Politecnico di Torino

POLITECNICO DI TORINO

Department of Mechanical and Aerospace Engineering

Master Thesis in Biomedical Engineering

Arousal and Valence Recognition in Videos: Comparing the Power of Traditional Machine Learning and Deep Learning Models

Supervisor

Prof. Gabriella OLMO

Ing. Gianluca AMPRIMO

Candidate

Martino CONVERSANO

A.Y. 2022/2023

Abstract

This thesis explores the field of image/video recognition of continuous emotional states, with the goal of improving our understanding of human emotions and the role of non-verbal cues in their expression. This is a critical area of research that has numerous practical applications such as mental health, human-computer interaction, and marketing. One of the most important viewpoint on emotion recognition is the affective state, which can be described by two primary dimensions: arousal and valence. Arousal refers to the intensity or the energy level of the emotion, while valence refers to its pleasantness or unpleasantness. In further details, this thesis is focused on arousal and valence automatic recognition from video frames containing human subjects' faces, by applying machine learning and deep learning techniques. The purpose of this study is to compare performance between simpler models (e.g., SVM, MLP) and deep learning architectures (e.g., Resnet, VGG, MobileNet) to appreciate whether simpler models could produce comparable performance in the task, given an effective preprocessing of the input data. As a preprocessing, the raw images were cropped and realigned. Then, face landmarks were computed using the Mediapipe library and Histogram of Gradients using the Py-feat library. To reduce the number of features obtained, a principal component analysis was performed on the HOGs.

The employed data contain more then five hours of video recordings of stress-eliciting experiments in a controlled environment - e.g., a public speaking task in front of an audience. Video clips of different subjects, capturing individuals exhibiting a variety of expressions are included and annotated with arousal and valance values for each video frame. Several state-of-the-art deep learning models, including Convolutional Neural Networks (CNNs) were used to evaluate the performance in recognizing arousal and valence.

Results showed that deep learning models do not necessarily outperformed traditional machine learning models in recognizing arousal and valence, therefore a powerful preprocessing, based on relevant features of the input image could produce similar effects while saving long training time typical of deep architectures.

This work may contribute to the development of more accurate and reliable video recognition systems based on simpler and faster models.

Table of Contents

List of Tables

List of Figures

Acronyms

AI Artificial Intelligence

AU Action Unit

BPM Beat Per Minute

CCC Concordance Correlation Coefficient

CK+ Cohn-Kanade+

CNN Convolutional Neural Network

DL Deep Learning

DRML Deep Region Multi-label

ECG Electrocardiogram

EDA Electrodermal Activity

EEG Electroencephalogram

FACS Facial Action Codic System

FAU Facial Action Unit

FER Facial Emotion Recognition

HOG Histogram of Oriented Gradients

KDEF Karolinska Directed Emotional Faces

LBP Local Binary Pattern

LSTM Long Short-term Memory

MAE Mean Absolute Error

MER Multimodal Emotion Recognition

ML Machine Learning

MLP Multi-layer Perceptron

MSE Mean Squared Error

NIR Near-Infrared

PCA Principal Component Analysis

RAAW Rater Aligned Annotation Weighting

RaFD Radboud Faces Dataset

ResNet Residual Network

RESP Respiration

RNN Recurrent Neural Network

SAGR Sign Agreement Metric

SSR Sum of Squares of Regression

SST Total Sum of Squares

SVM Support Vector Machine

SVR Support Vector Regressor

Ulm-TSST Ulm-Trier Social Stress Test

VGG Visual Geometry Group

Chapter 1

Introduction

The field of artificial intelligence has become increasingly involved in various aspects of human life, enabling technology to better cater to our needs. One area where AI may be particularly useful is in the recognition of human emotions. Non-verbal communication constitutes a significant portion of human communication, and AI learning techniques can be used to help machines identify and understand these emotions.

Arousal and valence have a significant impact on many areas of research, including psychology, neuroscience, healthcare and human-computer interaction [1]. The ability to accurately recognize and measure emotions such as arousal and valence from videos can help us better understand how people respond to different stimuli and situations, and can provide valuable insights into the factors that influence human behavior. In psychology and neuroscience, the measurement of these two dimensions is used to study emotions and their effects on behavior, cognition, and physiological responses [2]. In human-computer interaction, they are used to design more effective and engaging systems that can adapt to the user's emotional state [3].

Facial expressions are a universal means of conveying emotions, and algorithms can be trained to recognize these expressions and estimate the corresponding emotions. Unlike humans who have been using facial recognition and emotion recognition in our daily lives for a long time, computers were initially not as adept at these tasks. However, with the advancement of computer hardware, computers have now become capable of performing these tasks. This has led to increased interest in face

recognition, emotion recognition, and related topics among researchers in various fields. The face is considered the area where emotions are most concentrated, and expressions have been divided into several patterns that can be recognized by computers.
The main objective of this thesis is to evaluate the performance of different machine learning algorithms for recognizing arousal and valence from video data, and to develop a model that can achieve high accuracy while maintaining a structure that is as simple as possible. Performance of different algorithms (e.g., support vector machines, multilayer perceptron) and preprocessing methods are compared to find the best combination among them. Subsequently, the results obtained were compared with state-of-the-art deep neural networks to verify the goodness of the selected model.

1.1 Thesis Outline

To accomplish the primary objectives of this thesis, the following outline is proposed:

- Examine and elucidate the primary datasets and comprehend their benefits and limitations.
- Develop distinct solutions for preprocessing the data to train the models. Consider different modalities of data as features, such as facial landmarks (i.e., facial geometry), facial action unit.
- Analyze, implement, and evaluate a machine learning model to recognize facial action unit from video frames using Python, Sklearn.
- Analyze, implement, and evaluate a machine learning model to recognize facial expressions from video frames using Python, Sklearn, OpenCV.
- Analyze, implement, and evaluate a deep learning system for recognizing facial expressions from video frames using Keras.

Chapter 2

Background Theory

In this chapter, the theory within the relevant fields of the thesis are covered. It delves into the theoretical aspects of Emotions, Facial Action Unit, Machine Learning and Deep Learning. In addition, metrics and tools used in the thesis are described.

2.1 Theory Of Emotions

The theory of emotions [2] [4] has been the subject of study for several centuries. It proposes a categorical classification system, where emotions are classified as discrete entities, independent of each other and easily distinguishable. Emotions have been defined as complex psychological states that are linked to both subjective feelings and physiological changes. Researchers have identified a variety of emotions that are common to all humans, such as happiness, sadness, fear, and anger. Emotions play a significant role in human cognition, behavior, and decision-making. With the development of technology, the study of emotions has expanded into the field of affective computing, which involves developing machines and algorithms that can recognize, interpret, and respond to human emotions.

2.1.1 Affective Computing

Affective computing is a field of computer science that deals with the development of systems and devices that can recognize, interpret, process, and simulate human

emotions. The goal of affective computing is to create machines that can interact with humans in more natural and intuitive ways by understanding and responding appropriately to human emotions [5]. This involves using passive sensors, such as microphones and video cameras, to capture a person's physical state or behavior, including speech, tone of voice, facial expressions, body posture, and gestures. These data are then analyzed and processed to extract emotional information.
There are several major challenges in affective computing. One of the main challenges is the ambiguity and variability of human emotions. Emotions are complex and multi-dimensional, and can be influenced by a wide range of factors, such as culture, gender, personality, and context [6]. Another challenge is the lack of reliable and objective measures for assessing emotions. Emotions are typically self-reported or observed through external cues, such as facial expressions, vocal tone, and physiological signals, which can be subjective and prone to error. In addition, there is a need for interdisciplinary collaboration and integration of knowledge from different domains, such as psychology, neuroscience, computer science, and engineering, to advance the field of affective computing [7]. Despite the challenges that come with recognizing and interpreting human emotions, the development of affective computing has the potential to revolutionize the way we interact with technology and improve the quality of our lives.

2.1.2 Arousal and Valence

In the 2000s, a dimensional approach to emotions was proposed [1], which facilitates their identification and characterization. The circumflex model of emotions, in fig. 2.1, argues that actual states can be traced to two main neurophysiological axes. One axis explains the valence of the emotion (pleasantness-unpleasantness) while the other refers to the corresponding level of arousal/physiological activation.
In general, positive values of arousal indicate a high level of physiological activation, such as a high heart rate, increased sweating, rapid breathing. Conversely, negative values of arousal indicate a low level of physiological activation. Regarding valence, positive values indicate a feeling of pleasure, happiness, or gratification, while negative values indicate a feeling of unpleasantness, sadness, or frustration. However, it is important to consider that values of arousal and valence can vary considerably depending on the individual and the situation. Both variables can take values in the

range [-1,1] and each emotion can be explained as the linear combination between the two dimensions. Similarly, emotions such as excitement or fear are associated with high arousal, while emotions such as boredom or relaxation are associated with low arousal. This model allows for a more comprehensive explanation (than a categorical model) of data from neuroimaging studies and comorbidity among different affective and psychological disorders[8].

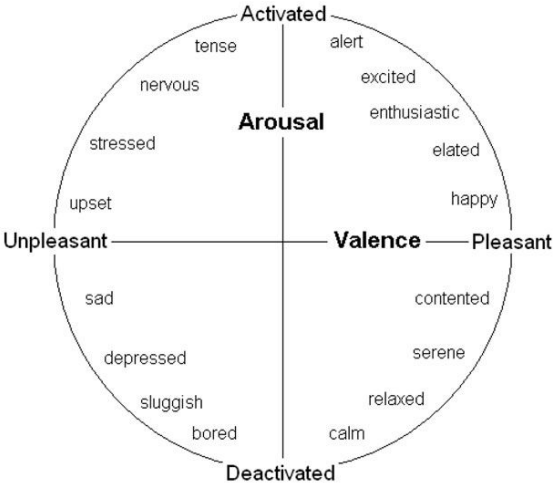


Figure 2.1: The Circumflex Model of Emotion[9]

2.2 Facial Emotion Recognition

Facial emotion recognition is a process that involves the detection of human emotions from facial expressions. In human communication, facial emotions are crucial because they help us to understand the intentions of others and infer their emotional state. In recent years, facial emotion recognition has become an increasingly active field in affective computing due to its potential in various applications. However, the task of accurate and robust facial emotion recognition by computer models is still a challenging one due to the heterogeneity of human faces, different face position, and different light conditions. Fig. 2.2 and Fig. 2.3 show two instances where individuals display comparable emotions, but respectively, the lighting conditions and facial angles differ. In addition, among different populations of the world, there

are differences in the way emotions are manifested and facial mimics making this task even more complex [6].

