaling
- Negation
- Temporal Inversion
- Permutation
- Time-warping

These modifications are applied to the signals, then a matrix is created with each transformed signal on top of each other, and the otput (labels) being the transformation functions (as a number from 1 to 6, 0 being original signal).It is then necessary to shuffle the input-output matrices in order to reorder the modifications and their associated outputs.

4.1.5 Model implementation and Training

For the training phase of the model, a 10-fold cross validation was chosen, which is a good trade-off according to already existing literature. Generally, the training on the SWELL dataset was generalluy better than the training on AMIGOS and DREAMER datasets. The Adam optimizer was used (lr = 0.001), with 100 epoch for the first network and 250 for the second.

During the training, both pretext tasks (ECG Representation) and emotion recognition tasks trained well and reached steady states using this approach.

4.1.6 Results

Table 2 shows how the emotion recognition network is able to categorize emotions using the labels that are present in each dataset. These results in set a new state-of-the-art in 2020 as the afore discussed model perfomed better than the ones that preceded it.

The self-supervised solution has the advantage of allowing an aggregate of the existing datasets to be used in training the self-supervised network, thus removing the main barrier of having different output labels across datasets. This aspect was analyzed by the authors by trying to train and test the model with all the datasets at the same time or with one at a time. With the first option, the model is able to learn more generalized features, resulting in better performances compared to the single dataset method.

4.1.7 Conclusions

In this paper it is shown how the self-supervised method actually improves the performances w.r.t the fully-supervised one, and also how a multi-task approach has better results when compared to a single-task network.

Table 2: The emotion categories, number of classes for each class, and multi-class emotion recognition results are presented for each of the four datasets

Dataset	Used attribute(s)	Number of classes	Accuracy	F1
AMIGOS	Arousal	9	0.796	0.777
	Valence	9	0.783	0.765
DREAMER	Arousal	5	0.771	0.740
	Valence	5	0.749	0.740
WESAD	Affect State	4	0.950	0.940
SWELL	Arousal	9	0.926	0.930
	Valence	9	0.938	0.943
	Stress	3	0.902	0.900

4.2 A multimodal approach to Emotion Recognition

In this paper the emotion recognition task is tackled from a different perspective: trying to utilize heterogeneous bio-signal data sources, namely EEG signals, eye data and face data.

4.2.1 Introduction

The new perspective also makes it possible to explore new ways to tackle heterogeneous data sources, in particular how to pre-process them and how to effectively select and use models for emotion recognition. This is done following the valence and arousal based description of emotions.

4.2.2 Data

The acquisition of the data was done manually by the team of researchers who wrote the paper, so no already existing dataset was used. The data utilized in this study are the following:

- **2-channel EEG signals** Brain activity signals obtained during a test phase, with two electrodes placed following the International 10-20 System of Electrode Placement
- **2D/3D face data** Two different kinds of face data were used:
  - Face depth: tracking the depth of facial points for a representation of the facial expressions, proved to be more consistent under different illumination conditions
  - Image data: the image of the face of the subject
- **2D eye data** As the subject watches stimuli on the screen, the pupil diameter and gaze position of both eyes are tracked.

Data acquisition The acquisition was done trough two phases, namely a video trial phase and a game trial phase. During the first, the subject watched fourteen 2-min long videos, and after the end of each one a completion of the SAM (Self-Assessment Manikin) questionnaire was required. In the second phase the subject played a video game (e.g. Pac-Man), followed by the SAM questionnaire.

Pre-processing and Data Augmentation As for the pre-processing steps, for the EEG signals only a band-pass filter was applied, reducing the frequency of the signal to the 4-40 HZ band. Noise reduction steps were done to the eye data, namely blink correction, saccade correction and pupil diameter fluctuations. The face data was subjected to noise correction and head translation normalization.

The data augmentation step was very important, as the collected data wasn't enough for a neural network to train on in an effective way (there being only 49 participants). In order to perform this step, a random value was generated from a Gaussian distributions with mean being equal to zero and a standard deviation of 3 (which was demonstrated to be the best value for this specific task),and an original data component was added by a percentage equal to this value.

4.2.3 Model

The proposed model is a multi-branch deep CNN which employs all three modalities of two channels of EEG signals, 2D/3D face data and 2D eye data, as it can be seen in Figure 7.

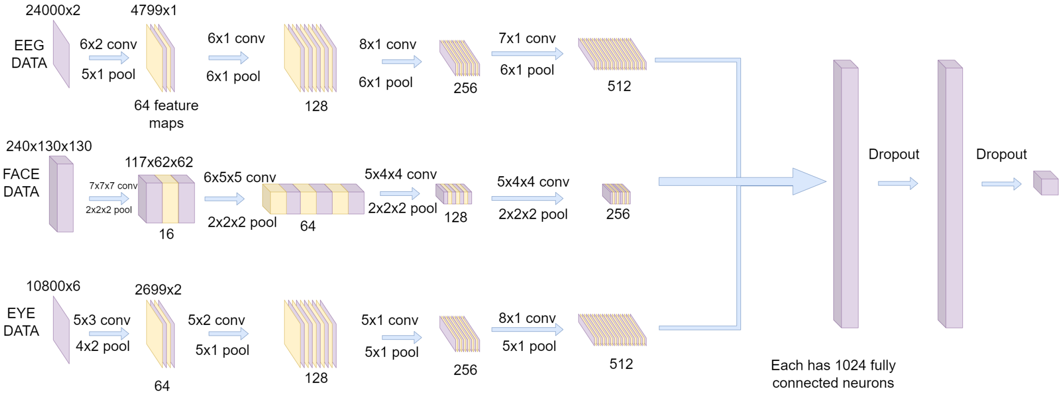

Figure 7: Multi-branch deep CNN

Of course, for each branch of the network, there is a different modality as input: the data of the different modalities are first processed by a different sequence of convolutional layers and max-pooling layers, which are organized into four consecutive batches. The final outputs of all batches for all modality data are then processed together by a sequence of three fully connected layers and two dropout layers (to reduce overfitting).

The final layer has one neuron and it outputs the probability of the high class (6 to 9) being the true class of the arousal or valence level of the input. The probability of low class (1 to 5) being the true class is considered to be 1 minus that of the high class.

4.2.4 Results

Arousal and valence were categorized into low-level (no more than 5) and high-level (no less than 6) levels in order to make the experimental results comparable with other state-of-the-art baselines.

The results of this system were better than some state-of-the-art results, being the accuracies as high as of 67.8% and 77.0% for valence recognition and arousal recognition, respectively.

An interesting discovery of this paper revolves around the use of the face depth data, which is an uncommon feature to introduce in a system. In particular, the focus here is the comparison with the use of normal face image data: Figure 8 shows how the performances obtained with the usage of face depth data are better than the case in which face image data is utilized. This discovery contributes to the whole emotion recognition research field, according to the authors. Also, using depth information to track a face rather than an image reduces the effects of illumination conditions as well as background objects, so it makes sense that the performances turn out to be better.

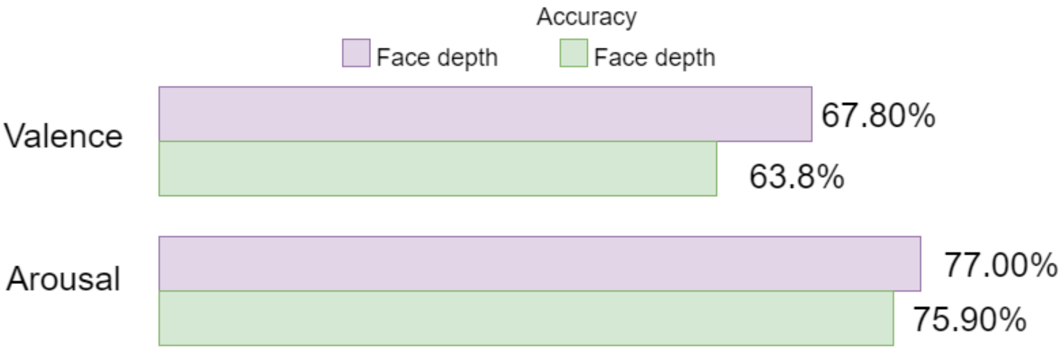

Figure 8: Comparison of results between face depth and face image

4.2.5 Conclusions

The authors note that some other systems which represent the state-of-the-art are able to reach slightly better results: this is because of the higher number of EEG channels used. They argue that the higher the number of channels, hence higher the number of electrodes used in the acquisition phase, the worse experience it is for the subject who is being tested. Hence, the more uncomfortable the participant gets, the more the emotions can be altered during the testing phase. This, along with economic factor, is the reason that led them to opt for a 2-channel EEG acquisition rather than a higher number.

Although more electrodes can cause discomfort, they understand the importance of better EEG signals, setting this as a goal for future works.

5 Our experiment on emotion recognition trough EEG analysis

Even if we are not supposed to produce code, since we are very interested to this subject, we decided to go ahead and try to implement something of our own. Our idea is to consider the DREAM dataset [7] – which contains EEG data – and use it to train and test a deep learning model, after pre-processing the signals.