While in previous years images constituted the most widely used datasets to train models on this task, the use of videos is gaining more attention, as it allows for the prediction of dynamic facial emotion expressions, leveraging the temporal continuity in the transition between emotional states. The integration of CNNs and LSTM has been proposed to address the temporal aspect of emotion recognition in videos [10].

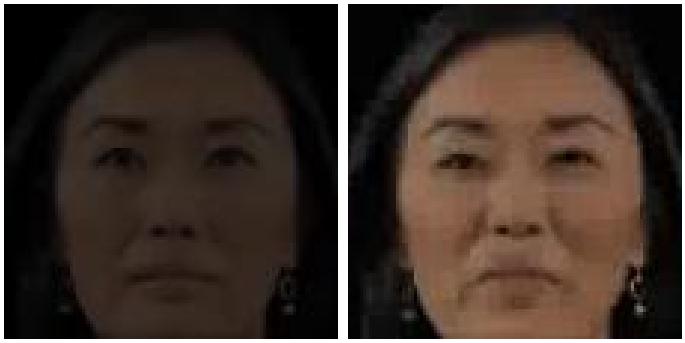


Figure 2.2: Example of a subject expressing a similar emotion under two different lighting conditions. Images taken from the Aff-Wild2 dataset [11].

Figure 2.3: Example of a subject expressing a similar emotion with two different face position. Images taken from the MuSe-Stress sub-challenge dataset [12].

2.2.1 Preprocessing and Features

Preprocessing is a key step for all FER models. Considering the challenges presented in the previous section, preprocessing is used to reduce all changes in the image that are not due to the change in the subject's emotional state. When images of a face changing expressions are provided, there are many other features that can vary such as pose and lighting. The predictive model must be able to focus only on changes in emotional state while neglecting all others.
The most common pipeline for preprocessing consists of the following steps: face detection, resizing and normalization. Face detection consists of detecting the face of the subject in the image so that it can be extracted by removing all unnecessary features in the image. After that, image resizing is performed to make all frames equal and fit the input shape of the model. Finally, a normalization of the image is performed to reduce the differences in illumination.
Especially for machine learning models, the next step is feature extraction. It involves selecting the highlight features of the image or video, such as lines, contours, textures that can help distinguish and classify the subject's emotionality. The most common features extracted for this task are shown below [13].

Histogram of Oriented Gradients
The Histogram of Oriented Gradients (HOG) works by analyzing the distribution of gradient directions in an image. First, the image is divided into small cells, and for each cell, a histogram of gradient orientations is computed. The orientations are calculated by taking the gradient of the image in the x and y directions. The histograms are then normalized, and neighboring cells are combined into blocks. The resulting feature vectors are used to represent the image as shown in fig. 2.4. The HOG algorithm is particularly effective at capturing local image features because when a variation in intensity is detected in the image, the amplitude of the gradient changes.

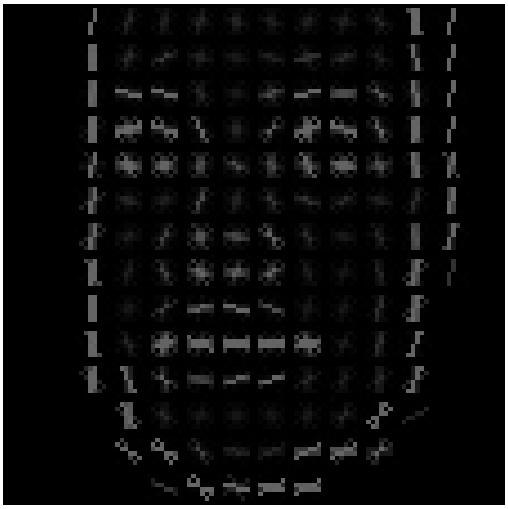


Figure 2.4: Example of HOG calculated with Py-feat

Local Binary Pattern

Local Binary Pattern (LBP) is a texture descriptor that encodes the local texture pattern of an image. LBP was first introduced by [14] and has since become a popular feature extraction technique in computer vision and image analysis. The method involves dividing an image into small cells and comparing the intensity of the center pixel to its neighbors in a circular pattern. A binary code is then generated based on whether the intensity of the neighbors is greater or less than the center pixel. This binary code is used to describe the texture of the cell, and the process is repeated for each patch in the image. The resulting LBP histogram represents the distribution of different texture patterns in the image. An example of the output obtained on an image of a face is presented in fig. 2.5. LBP is computationally efficient and can be applied to both grayscale and color images and is also robust to monotonic grayscale changes caused by, for example, illumination.

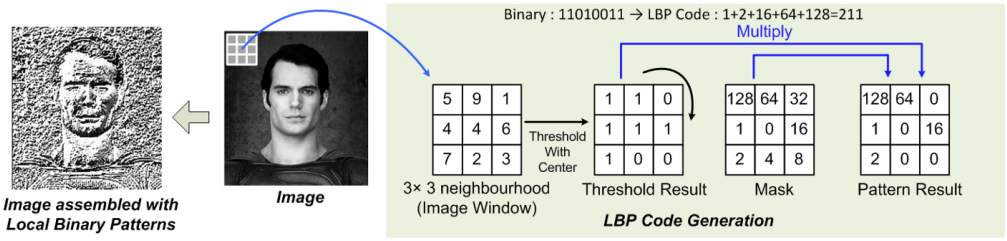


Figure 2.5: Example of LBP code generation [15]

Face Landmarks

Another feature often used in this area is face landmarks because they allow to find the location of the major points of interest on the subject's face. Face landmarks are specific points on a face that can be used to identify and track facial features. They are typically represented as 2D or 3D coordinates that correspond to key facial features, such as the eyes, nose, mouth, and eyebrows. The accuracy of landmark detection is critical for the success of many computer vision applications, as even small errors in landmark detection can significantly impact downstream tasks. The illustration in fig. 2.6 displays an instance of a facial mesh that has been computed using Mediapipe, which includes 478 three-dimensional landmarks.

Figure 2.6: Example of face mesh calculated with MediaPipe

CNN Feature Extraction

Although traditional feature extraction methods have been successful, recent advances in CNNs have shown significant progress in learning features automatically. Compared to traditional methods where features are manually defined, CNN has the ability to extract undefined features from a training database. CNN achieves high performance by extracting shift-invariant local features from input images through the concept of the local receptive field, shared weight, and spatial subsampling [16].

2.2.2 Facial Action Unit

Facial Action Units (FAUs) are the fundamental building blocks of facial expressions [17]. Each FAU represents a specific movement or deformation of facial muscles, and their combinations generate a wide range of emotional expressions. In the 1970s, Paul Ekman and Wallace Friesen developed the Facial Action Coding System (FACS) to identify and describe the different FAUs. The system is widely used in emotion recognition research and has been adopted in the development of automatic systems for facial expression analysis.

There are 27 FAUs in total, including both observable movements (e.g., raising eyebrows, wrinkling nose) and subtle muscle contractions (e.g., lip corner puller). Each FAU is represented by a code, which indicates the specific muscles involved in the facial expression. Table 2.1 outlines the codes and corresponding facial movements of the action units in the MuSe-Stress sub-challenge dataset [12]. For example, AU12 represents the contraction of the zygomatic major muscle, which raises the corners of the mouth and creates a smile. Fig. 2.7 is an example of comprehensive FACS coding of a facial expression.

By detecting and measuring the intensity of specific FAUs, it is possible to identify and classify emotional states. Automatic systems for facial expression analysis use machine learning algorithms to recognize and classify FAUs in real-time.

Although FAUs provide a standardized way to describe facial expressions, they do not capture the full range of human emotional experience. Emotions are complex and dynamic, and facial expressions are only one aspect of emotional communication.

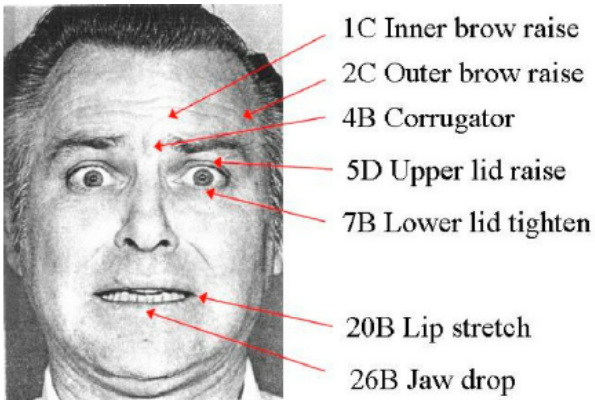


Figure 2.7: An illustration is provided to demonstrate a thorough FACS coding of a facial expression. The coding involves assigning numerical values to action units, which correspond to individual facial muscles, and using letters (A-E) to denote the level of activation.[18]

AU	Name	AU	Name
01	Inner brow raiser	14	Dimpler
02	Outer brow raiser	15	Lip corner depressor
04	Brow lowerer	17	Chin raiser
05	Upper lid raiser	20	Lip stretcher
06	Cheek raiser	23	Lip tightener
07	Lid tightener	24	Lip pressor
09	Nose wrinkler	25	Lips part
10	Upper lip raiser	26	Jaw drop
11	Nasolabial deepener	28	Lip suck
12	Lip corner puller	43	Eyes closed

Table 2.1: Facial Action Coding System. The number assigns a code to the specific facial movement.

2.3 Machine Learning

Machine Learning is a subset of Artificial Intelligence that involves the use of algorithms and statistical models to enable computer systems to learn and improve their performance on a specific task without being explicitly programmed [19]. In other words, the computer is trained on a dataset and uses that information to make predictions or decisions about new data it encounters. The main goal of

machine learning is to create models that can accurately predict or classify new data based on patterns and relationships discovered in the training data. There are several types of machine learning, including supervised learning, unsupervised learning, and reinforcement learning, each with its own approach and applications. Machine learning is used in a variety of fields, including natural language processing, computer vision, and data analysis.

2.3.1 General Machine Learning Terms

Epoch

An epoch refers to a complete pass through a training dataset during the learning process of a model. During an epoch, the model processes the entire dataset, and the weights of the model are adjusted to minimize the error between the predicted output and the target one. The number of epochs is a hyperparameter that is set before training and determines how many times the model will cycle through the entire dataset.

Learning rate

The learning rate is a hyperparameter that determines how much the model should adjust the weights and biases of the input features in each iteration of the training process. A high learning rate may cause the optimization algorithm to overshoot the minimum of the cost function, leading to unstable training and poor performance. On the other hand, a low learning rate may result in slow convergence and longer training times. Therefore, selecting an appropriate learning rate is a crucial step in training models.