5.1 The dataset

Inspired by [8], we decided to use the DREAMER dataset, its structure is explained in 2.2.2.

As described in the following section, we used the dataset in an "abstract" way: we used each piece of data – a single measurement from a single electrode – as if it was a stand-alone record.

5.2 Production of data

This of course was the most difficult part of the project, as it required us to study and understand better the data we were working with. It is important to note that this process required a fair share of trial and error: we tried many combinations of data, trying to understand which one was suitable for our goal, which was to reach at least a 60%/65% accuracy on the classification task.

The better idea we had consists in taking the raw signal, subtracting the baseline from it and compute its spectrogram. The main thing about this data is that it is a 14-channel EEG, meaning that during the acquisition part, 14 electrodes were placed on the participant's head: this resulted in 14 different wave signals.

Since every electrode is able to detect voltage changes in the brain during the stimulus, it is able to describe how the subject is reacting from a different perspective w.r.t. the other electrodes: we can consider the signal from every electrode as independent, and doing so we can multiply by a factor of 14 the usable data.

This concept may make one think of redundancy, but in this case it works almost as a data-augmentation step: for one test, every electrode picks up the same reaction of the subject (feelings) by a different perspective, but all 14 electrodes are actually describing the same event. So every one of the 14 signals is just a different version of the same subject's reaction, and they are treated as independent entries.

Following this idea, we now have a much bigger pool of data, as the number of participants is 23, each of them doing 18 trials with 14 electrodes: 23x18x14 = 5796 usable data samples. These samples are EEG signals, that were used to plot the spectrograms. In section 5.2.2 we will explain the exact procedure that we used.

5.2.1 Labels

About the labels we obviously used the same one for all the 14 "generated" data that originally belonged to the same registration. Also, following the work done by [8] and [9] we modified the labels making them vectors with three binary values which indicates if the corresponding data was low or high. In particular if a value was less than or equal to 2 we labelled it as 'low' while if it was greater than 2 we labelled it as 'high'. So the transformation is the following one: $\forall$ label l in the form $[n_1, n_2, n_3]$ with $n_1, n_2, n_3 \in \{1, 2, 3, 4, 5\}$ we compute a label $\hat{l}$ int he form $[b_1, b_2, b_3]$ where b_1, b_2, b_3 are binaly lables with 1 that means 'high' and 0 that means 'low' (i.e.: $[4, 3, 2] \rightarrow [1, 1, 0]$).

5.2.2 Spectrograms generations

This phase was performed in two steps:

Data extraction The datased (a .mat file) is organized in a lot of sections and so we needed to navigate it to find our data. In particular we extrapolated the *stimuli* field ($DREAMER \rightarrow Data \rightarrow EEG \rightarrow stimuli$) and the *baseline* field ($DREAMER \rightarrow Data \rightarrow EEG \rightarrow baseline$). After that, as suggested by our friend Francesca Mannini (francesca.mannini@mail.polimi.it) – who is studying biomedical engineering at the master and helped us in the more EEG related stuff – we built a matrix with the baseline subtracted from the stimuli such that each column is the signal from a single electrode – while the rows are the time unit of the sampling –.

we also extracted the labels, we pre-processed (§ 5.2.1) them and we putted them in a .csv file.

Spectogram generation At this point the spectrograms generation was very straightforward, we just passed each electrod data to the funcion *specgram* of matplotlib [15].

The resulting images are like the one in figure 9.

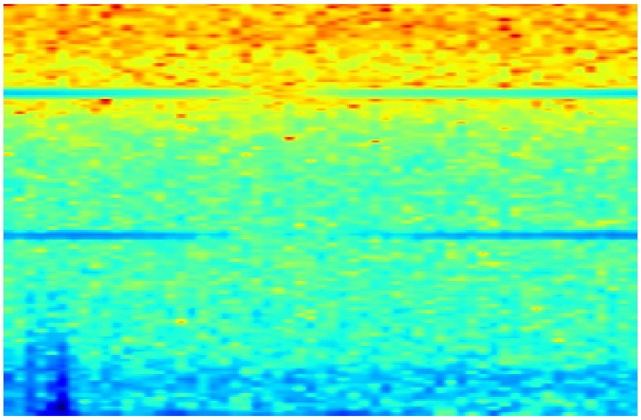

Figure 9: An example of a generated spectrogram

5.3 The model

The idea for the project was to use a pre-trained network and fine tune it in order to fit to our data. We tried many networks, but the one that was chosen in the end was VGG-16, as it performed better than the other ones.

We took inspiration for this method during one of the lectures about emotion recognition from speech.

VGG-16 Architecture This is a well known network used for image classification, it consists of 16 convolutional layers, as can be seen in Figure 10. The input of the network is our generated spectrogram, which has dimensions (217, 334, 3). Two layer have 64 channels with a kernel size of (3,3) and the same padding, followed by a max pool layer of stride (2,2) and two layers that have 128 filter sizes and 3 kernel sizes. This is followed by a max-pooling layer of stride (2, 2) which is the same as the previous layer. After that, we find two 256-filter convolution layers with a kernel size of (3, 3). In the end, there are two sets of three convolutional layers and a max pooling layer. Each has 512 filters of (3, 3) size with the same padding.

Following the max pool layer, we added a flatten layer, then 3 blocks of dense layers with ReLU activation function and dropout layers, followed by the last layer, which is the output layer, being a dense layer with Sigmoid activation function .

Fine Tuning In order to keep the weights of this big model, we opted for fine tuning the network rather than training it from the scratch, since we didn't have neither the time nor the resources to do so. The first 17 layers are hence frozen, then the final layers are allowed to train on our dataset made of spectrograms.

Training For the afore-mentioned reasons, we opted for a lighter method for training the network, including the number epochs: we chose to let the model train for only 10 epochs using Adam optimizer and binary cross-entropy as a loss function. After the 5th epoch the validation loss started to increases (figure 11) and so the model started showing signs of overfitting, that we managed to reduce using the dropout layers with 0.1 rate. Regard the split of the data, we divided them randomly and we used the 90% of them for the training and the 5% for both validation and test sets.

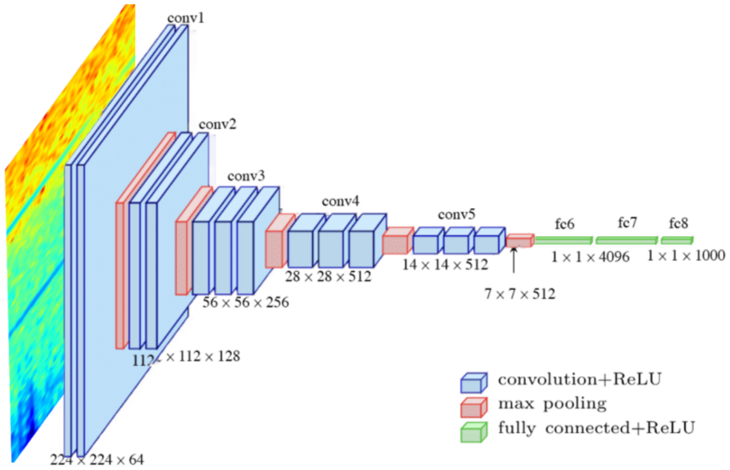

Figure 10: VGG-16 Architecture

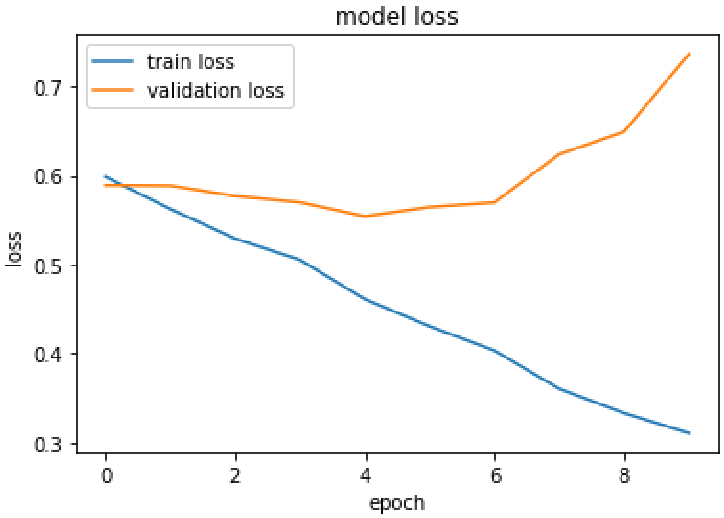

Figure 11: Train and validation losses of our model

5.4 Results

The results were higher than we expected, as we were able to reach 73.1% accuracy, while we were anticipating a 60% or 65% maximum on this task, given the short training time.

To compare our model with others we used this [8] paper, since they reached very good scores, they used the DREAMER dataset as we did and they already made a comparison with other state of the art techniques – we reported their comparison in table 3 –.

As can be seen, our results are good and we think that they could be even better, for this reason in the following section we list some experiment for a future work.