Activation function

In artificial neural networks, an activation function is a mathematical function applied to the output of a neuron or a group of neurons. It determines whether the neuron should be activated or not, based on the weighted sum of its input values. The activation function adds non-linearity to the neural network and is essential in the process of training the network. There are various types of activation functions, including sigmoid, tanh, ReLU, and softmax, each with its unique characteristics and suitability for different types of task. The choice of activation

function can affect the performance of the neural network in terms of its accuracy, speed of convergence, and ability to handle non-linear relationships in the data.

Loss function

A loss function is a mathematical function that measures the difference between the predicted values of a model and the actual target values. It represents the discrepancy between the predicted output and the true output, and its goal is to minimize this difference during the training process. In other words, the loss function is a way to quantify the error of the model, and the optimization algorithm tries to minimize it by adjusting the model's parameters. The choice of the loss function depends on the specific task and the type of data being used. For example, the mean squared error (MSE) loss function is commonly used for regression problems. The design of an appropriate loss function is crucial for the success of a machine learning model, as it affects the model's ability to accurately predict outcomes.

Overfitting

Overfitting is a common issue that can occur when training machine learning models. It refers to the phenomenon where the model becomes too complex and starts to fit the training data too closely, leading to poor performance on new, unseen data. This can be caused by various factors, such as using an overly complex model, having insufficient training data, or training for too long.

To mitigate overfitting, various techniques can be used, such as regularization, early stopping, and data augmentation. Regularization involves adding a penalty term to the loss function to discourage large weights and reduce model complexity. Early stopping involves monitoring the validation loss and stopping the training when the performance starts to deteriorate. Data augmentation involves generating new synthetic training samples by applying transformations such as rotation, scaling, and flipping.

Variance and Bias

In machine learning, the concepts of variance and bias are critical for evaluating the performance of a predictive model. Variance refers to the model's ability to fit to the training data, while bias refers to the model's ability to generalize to new

data. Models with high variance are overly complex and have a high degree of flexibility, leading to overfitting to the training data. In contrast, models with high bias are too simplistic and unable to capture the underlying patterns in the data, resulting in underfitting.
To evaluate the performance of a model, it is essential to assess both its variance and bias. Techniques, such as cross-validation, can help to identify whether a model is overfitting or underfitting and find the optimal balance between variance and bias.

Hyper-parameters

Hyperparameters refer to the parameters that are set prior to the training of a model and cannot be learned during the training process itself. These parameters include settings such as learning rate, hidden layers, and batch size. Choosing appropriate hyperparameters is a critical step in building effective machine learning models. The optimal values for these parameters can vary based on the dataset, the complexity of the model, and the computational resources available. As a result, tuning hyperparameters is often an iterative process that involves running experiments and evaluating performance on a validation set. Hyperparameter tuning can be a time-consuming and computationally expensive task, but it is necessary for achieving optimal model performance. Techniques such as grid search, random search, and Bayesian optimization can be used to efficiently search the hyperparameter space and find the best values for a given model.

Grid search

Grid search is a common technique used to identify the optimal hyperparameters for a given model. This method involves systematically testing different combinations of hyperparameters to determine which combination results in the best model performance. The hyperparameters to be tuned are selected a priori and a grid of possible values for each hyperparameter is defined. The grid search algorithm then performs an exhaustive search through the grid, evaluating the performance of the model for each combination of hyperparameters. The combination that yields the best performance, as determined by a chosen evaluation metric, is then selected as the optimal set of hyperparameters for the model. Grid search is a computationally expensive, but it is often necessary to ensure that the model is performing at its best.

Cross-validation

Cross-validation and train/test split are techniques commonly used to evaluate the performance of a model on unseen data. Cross-validation involves dividing the dataset into k-folds, where each fold is used as a validation set while the remaining k-1 folds are used for training. This process is repeated k times, with each fold being used once as the validation set. The performance of the model is then averaged across all k-folds. On the other hand, train/test split involves randomly dividing the dataset into a training set and a test set. This approach is straightforward and enables the model to become less reliant on the selected split, resulting in more accurate performance. Both cross-validation and train/test split are important techniques to ensure that the model is not overfitting to the training data and can generalize well to new, unseen data.

2.3.2 Supervised Learning

Supervised learning is a type of machine learning where the algorithm learns to map input data to output data based on labeled examples provided during the training phase. In other words, the algorithm is presented with a set of inputs and their corresponding correct outputs, and it learns to make predictions on new inputs based on this labeled data.

During the training phase, the algorithm adjusts its parameters to minimize the difference between its predicted output and the true output. This process is also known as optimization, and the algorithm uses a loss function to measure how well it is performing.

Once the model is trained, it can be used to make predictions on new, unseen data. The accuracy of the model is evaluated on a separate set of data, called the validation set, to ensure that it is not overfitting to the training data.

2.3.3 Support Vector Machine

Support Vector Machine (SVM) is a popular machine learning algorithm used for classification and regression analysis. The basic idea behind SVM is to find a hyperplane in a high-dimensional space that can best separate different classes or

predict a continuous output.
In the case of classification, SVM finds the hyperplane that maximizes the margin between different classes. The margin is the distance between the hyperplane and the closest data points of each class. The hyperplane is chosen so that the margin is maximized, which means it is the most robust and generalizable classifier. SVM can handle both linearly separable and non-linearly separable data by using a kernel function to map the original features into a higher-dimensional space, where it is possible to find a hyperplane that can separate the classes [20].
In the case of regression, SVM is used to predict a continuous output. The goal is to find a function that can fit the data with minimum error. The regression version of SVM is called Support Vector Regression (SVR). SVR uses the same principle as SVM, but instead of finding a hyperplane that maximizes the margin, it finds a hyperplane that minimizes the deviation between the predicted output and the actual output. SVR can also use kernel functions to handle non-linear data. In fig. 2.8, a graphical representation of the research for hyperplanes that maximize the margin is shown.
In summary, SVM is a versatile machine learning algorithm that can be used for both classification and regression tasks. Its ability to handle non-linear data and find the most robust hyperplane or function makes it a popular choice in various fields.

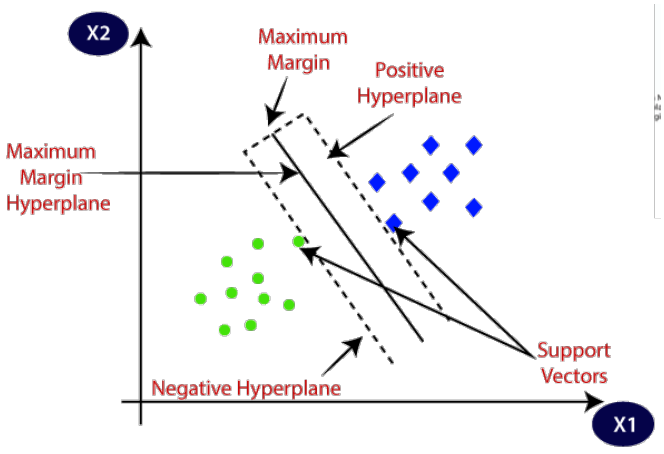


Figure 2.8: SVM hyperplane separation[21]

2.3.4 Multi-layer Perceptron

A multi-layer perceptron (MLP) is a type of neural network that consists of multiple layers of interconnected nodes or neurons. The MLP is called a feedforward network because the input is fed forward through the network to produce an output. The first layer of the MLP is the input layer, followed by one or more hidden layers, and an output layer. Each neuron in a hidden layer receives inputs from the neurons in the previous layer and produces an output based on its weights and activation function. The output of the final layer is the output of the network. The weights and biases of the network are learned through the training process, which involves adjusting the weights to minimize the error between the predicted output and the actual output [22]. Fig. 2.9 shows an example of MLP architecture.
In the case of a MLP regressor, the output of the network is a continuous value, rather than a discrete class label as in the case of a classifier. The MLP regressor is trained on a set of input-output pairs, and the goal is to learn a function that maps the inputs to the corresponding outputs. During training, the weights of the network are adjusted to minimize the error between the predicted output and the true output. The choice of loss function and activation function can affect the performance of the MLP regressor, and hyperparameter tuning is often necessary to achieve optimal performance.

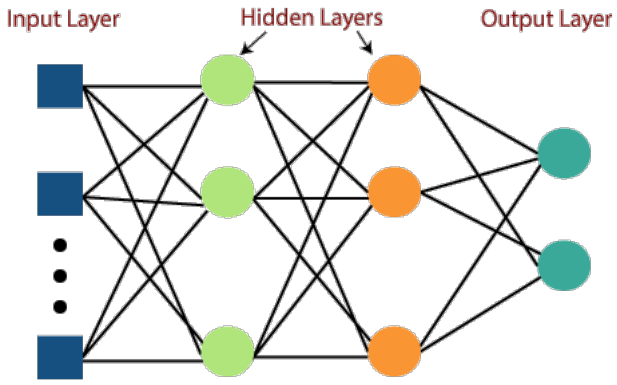


Figure 2.9: MLP basic architecture[22]

2.4 Deep Learning

Deep learning is a subset of machine learning that involves the use of artificial neural networks with multiple layers to learn and make predictions from complex data. Deep learning algorithms are designed to automatically learn hierarchical representations of the input data by using layers of interconnected nodes, also called neurons. Each layer in the network performs a set of mathematical operations on the data, transforming it into a higher-level representation that can be used by the next layer. The more layers the network has, the deeper it is considered, hence the term "deep learning". One of the key advantages of deep learning is its ability to learn features directly from raw data, eliminating the need for manual feature extraction. This makes deep learning particularly well-suited for applications such as image recognition, natural language processing, and speech recognition, where the input data is complex and high-dimensional. However, deep learning algorithms require a large amount of data to be trained and can be computationally intensive, making them more challenging to implement than traditional machine learning algorithms.
A CNN is a type of deep neural network commonly used in image and video recognition. CNNs are designed to automatically extract features from input images through multiple layers of convolutional, pooling, and fully connected layers.

2.4.1 General Deep Learning Terms

In addition to the terms already presented for machine learning, an overview is given on the fundamental aspects of deep learning.

Convolutional Layers
The convolutional layer is a fundamental component of CNNs used in deep learning. It performs a mathematical operation called convolution on the input data, which involves applying a set of filters or kernels to extract features from the input data. A mathematical example is provided in fig. 2.10. The output of the convolutional layer is then passed on to other layers for further processing. The filters are learned

during training using backpropagation and are optimized to extract relevant features from the input data.

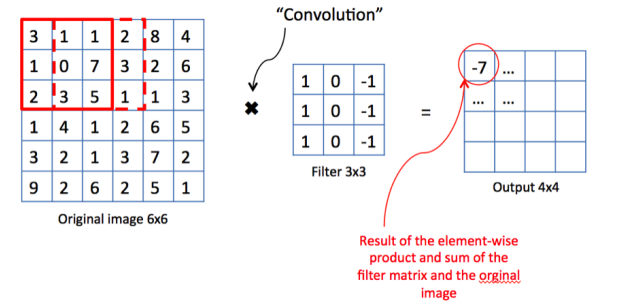


Figure 2.10: Convolution layer example[23]

Pooling Layers

A pooling layer is a type of layer used to reduce the spatial dimensions of the input feature maps. The pooling layer operates on each feature map independently and replaces a local neighborhood of the map with a summary statistic, such as the maximum value or average value within that neighborhood. A mathematical example of max pooling layer is provided in fig. 2.11. The main purpose of pooling is to help reduce the spatial resolution of the feature maps, making them more computationally efficient to process in later layers of the network. Pooling can also help to extract important features and reduce the effect of small translations or distortions in the input data.

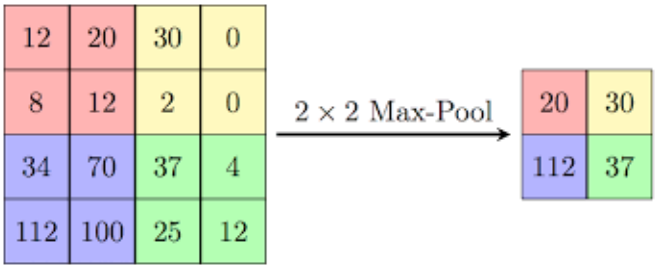


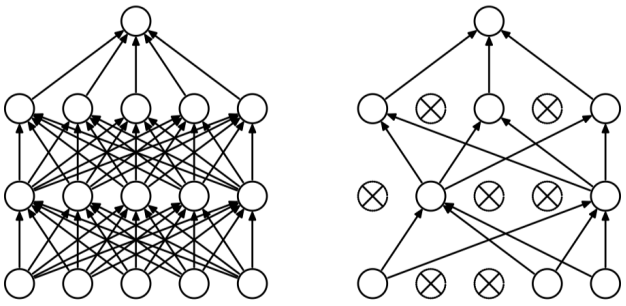
Figure 2.11: Max pooling layer example[23]

Fully Connected Layers

Fully connected layers, also known as dense layers, are a type of neural network layer where each neuron is connected to every neuron in the previous layer. In a fully connected layer, the output of each neuron is a weighted sum of the inputs, followed by an activation function. These layers are typically placed at the end of a neural network architecture, where they take the output of the preceding layers and produce a final output or prediction. The number of neurons in a fully connected layer is typically a hyperparameter that needs to be tuned during the training process.

Dropout Layer

A dropout layer is a type of regularization technique used in neural networks to prevent overfitting [24]. During training, a certain percentage of randomly selected neurons in the layer are ignored or "dropped out," meaning their outputs are set to zero. A visual example of the change in neural network after inserting a dropout layer is presented in fig. 2.12. This helps prevent the network from relying too heavily on any one neuron or feature and forces it to learn more robust and generalizable representations. Dropout layers are typically inserted between fully connected layers in a neural network and the dropout rate is a hyperparameter that can be tuned to achieve optimal performance.



(a) Standard Neural Net (b) After applying dropout.

Figure 2.12: Dropout layer example[23]

Transfer Learning

Transfer learning is technique that involves reusing pre-trained models on a new task that is related to the original task. It is a highly effective and widely used approach in deep learning, especially for computer vision tasks. In transfer learning, the pre-trained model is fine-tuned on a new dataset, allowing for faster convergence and better performance than training a model from scratch. This is particularly useful when the new dataset is small or when computational resources are limited. The popularity of transfer learning has led to the development of many pre-trained models, such as VGG, ResNet, and MobileNet, which can be used as a starting point for transfer learning tasks.

2.5 Deep Learning Neural Networks

In this section, an overview of the most commonly used deep learning neural networks is presented. The architectures of these networks are described along with their specific applications and advantages. The aim is to provide a comprehensive understanding of the current state-of-the-art model in deep learning, their applications, and their potential for future research.

2.5.1 ResNet

Residual Network (ResNet) is a type of deep neural network architecture that was introduced in 2015 by researchers at Microsoft Research [25]. The key innovation of ResNet was the use of residual connections, which allows the network to be deeper without suffering from the vanishing gradient problem. In a traditional neural network, each layer is trained to fit the input to its corresponding output. However, in a ResNet, each layer is instead trained to fit the residual, or the difference between the input and its corresponding output. This residual block architecture helps to prevent the degradation problem that occurs when adding more layers to a neural network, and enables the construction of very deep neural networks with improved performance. Different models of ResNet can be found, each with a different number of layers. The first models, ResNet-18 and ResNet-34, have 18 and 34 layers, respectively. Then there are ResNet-50, in fig. 2.13, ResNet-101 and

ResNet-152, which have 50, 101 and 152 layers.

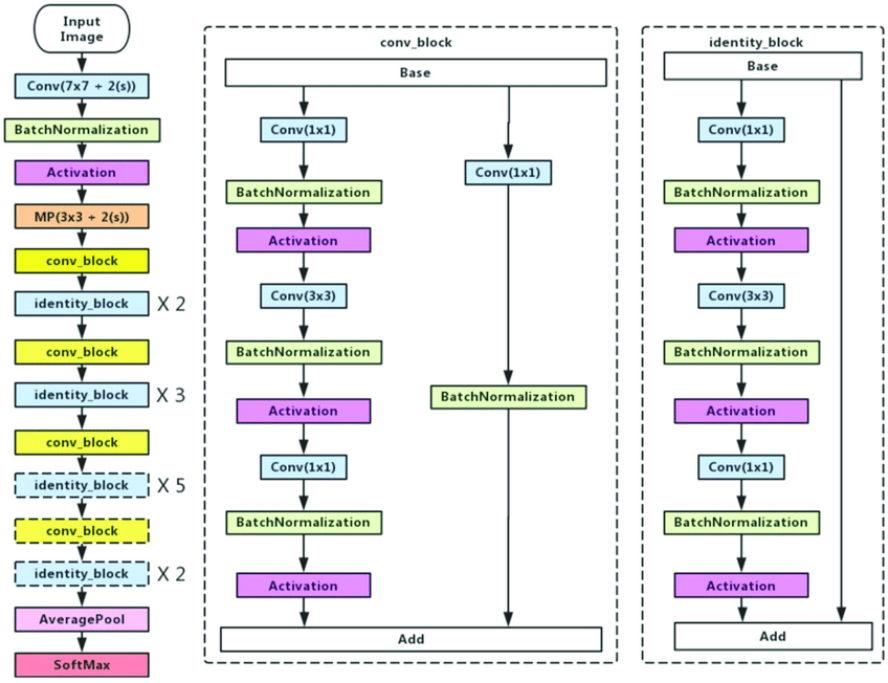


Figure 2.13: ResNet 50 architecture[26]

2.5.2 VGG

Visual Geometry Group (VGG) is a deep convolutional neural network architecture developed by researchers at the University of Oxford in 2014 [27]. The network is named after the group's initials and the number of layers it contains (VGG-16 and VGG-19, for example, have 16 and 19 layers, respectively). The VGG architecture achieved impressive performance on image classification tasks and is known for its simplicity and uniformity of design, with the same filter size of 3x3 and max pooling of 2x2 used throughout the entire network. VGG has since become a benchmark for evaluating the performance of new convolutional neural network architectures.

2.5.3 MobileNet

MobileNet is a convolutional neural network architecture designed for mobile and embedded devices with limited computing resources [28]. It uses depthwise separable convolutions, which factorize the standard convolution operation into a depthwise convolution and a pointwise convolution. This reduces the number of computations required for each convolution operation and allows MobileNet to achieve a good trade-off between accuracy and efficiency. MobileNet has several versions with different levels of complexity, ranging from MobileNetV1 to MobileNetV3, and has been widely used in applications on mobile and embedded devices.

2.5.4 Long Short-Term Memory

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture that has gained popularity in recent years due to its ability to handle long-term dependencies [29]. LSTM is designed to overcome the vanishing gradient problem faced by traditional RNNs by introducing a memory cell and three gating mechanisms: input gate, forget gate, and output gate. These gates control the flow of information into and out of the memory cell, allowing the network to selectively remember or forget information over time.

2.6 Multimodal Emotion Recognition

Multimodal emotion recognition is a challenging task that aims to recognize and interpret human emotions using multiple modalities such as facial expressions, voice, and physiological signals. To achieve accurate recognition, several techniques for fusing the modalities have been proposed. One such technique is early fusion, which combines the feature vectors of all modalities into a single vector, allowing for a single classifier to be trained. Another technique is late fusion, where each modality is processed separately, and the decisions are combined at a later stage [30]. Late fusion can be done using several approaches, including decision-level fusion, feature-level fusion, and score-level fusion. In decision-level fusion, the decisions of each modality are combined using a voting scheme or other aggregation methods. In feature-level fusion, the features of each modality are combined before

classification. Finally, in score-level fusion, the scores of each modality are combined to make a final decision.
Multimodal emotion recognition has several advantages over unimodal approaches. By combining multiple modalities such as audio, video, and physiological signals, a more complete picture of a person's emotional state can be obtained, leading to more accurate recognition. Additionally, multimodal approaches are more robust to noise and variability in the input data, as different modalities may capture different aspects of the emotional state.
However, there are also some disadvantages to using multimodal approaches. First, integrating multiple modalities requires careful design and implementation of feature extraction, feature fusion, and classification algorithms, which can be challenging and time-consuming. Second, multimodal systems require more complex hardware and software, which can be more expensive and difficult to deploy in real-world scenarios, and more invasive data collection methods.
While each fusion technique has its strengths and weaknesses, selecting the appropriate technique depends on the characteristics of the modalities and the requirements of the application.

2.7 Metrics

This section presents metrics that are often used to evaluate the quality of regression models in valence/arousal recognition.

2.7.1 Concordance Correlation Coefficient

The Concordance Correlation Coefficient (CCC) is a statistical measure used to evaluate the agreement between two quantitative variables. It is a modification of the Pearson correlation coefficient that takes into account both the precision and accuracy of the measurements. The CCC ranges from -1 to 1, where a value of 1 indicates perfect agreement, 0 indicates no agreement, and -1 indicates perfect disagreement. The CCC is widely used in various fields to assess the reliability of

measurements and to compare different measurement methods.

$$CCC = \frac{2 * \sigma 12}{(\mu 1 - \mu 2)^2 + \sigma 1^2 + \sigma 2^2} \tag{2.1}$$

Where $\sigma 1$ and $\sigma 2$ represent the standard deviations of the variables, $\mu 1$ and $\mu 2$ denote their means, and $\sigma 12$ stands for their covariance.