5.5 Possible future work

We stopped to improve our system due time issues: understand how to use the data and which model use was very time consuming and we wanted to write about this project in this report. We list here some experiments that in out mind are worth trying:

- Try with other pretrained model (we tried with ResNet but the performance was worse);

Table 3: Comparison of accuracy of Valence, Arousal and Dominance between our method and other state of the art

Method	Valence	Arousal	Dominance
SVM	60.40%	68.84%	75.84%
GraphSLDA	57.70%	68.12%	73.90%
GSCCA	56.65%	70.30%	77.31%
DGCNN	86.23%	84.54%	85.02%
Our method	67.06%	76.03%	76.20%

- Try to use data augmentation on the raw signal;
- Try to use data augmentation on the spectograms;
- Try to do some preprocessing to the raw signal, for example it could be interesting divide the signals in waves ($\delta, \theta, \alpha, \beta$ and γ), generate 5 spectograms, one for each wave, and feed the 5 images in parallel to a CNN;
- Try to use a LOSO partitioning of the data; the results will probably be worst, but more reliable;
- Try a multi-modal approach combining EEG with ECG.

6 Contributions

The workload was equally distributed: we both worked on the project that we did (§ 5) – we discussed each decision on how to proceed, and we both had ideas that led to the final result – and on the EEG mandatory papers (§ 2), Edoardo worked alone on the deepfake speech detection through emotion recognition (§ 3) and he wrote the intro of the paper (§ 1), while Alessandro worked alone on Emotion recognition using a combination of EEG, ECG and GSR (§ 4) and he wrote also the sections 1.1 and 1.2 which are introductions on emotions and emotion recognition through EEG.

Although some parts of the work were done by one person, we always discussed the topics to be covered and the validity of the papers.

References

[1] *Butterworth filter on wikipedia:* URL: https://en.wikipedia.org/wiki/Butterworth_filter.

[2] Mingyi Chen et al. “3-D Convolutional Recurrent Neural Networks With Attention Model for Speech Emotion Recognition”. In: *IEEE Signal Processing Letters* 25.10 (2018), pp. 1440–1444. DOI: 10.1109/LSP.2018.2860246.

[3] Emanuele Conti et al. “Deepfake Speech Detection Through Emotion Recognition: A Semantic Approach”. In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* 2022, pp. 8962–8966. DOI: 10.1109/ICASSP43922.2022.9747186.

[4] *Cross entropy loss function:* URL: https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html#cross-entropy.

[5] *Delta and delta-delta features:* URL: https://wiki.aalto.fi/display/ITSP/Deltas+and+Delta-deltas.

[6] Brian Dolhansky et al. “The DeepFake Detection Challenge Dataset”. In: *CoRR* abs/2006.07397 (2020). arXiv: 2006.07397. URL: https://arxiv.org/abs/2006.07397.

[7] *DREAMER dataset website:* URL: https://zenodo.org/record/546113#.YzMBxexBwV8.

[8] *EEG Emotion Recognition Using Dynamical Graph Convolutional Neural Networks*. URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8320798.

[9] *Emotion recognition based on convolutional neural networks and heterogeneous bio-signal data sources*. URL: https://www.sciencedirect.com/science/article/pii/S1566253521001457.

[10] *Emotion Recognition Using EEG Signals: A Survey*. URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7946165.

[11] *FACS on wikipedia:* URL: https://en.wikipedia.org/wiki/Facial_Action_Coding_System.

[12] Brian Hosler et al. “Do Deepfakes Feel Emotions? A Semantic Approach to Detecting Deepfakes Via Emotional Inconsistencies”. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).* 2021, pp. 1013–1022. DOI: 10.1109/CVPRW53098.2021.00112.

[13] *Jordan Peele video:* URL: https://www.theverge.com/tldr/2018/4/17/17247334/ai-fake-news-video-barack-obama-jordan-peele-buzzfeed.

[14] *LSTM on wikipedia:* URL: https://en.wikipedia.org/wiki/Long_short-term_memory.

[15] *Matplotlib specgram:* URL: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.specgram.html.

[16] Gary McKeown et al. “The SEMAINE Database: Annotated Multimodal Records of Emotionally Colored Conversations between a Person and a Limited Agent”. In: *IEEE Transactions on Affective Computing* 3.1 (2012), pp. 5–17. DOI: 10.1109/T-AFFC.2011.20.

[17] *opensmile:* URL: https://www.audeering.com/research/opensmile/.

[18] *SEED dataset:* URL: https://bcmi.sjtu.edu.cn/home/seed/.

[19] *Spectral graph theory:* URL: https://en.wikipedia.org/wiki/Spectral_graph_theory.