2.7.2 Root Mean Squared Error

Root mean squared error (RMSE) is a commonly used metric to measure the difference between predicted and actual values in regression analysis. It is calculated as the square root of the average squared difference between the predicted values and the actual values. The RMSE provides a measure of how well the model fits the data, with lower values indicating a better fit. It is particularly useful when there are outliers in the data that can have a large effect on the accuracy of the model.

$$RMSE = \sqrt{(\frac{1}{n}) \sum_{i=1}^{n} (y_i - x_i)^2} \tag{2.2}$$

2.7.3 Mean Absolute Error

Mean absolute error (MAE) is a common metric used to evaluate the performance of a model in regression tasks. It measures the average magnitude of the errors between predicted and actual values, without considering their direction. It is calculated by taking the absolute difference between the predicted and actual values, and then averaging these differences across all samples. The MAE is a measure of the model's accuracy, with lower values indicating better performance.

$$MAE = (\frac{1}{n}) \sum_{i=1}^{n} |y_i - x_i| \tag{2.3}$$

2.7.4 Sign Agreement Metric

In the study presented in [31], the authors introduce another metric for assessing the effectiveness of a system's ability to predict valence and arousal, which is known

as SAGR. SAGR is defined as:

$$SAGR = (\frac{1}{n}) \sum_{i=1}^{n} \delta \left(sign \left(y_i \right), sign \left(x_i \right) \right) \tag{2.4}$$

Where δ is the Kronecker delta function, defined as:

$$\delta \left(a, b \right) = \begin{cases} 1 & , \ a = b \\ 0 & , \ a \neq b \end{cases} \tag{2.5}$$

2.7.5 Coefficient of determination

The coefficient of determination, denoted as R^2, is a statistical metric used to measure the proportion of variability in the dependent variable that is explained by the independent variable(s) in a regression model. It is a value between 0 and 1, where 0 indicates that the model explain no variability in the dependent variable and 1 indicates that the model explains all the variability. R^2 is calculated by taking the ratio of the sum of squares of the regression (SSR) to the total sum of squares (SST), where SSR represents the sum of squared differences between the predicted and actual values, and SST represents the sum of squared differences between the actual values and the mean of the dependent variable. An R^2 value closer to 1 indicates a better fit of the model to the data, while a value closer to 0 indicates a poor fit.

$$SS_{res} = \sum_{i=1}^{n} \left(y_i - x_i \right)^2 SS_{tot} = \sum_{i=1}^{n} \left(y_i - \hat{y} \right)^2 \tag{2.6}$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{2.7}$$

2.8 Tools

This section provides an overview of the tools and libraries used in this thesis.

2.8.1 Scikit-Learn

Scikit-learn is a popular open-source machine learning library for the Python programming language. It provides a wide range of supervised and unsupervised learning algorithms, as well as tools for data preprocessing, model selection and evaluation, and data visualization. Scikit-learn is built on top of other scientific computing packages such as NumPy, SciPy, and matplotlib, making it easy to integrate into existing Python data analysis workflows.

2.8.2 Keras

Keras is an open-source software library for building and training deep neural networks. It provides a user-friendly interface for designing, training, and evaluating neural network models, and supports various types of neural network architectures such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and more. Keras is written in Python and supports both CPU and GPU computations.

2.8.3 Py-feat

Py-feat is a Python library for feature extraction and preprocessing in machine learning and signal processing applications. It provides a set of functions and tools to extract a variety of features from audio, image, and time-series data. The library aims to be fast, efficient, and user-friendly, and is designed to work seamlessly with other popular Python libraries such as scikit-learn, TensorFlow, and Keras.

2.8.4 Mediapipe

MediaPipe is a cross-platform framework developed by Google that provides tools and building blocks for building multimedia pipelines. It offers a wide range of pre-built components that can be easily integrated to build complex pipelines for

tasks such as object detection, pose estimation, face detection and recognition, facial landmarks detection, and more.

Chapter 3

Related works on preprocessing and neural networks

In this chapter, a review of the existing research in the field of FAUs detection is presented, followed by a discussion on the state-of-the-art approaches for FER and the challenges faced in the field. Additionally, a section is dedicated to the presentation of the state-of-the-art approaches within the field of multimodal emotion recognition.

3.1 Methods In Facial Action Unit Detection

This section provides a description of methods related to preprocessing, feature extraction, and classification in the field of FAUs detection.

3.1.1 Preprocessing

The primary purpose of facial image preprocessing is to improve the quality of images and enhance their features for further processing. Therefore, the first steps in image preprocessing, for both FAUs detection and emotion recognition tasks, are aimed at achieving this objective. Face detection, which is crucial for further

analysis, has been implemented using various algorithms that have demonstrated good performance.
In the cited works, different algorithms and techniques are employed for face detection. The Haar Cascades and Viola-Jones algorithms are used in [32] and [33], respectively, to detect facial features in images. [34] compares the two algorithms and finds that HOGs is slightly more accurate in detecting faces, particularly in images with multiple faces. On the other hand, [35] highlights the challenges in facial recognition due to variations in illumination conditions, which can ultimately affect the recognition process. An experiment shows an improvement in face recognition by enhancing the intensity in regions that are inadequately illuminated and decreasing it in densely illuminated regions while retaining the intensity in fairly illuminated portions. [36] and [37], which were used within the project for feature extraction, rely on face detection using fast neural networks.

3.1.2 Feature Extraction

In FER and FAUs detection, features extraction, that accurately represent the class of the image, is essential. Following image preprocessing, the extracted features must be appropriately represented before being input into a machine learning classifier. In Section 2.2.1, various types of features were discussed, including CNNs, which are currently one of the most popular approaches. However, comparing the feature extraction performance between different CNNs is challenging, as it is done automatically as a result of the CNN layers. Classical approaches, such as Local Binary Pattern, Histogram of Oriented Gradients, and facial landmarks have demonstrated good performance as noted in [38].

3.1.3 Classification

In [38], SVM is used for multi-class classification using one-against-one strategy. The classes in the SVM correspond to the five levels of intensity of a specific AU plus a class for the absence of AU, resulting in a total of six classes. Using multiple kernels instead of a single one can improve classifier performance, with a common approach being to use a convex linear combination of basis kernels. In this work, Gaussian and interaction kernels are integrated into the multiple kernel framework,

with different data sources associated with each kernel.
In [39], a solution with a more complicated architecture was proposed that led to slightly better results. A DRML network has been constructed, with a newly proposed region layer, for multi-label AU detection. A region layer captures local appearance changes for different facial regions. Such regional information has shown to provide unique cues to recognize AUs and holistic expressions. For the recognition of certain AUs, this approach has proven particularly effective.

3.2 Methods In Facial Emotion Recognition

This section provides a description of methods related to FER.

3.2.1 Feature Extraction

With regards to image preprocessing and subsequent face detection, identical models are employed for both FAUs detection and FER. In terms of feature extraction, the features outlined in section 2.2.1 are once again utilized, however, a combination of traditional methods and deep neural networks has progressively been employed.

3.2.2 Models

This article [40] proposed one of the best approaches that exploits traditional techniques for recognizing facial expressions by combining Gabor and Local Binary Pattern (LBP) features. The Gabor filter is used to extract facial features, and LBP is used to encode the texture of the face. The proposed method first involves several steps to preprocess and extract facial features from raw expression images; the images are normalized to reduce any variations in lighting, pose, and other factors. Next, two types of features, Gabor and LBP, are extracted from the preprocessed images. Dimensionality reduction is then performed using PCA to reduce the number of features and minimize redundancy. The resulting feature vectors from Gabor and LBP are then fused together and optimized to improve the overall performance of the model. Finally, a SVM classifier is trained on the fused feature vector to recognize facial expressions. The experimental results show that the proposed method outperforms other state-of-the-art methods of those years on

the CK+ dataset. In fig. 3.1 the flowchart of proposed model.

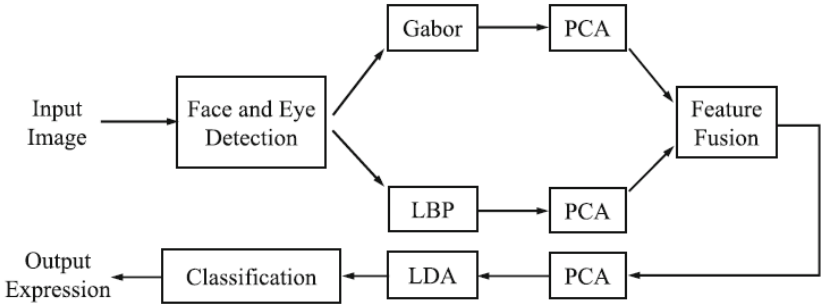


Figure 3.1: Flowchart of proposed model[40]

In [41] is reported for the first time this type of DL architecture for facial expression recognition in videos. The model is based on a frame attention mechanism that automatically learns which frames are important for accurately recognizing facial expressions over time. The model leverages both spatial and temporal information by processing frames through a convolutional neural network (CNN) and a recurrent neural network (RNN), respectively.
Lastly, [42] proposed a different approach with a single network that performs facial landmark detection and estimates both categorical and continuous emotions. The proposed method takes advantage of the facial points detected by a face-alignment network used for facial points detection, which are relevant for the emotion recognition task. Additionally, a set of steps are introduced to further enhance the performance of the model. These steps include a joint prediction of categorical and continuous emotions to increase the model's robustness to outliers in the dataset, an spatial attention mechanism that focuses on relevant regions of the face for affect estimation, a student-teacher training framework called knowledge distillation that smooths labels learned by the network, and a customized loss function designed to optimize CCC metric. Fig. 3.2 shows an overview of the architecture of their proposal.

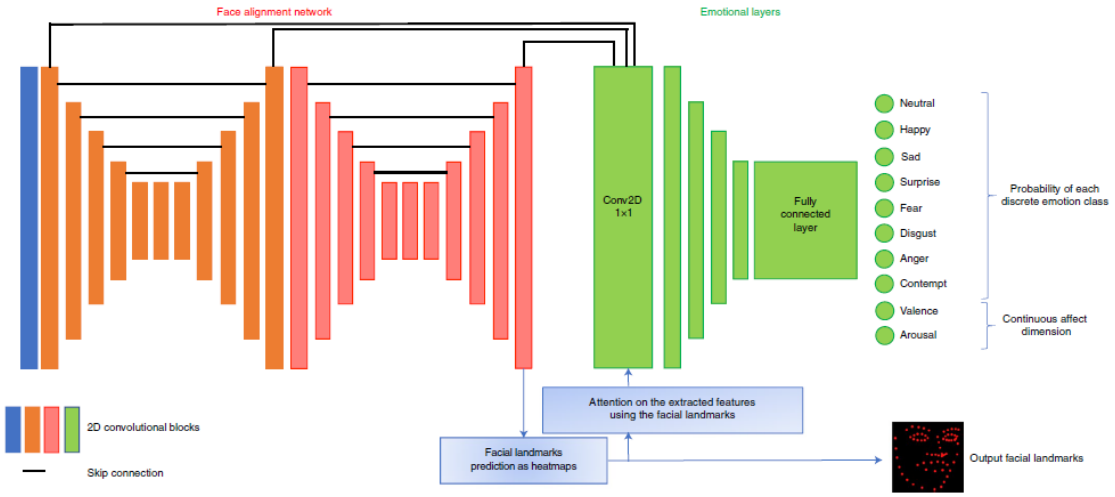


Figure 3.2: Flowchart of proposed model[40]

3.2.3 Emotion Recognition From Near-Infrared Videos

To overcome the problem of brightness, in [43] the authors developed a novel approach to dynamic facial expression recognition using NIR video sequences. NIR imaging combined with LBP features offer an illumination-invariant description of face video sequences. Specifically, a dataset of NIR videos, containing facial expressions of six basic emotions, is used: happiness, sadness, surprise, anger, disgust, and fear. A component-based facial features are introduced to combine geometric and appearance information, providing an effective representation of facial expressions. Overall, the paper demonstrates that NIR videos can be a promising source of information for FER in situations where ambient light may interfere with facial expression detection because SVM demonstrate robust results, offering a baseline for future research on NIR-based facial expression recognition.. Fig. 3.3 shows that NIR images are more robust to changing brightness and allow the network to focus on key aspects of the face.

Figure 3.3: VIS (top) and NIR (bottom) images in different light conditions[43]

3.3 Methods In Multimodal Emotion Recognition

In the realm of multimodal analysis of emotions, there remain numerous unresolved issues, as described in section 2.6, but the potential is considerable. Although detecting emotion is a common objective, there are variations in the approaches taken and the subtasks investigated, such as whether emotions should be classified in a continuous dimensional format or with the use of discrete categories. Additionally, there is significant variation in the number of emotional categories to classify and the dataset size, making it difficult to compare results.
This section provides a description of all different multimodal approches.

Facial Expression and Textual feature
In the field of text-based emotion recognition, the focus is on the emotions that prompt individuals to use certain words at specific times. Humans possess a certain degree of ability to comprehend emotions from text, which motivates the development of computers that can do the same. Nonetheless, textual interpretation is a complicated task for both humans and computers, given the absence of contextual

information, the presence of sarcasm, and the relationship between the author and reader. In a 2020 study, [44] explored the possibility of combining facial emotion recognition with text to classify the facial images of characters in a Korean TV series into seven emotions: anger, disgust, fear, happiness, neutral, sadness, and surprise. A multimodal deep learning model was created, using facial images and text descriptions of the situation as input values. The findings of the experiment revealed an increase in F1-score for 5 of 7 emotions when utilizing text descriptions of the characters and facial expressions, as opposed to a unimodal approach that only used facial expressions.

Facial Expression and EEG

In [45], a multimodal approach to emotion recognition was proposed for developing an HRI (human-robot interaction) system with low disharmony. The authors conducted a multimodal experiment that combined facial expressions and EEG. EEG data were obtained using an electrode cap in a laboratory setting, while facial expressions were captured using a camera with elicitation provided by a video. The data were then self-labeled by the subjects and passed through separate classifiers for facial and EEG. The Monte Carlo method was employed to combine the facial expressions and EEG results for the multimodal experiment. The model was able to classify four emotions with 83.33 % accuracy, which was an improvement compared to the unimodal approaches.

Facial Expression and EDA, Heart Rate, Respiration

In 2017, [46] proposed a multimodal approach combining facial expressions with several physiological signals to improve the recognition rate of emotion compared to a unimodal approach solely using facial expressions. The study used data including facial expressions, EDA, heart rate, and respiration. Facial features were extracted using AFFDEX, a facial expression analysis toolkit, while a total of 130 features were extracted from physiological signals, including time, minima, maxima, frequency, statistical, and spectral features. Late fusion was used to fuse the features, where a simple concatenation between the vectors was applied. The results showed an increase in recognition of both valence and arousal compared to a unimodal approach using facial expressions.

Facial Expression, Acoustic feature and Bio-signal

In [47] a model is proposed that utilizes LSTM networks with a self-attention mechanism to capture complex temporal dependencies within the feature sequences, including bio-signal features such as Electrocardiogram (ECG), Respiration (RESP), and heart rate (BPM). The architecture of the model comprises three modules: the LSTM networks with self-attention module for acoustic and visual features, the LSTM module for bio-signal features, and the late fusion module. Firstly, are fed the eGeMAPS feature and the VGGface feature separately to the LSTM network with self-attention module to obtain predictions from both audio and visual modalities. Then, the bio-signal features are concatenated and send to the LSTM module to obtain the predictions of the bio-signal modality. Finally, the predictions from all modalities are concatenated and sent to the late fusion module for regression.

The self-attention mechanism is used to transform input sequences into high-level representations, capturing relationships across the sequences. In addition, the visual feature and acoustic sequence are sent to the self-attention module to capture contextual information. The outputs of the self-attention modules are then sent to the LSTM to model complex temporal dependencies within the sequence.

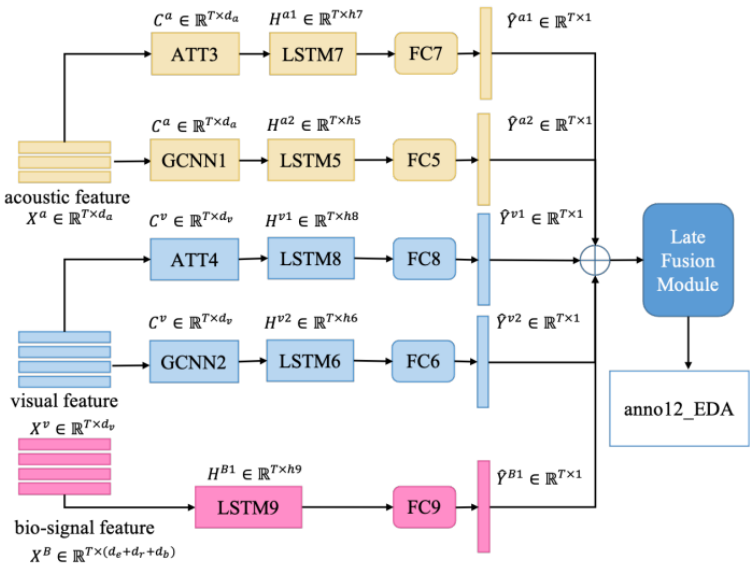


Figure 3.4: Overview of architecture used by[47]

Facial Expression, Acoustic and Textual features

The authors of [48] have developed ViPER (Video-based Perceiver for Emotion Recognition), a multimodal architecture designed to recognize emotions from videos. ViPER employs an attention-based and modality-agnostic late fusion strategy, which receives both the visual component, consisting of images corresponding to video frames, and the acoustic component, which is the audio recording associated with the video. The visual component is utilized to extract various features, including transformer-based visual embeddings, Facial Action Units (FAUs), and frame captions. A new modality is introduced through frame captioning, which involves providing a textual description of the video frames. This description is then encoded using a state-of-the-art contextualized embedding model.

The original and augmented data are used to generate latent features that include visual features such as Vision Transformer and FAUs, augmented textual features, and acoustic features such as x-vector representations of the audio waveforms. These features are combined using a late fusion network, which relies on a modality-agnostic approach, making the network adaptable to various modality combinations.

Chapter 4

Dataset, Experiment and result

4.1 Dataset

A difficult problem in computer vision is the application of affective computing in real-world scenarios. The current availability of annotated facial expression databases for naturalistic settings is limited and primarily focused on discrete emotions that can be classified into seven basic categories, including happiness, sadness, anger, disgust, fear, surprise, and neutral. In contrast, the continuous dimensional model that considers valence and arousal has very few annotated facial databases. In the following, the most widely used databases in FER and the one chosen to train our models are presented.

4.1.1 Existing Dataset

Overview on the most common dataset used to train and evaluate FER models.

CK+

CK+ (Cohn-Kanade+) is a facial expression database widely used in computer vision and affective computing research. It consists of over 5000 images of 123 subjects displaying six basic facial expressions (anger, disgust, fear, happiness,

sadness, and surprise) and a neutral expression [49]. Each expression is shown in a sequence of increasing intensity levels, allowing researchers to study the temporal dynamics of facial expressions. The database also provides manual annotations of facial landmarks and action units, which enable the development and evaluation of algorithms for facial expression recognition, facial landmark detection, and facial action unit detection. This dataset does not have the continuous annotations (e.g., valence and arousal).

KDEF

KDEF (Karolinska Directed Emotional Faces) is a database of facial expressions that contains images of 70 individuals (35 women and 35 men) expressing six basic emotions: happiness, anger, fear, disgusted, surprised, and sad [50]. The images are taken under controlled conditions with standardized lighting and background. Each individual displays the six emotions at three different intensity levels, making a total of 1260 images. This dataset does not have the continuous annotations (e.g., valence and arousal).

RaFD

The Radboud Faces Database (RaFD) is a facial expression database that was developed at Radboud University Nijmegen, Netherlands [51]. It consists of 67 participants displaying eight different facial expressions: neutral, happiness, sadness, anger, fear, disgust, surprise, and contempt. Each participant's expressions are captured through a high-quality video camera. The database includes both frontal and 3/4th view of the face with naturalistic and uncontrolled lighting conditions. This dataset does not have the continuous annotations (e.g., valence and arousal).

Aff-Wild2

AFF-Wild2 is a widely used facial expression dataset that was created to advance research in automatic affective computing. It contains around 540 videos with spontaneous facial expressions, captured in real-world scenarios, and over 200,000 frames [11]. The dataset features diverse individuals from different cultures and ages expressing various emotions, including happiness, sadness, anger, surprise, fear, and disgust. AFF-Wild2 also includes some of the most challenging situations,

such as low-quality videos, occlusion, and temporal misalignment. To ensure its reliability, the dataset has been annotated by multiple experts, and it provides detailed annotations, including arousal and valence values for each facial expression.

Oulu-Casia Nir Vis

Oulu-Casia NIR-VIS dataset is a facial expression database that includes both near-infrared (NIR) and visible light (VIS) image modalities [52]. This dataset is designed to address the limitations of previous databases by providing a high-quality, diverse set of facial expressions under different illumination conditions, head poses, and subjects. The Oulu-Casia NIR-VIS database contains a total of 4800 images of 80 subjects, each of whom displays six facial expressions: anger, disgust, fear, happiness, sadness, and surprise. The images are captured using both NIR and VIS cameras, with varying illumination conditions and head poses. The Oulu-Casia NIR-VIS dataset has been widely used in research on facial expression recognition, cross-modal learning, and face anti-spoofing. This dataset does not have the continuous annotations (e.g., valence and arousal).

4.2 Muse Challenge Dataset

In this section, the MuSe-Stress sub-challenge dataset is presented as the selected dataset for the development of the model. This dataset was chosen due to its emphasis on the continuous analysis of emotions through arousal and valence measurements [12]. In addition, this dataset offers a diverse set of features that can be utilized in future studies to compare the effectiveness of our model with that of a multimodal approach. This dataset was utilized to train a model that could extract FAUs from the raw images and use them as input features for a second model. This approach was only feasible due to the availability of FAU features in the dataset, which are not commonly found combined with labels of arousal and valence. Unlike other datasets that utilize images from the Internet with assigned arousal and valence values, this dataset offers real-life videos and images of subjects in a more authentic stress-inducing environment.

The regression task utilizes the Ulm-Trier Social Stress Test dataset (Ulm-TSST),

which features individuals in a stress-inducing scenario following the Trier Social Stress Test (TSST)[53]. In addition to audio, video, and textual features, the Ulm-TSST dataset includes four biological signals (EDA, ECG, RESP, and BPM) and FAUs. In this protocol, 69 participants give a five-minute free speech presentation in a simulated job interview setting, supervised by two interviewers who do not interact with them. The data has been annotated by three raters for valence and arousal, with valence annotated using the Rater Aligned Annotation Weighting (RAAW) method to fuse the ratings of the three raters[12].
In this project, it was decided not to totally follow the dictates of the challenge and no multimodal models were developed. This decision was made because our model should not need sensors placed on the patient. In this way, our model can be more accessible and easier to use. Fig. 4.1 and fig. 4.2 display, respectively, the environment and an extracted face of MuSe-Stress sub-challenge dataset [12].

Figure 4.1: Environment of the video

Figure 4.2: Extracted face

4.3 Machine Learning Model Architecture

In this section, the preprocessing steps taken to prepare the MuSe dataset and the various models trained on this data are presented. The final part presents the performance obtained from the various machine learning models tested with the different preprocessing steps.

4.3.1 Preprocessing

In order to extract relevant features from facial images, preprocessing was performed on each frame of the video data. Specifically, two popular techniques in computer vision were utilized: HOGs and facial landmarks detection. For the HOG features, the pipeline described by Py-feat was followed to extract HOG descriptors from each frame of the video data. The pipeline includes the following steps:

- extraction of the bounding box containing the subject's face to reduce the image information to be processed;
- calculation of 2-D landmarks to identify the cardinal points of the face;
- face alignment in order to reduce differences due to different pose angles of the subject;
- extraction of the convex hull containing the realigned subject's face;
- calculation of hogs as described in section 2.2.1.

These HOG descriptors capture the distribution of edge orientations in each image. Since feature hogs were too heavy to process for simple machine learning models, a PCA was performed to reduce the number of features and obtain a vector containing 1195 elements.
In addition to HOG features, also facial landmarks were computed detection to extract information about specific points on each face. The MediaPipe library was used to detect and extract the locations of 478 3-D facial landmarks on each convex hull of frame of the video data. These landmarks can provide information about facial expression, gaze direction, and other important cues that are relevant to emotional analysis. Fig. 4.3 and fig. 4.4 show the effect of preprocessing and feature extraction steps. Together, these preprocessing techniques allowed us to extract rich and informative features from each frame of the video data, which were then used as input to our emotion recognition models. Our models were trained with a vector (2629 elements) containing the x, y, and z coordinates of the 478 face landmarks (1434 elements) concatenated with the features obtained in output to the PCA performed on the hogs (1195 elements).

Figure 4.3: Raw face and face after extraction and realignment

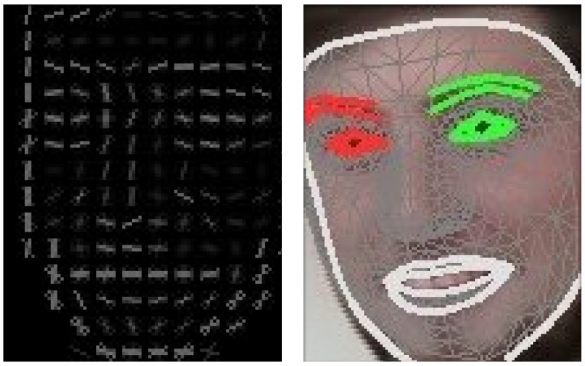


Figure 4.4: Hogs and 3D face landmarks extracted

4.3.2 FAUs Detection

An alternative solution was initially tried in which FAUs were extracted from video recordings of participants using FACS. To apply FACS, annotated frames of MuSe Dataset were employed; the intensities of 20 AUs between values 0 and 1 were provided for each frame. The AUs intensity were used as labels to train AU detection models. Machine learning models (e.g. SVM, MLP) were trained to automatically recognize them from video frames and also proved to be very reliable. The idea was to use the outputs of these models as features for another ML models

designed to recognize the emotional states of the participants. These models were trained to predict both arousal and valence based on the facial expressions of the participants. It was assumed that the FAUs would be a suitable indicator for recognizing arousal and valence. Through the concatenation of two models, it was expected to achieve good level of accuracy in predicting the emotional states of the participants. Unfortunately, as will be described in the results section 4.3.4, the ML models for arousal and valence recognition from AU performed very poorly. For this reason, this idea was abandoned and a different preprocessing was carried out. Fig. 4.5 displays schematic pipeline of proposed model for arousal and valence regression using FAUs.

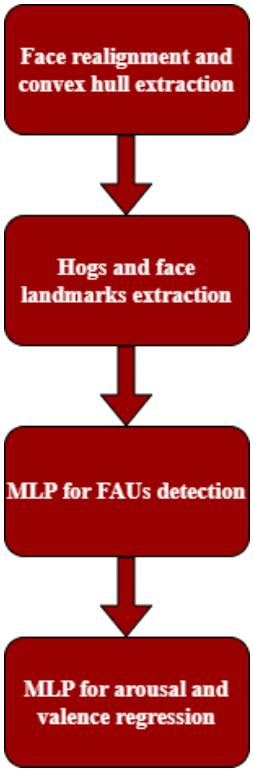


Figure 4.5: Pipeline for arousal and valence regression using FAUs

4.3.3 Shallow Learning

After the preprocessing step, two ML algorithms, Support Vector Machines (SVM) and Multilayer Perceptron (MLP), were trained to predict the valence and arousal of the participants. Before training the models, a grid search was performed to find the optimal hyperparameters for each algorithm. The grid search consisted of testing multiple combinations of hyperparameters, such as the loss function and penalty parameter for SVM, and the number of hidden layers, neurons and learning rate for MLP. Regarding the SVR models, a comprehensive grid search was not conducted as the obtained performance was consistently inferior to that of the MLP model. The optimal model, whose results will be presented, was obtained using the following hyperparameters:

- C (penalty parameter) = 10;
- Loss function = squared epsilon insensitive;
- Epsilon (epsilon-tube within which no penalty is associated in the training loss function) = 0.0

Regarding the MLP models, a more comprehensive grid search was performed, considering the good performance achieved. The optimal model was obtained using the following hyperparameters:

- Activation function = ReLU;
- Hidden Layer sizes = (3000, 5000, 3000, 1500, 500, 50);
- Learning Rate init = 0.0001

The performance of the models was evaluated using 3 cross-validation. The grid search and cross-validation were performed on a subset of the dataset, and the best hyperparameters were selected based on the highest accuracy and R^2 score. The selected hyperparameters were then used to train the models on the entire dataset, and the performance was evaluated on a separate test set. Fig. 4.6 displays schematic pipeline of proposed model. The results are shown in the section 4.3.4.

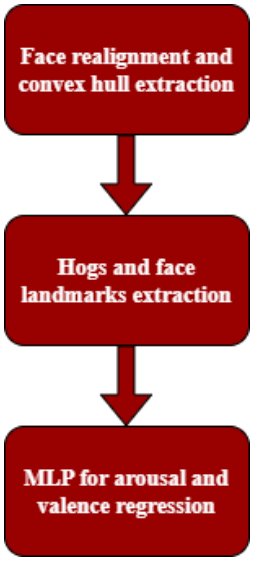


Figure 4.6: Pipeline for arousal and valence regression using ML techniques

4.3.4 Results

In this section all performance obtained for models described in the previous section are reported.

Performance for best SVR model:
Table 4.1 shows that the best SVR model achieved decent performance, with a CCC of 0.71 for Arousal and 0.77 for Valence on the training set. However, there was a slight drop in performance on the test set, resulting in a CCC of 0.68 for Arousal and 0.74 for Valence. The SAGR values remained roughly constant between the training and test sets, indicating that the drop of the other metrics was primarily due to an inaccurate estimation of the intensity of Arousal and Valence, but not their sign.

NETWORK	TASK	CCC		SAGR		RMSE	
		Training	Test	Training	Test	Training	Test
SVR	Arousal	0.707	0.680	0.793	0.784	0.227	0.237
	Valence	0.774	0.742	0.799	0.785	0.125	0.133

Table 4.1: Evaluation of best SVR model for Arousal and Valence regression

Grid search for MLP:

Tables 4.2 and 4.3 present the R^2 values obtained for the various hyperparameter combinations tested during the grid search. The results indicate that the best scores were obtained with lower learning rates, with ReLU activation function and models with a greater number of hidden layers. The highest R^2 value obtained was 0.73, achieved by the best-performing model.

Activation Function	Hidden Layers	Learning Rate	R^2
ReLU	3000,4000,6000,4000,2000,1000,100	0.1	-5.46
ReLU	3000,4000,6000,4000,2000,1000,100	0.01	-0.46
ReLU	3000,4000,6000,4000,2000,1000,100	0.001	0.61
ReLU	3000,4000,6000,4000,2000,1000,100	0.0001	0.72
ReLU	3000,5000,3000,1500,500,50	0.1	-0.00
ReLU	3000,5000,3000,1500,500,50	0.01	0.60
ReLU	3000,5000,3000,1500,500,50	0.001	0.63
ReLU	3000,5000,3000,1500,500,50	0.0001	**0.73**
ReLU	1000,500,50	0.1	-0.0
ReLU	1000,500,50	0.01	0.59
ReLU	1000,500,50	0.001	0.65
ReLU	1000,500,50	0.0001	0.69
ReLU	500	0.1	0.21
ReLU	500	0.01	0.55
ReLU	500	0.001	0.62
ReLU	500	0.0001	0.64
ReLU	2000,1000,500,50	0.1	-0.01
ReLU	2000,1000,500,50	0.01	0.53
ReLU	2000,1000,500,50	0.001	0.64
ReLU	2000,1000,500,50	0.0001	0.69

Table 4.2: Grid search to find the optimal combination of hidden layers and learning rate

Activation Function	Hidden Layers	Learning Rate	R^2
Tanh	3000,5000,3000,1500,500,50	0.001	-0.01
Tanh	3000,5000,3000,1500,500,50	0.0001	0.61
Tanh	3000,5000,3000,1500,500,50	0.00001	0.63
Tanh	2000,1000,500,50	0.001	**0.64**
Tanh	2000,1000,500,50	0.0001	0.61
Tanh	2000,1000,500,50	0.00001	0.60

Table 4.3: Second grid search to try another activation function with the best combinations of the previous grid search

Performance for best MLP model:

Table 4.4 displays the metrics values obtained for the best MLP model found through the previous grid search. High values for CCC are achieved on the training set, with 0.98 for Arousal and 0.99 for Valence. However, a significant drop is observed on the test set, resulting in a CCC of 0.82 for Arousal and 0.87 for Valence. Similarly, other metrics exhibit a similar trend.

TASK	CCC		SAGR		RMSE	
	Training	Test	Training	Test	Training	Test
Arousal	0.983	0.821	0.975	0.839	0.061	0.188
Valence	0.988	0.872	0.973	0.857	0.031	0.098

Table 4.4: Evaluation of best MLP model for Arousal and Valence regression

Performance for best MLP trained for FAUs detection:

Table 4.5 displays the R^2, MSE, and MAE values obtained by the best MLP model for recognizing FAUs from preprocessed images. This model achieves excellent performance on both the training and test sets, particularly with a MAE of 0.017 on the training set and 0.024 on the test set.
Hyperparameters of MLP:

- Activation function : Logistic;
- Hidden layers: 2000,1000,500,50;
- Learning rate init: 0.001

R^2		MSE		MAE	
Training	Test	Training	Test	Training	Test
0.939	0.873	0.001	0.003	0.017	0.024

Table 4.5: Evaluation of best MLP model for FAUs detection

Performance of MLP model for Arousal and Valence regression trained with the outputs of best model for FAUs detection:
Table 4.6 displays the metric values obtained for the models trained using the FAUs extracted by the best model for FAUs detection. Both models exhibit similar performance, slightly favoring the second model, with a CCC of 0.75 for Arousal and 0.77 for Valence for both training and test. Neither model shows a drop in metrics between the training and test sets.
Model 1 hyperparameters:

- Activation function : ReLU;
- Hidden layers: 2000,1000,500,50;
- Learning rate init: 0.001

Model 2 hyperparameters:

- Activation function : ReLU;
- Hidden layers: 2000,1000,500,50;
- Learning rate init: 0.0001

NETWORKS	TASK	CCC		SAGR		RMSE	
		Training	Test	Training	Test	Training	Test
Model 1	Arousal	0.711	0.713	0.793	0.795	0.226	0.225
	Valence	0.751	0.749	0.780	0.776	0.130	0.130
Model 2	Arousal	0.745	0.746	0.805	0.801	0.216	0.217
	Valence	0.767	0.766	0.787	0.783	0.126	0.127

Table 4.6: Evaluation of best MLP model for Arousal and Valence regression trained with FAUs

4.3.5 Evaluation Of Results

In this section, is presented an analysis of the results obtained from various ML models with different preprocessing approaches. Upon comparison of the models, it is evident that the MLP outperforms in every metric when compared to the SVR. The MLP, when trained with the specific preprocessing techniques for the task at hand, emerged as the best model in all simulations. However, the only advantage of the SVR over the MLP is its smaller drop between training and test performance, despite remaining inferior overall.

For the MLP model, a first grid search was conducted to determine the best combinations of the number of hidden layers, number of neurons, and learning rate while maintaining the activation function fixed. The R^2 values showed a general increase in performance for smaller learning rates and for combinations with more hidden layers. However, the grids with better R^2 values required greater computation time to be trained. Subsequently, another grid search was performed to test different activation functions on the combinations of layers and neurons that had produced better results in the previous grid search. Lower learning rates were also tested, as they generally led to improved performance. However, no better combinations were found, and therefore, the hyperparameters obtained in the first grid search were considered optimal.

The performance of the best MLP model exhibited a significant impairment on the test set, although it remained relatively satisfactory.

Furthermore, the best model trained for FAUs recognition demonstrated a negligible error on both the training and test sets. However, the high performance of the FAUs recognition model did not translate to improved performance on the cascade-trained MLP model. Consequently, this solution was abandoned. Nevertheless, the MLP model did not exhibit a performance impairment between the training and test sets in this scenario.

4.4 Deep Learning Model Architecture

In this section, the preprocessing steps taken to prepare the MuSe dataset and the various deep learning models trained on this data are presented. The final part presents the performance obtained from the different architectures.

4.4.1 Preprocessing

No significant preprocessing of the data was conducted for the deep learning models. The DL models were primarily trained to enable a valid comparison and evaluation of the performance against that of the ML models. Thus, a simple preprocessing step was undertaken, which involved resizing the subject's face to 224x224 pixels and normalizing the image using the preprocessing functions provided by Keras for the specific neural networks.

4.4.2 Neural Networks

Training deep learning models from scratch can be a challenging and time-consuming task. In transfer learning, the pre-trained models, such as ResNet50, ResNet152, VGG19, and MobileNet, are employed as a starting point for training on a target dataset with similar characteristics. The pre-trained models used in this study are all deep convolutional neural networks provided by the Keras library with different architectures and varying numbers of layers. The process of fine-tuning involves modifying the pre-trained models' weights and adjusting them to fit the new target dataset. On the model output, layers with randomly initialised weights were added to fit the model to our desired output. To achieve this, an initial average pooling layer was incorporated to reduce the dimensionality and extract relevant information from the network output. This was followed by two dense layers, which were interspersed with a dropout layer to obtain a regression of arousal and valence. Such adaptations were necessary to tailor the models to our specific needs. Fig. 4.7 shows the pipeline followed to train the different DL models. Section 4.4.3 shows the performance obtained by all models trained to have a benchmark for ML model.

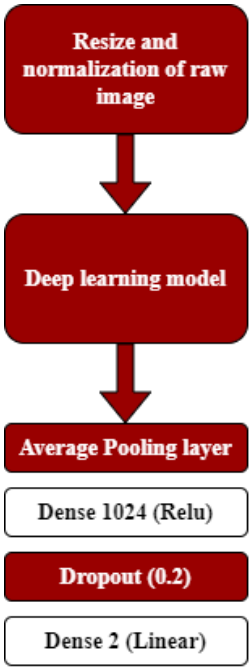


Figure 4.7: Pipeline for arousal and valence regression using DL techniques

4.4.3 Results

In table 4.7 all the performance results are reported, obtained for models described in the previous section. ResNet50 and ResNet152 achieved very similar performance with a CCC of 0.98 for arousal on the training set and 0.84 on the test set, while for valence, the CCC was 0.97 and 0.87 respectively for the training and test sets.

NETWORKS	TASK	CCC		SAGR		RMSE	
		Training	Test	Training	Test	Training	Test
ResNet50	Arousal	0.978	0.840	0.952	0.854	0.068	0.180
	Valence	0.967	0.869	0.913	0.859	0.053	0.103
MobileNetV2	Arousal	0.918	0.786	0.900	0.834	0.125	0.202
	Valence	0.898	0.808	0.853	0.824	0.088	0.119
VGG19	Arousal	0.769	0.736	0.835	0.814	0.194	0.210
	Valence	0.789	0.756	0.820	0.805	0.120	0.128
ResNet152	Arousal	**0.981**	**0.843**	**0.961**	**0.855**	**0.063**	**0.177**
	Valence	**0.968**	**0.871**	**0.916**	**0.862**	**0.052**	**0.102**

Table 4.7: Evaluation of DL models for Arousal and Valence regression

4.4.4 Evaluation Of Results

Regarding the deep learning models that were trained, it is evident that the ResNets exhibit superior performance at the expenses of larger architectural complexity. Although there is a marginal discrepancy in the results of the two tested ResNet models, it should be noted that the ResNet152, with its more elaborate structure, is slower in both training and prediction phases. All other networks achieve inferior performance with respect to ResNet. It is possible to observe that the best deep networks exhibit better performance than the best MLP model on both the training and test sets. Nevertheless, the difference remains quite limited, confirming the validity of the proposed ML solution and suggesting that both solutions can be used.

Chapter 5

Discussion and Future Works

5.1 Discussion

In the last decade, the field of emotion recognition from video has been increasingly explored and applied in various fields. This area is vast and presents several types of applications, which is why there are still many solutions that need to be explored. Recently, numerous studies have focused on multimodal emotion recognition, but this leads to a lower applicability of the system. Although using multiple recognition modalities together improves accuracy as demonstrated in the literature, it requires more complicated sensor systems. For these reasons, this thesis focuses on analyzing a subject's emotional state based solely on the frames of a video.

The first major problem encountered was choosing the dataset to train the model. There are FER datasets available online, but each has limitations, especially those that include arousal and valence recognition, which are limited. The chosen dataset, Muse stress sub-challenge, has numerous features and arousal and valence labels. However, most frames have arousal and valence values that are very similar and close to zero. This dataset composition could lead to poor generalization capabilities of the system, making it unusable for situations where the subject may present arousal and valence values closer to the extremes of the range -1 and 1. Unfortunately,

these tests could not be confirmed as the dataset has few subjects with extreme values, not sufficient to allow for a robust analysis of the results.
Another major problem faced was choosing the features and model. After analyzing the features and shallow learning models that have been most used in the literature, we proceeded to analyze the best possible combinations with 3-fold cross-validation, trying to train the models using the features individually and collectively. This analysis was time-consuming because the combinations to try were numerous, and some of them did not yield the expected results. Shallow learning models were chosen for analysis as they are simpler, maintain greater explainability, and require less computational time in many cases. Regarding the choice of features, it has been demonstrated that the combination of 3D face landmarks and HOGs is the best choice for training a shallow model. This is likely due to the fact that this combination of features provides the model with simplified yet informative information to evaluate the subject's emotional state. However, extracting these features for each frame to be analyzed carries a computational cost that should not be underestimated if a real-time system is desired.
The model obtained with the best hyperparameter combination, despite a drop in the test set, still yields good results in all metrics analyzed (CCC, SAGR, RMSE). Although there is a certain degree of overfitting, the model performs well enough to be usable. Unfortunately, comparing it with other models trained on the same dataset is challenging and not very informative since they are all multimodal and far more complex than the model proposed in this thesis.
Finally, state-of-the-art deep models were trained to be used as benchmarks for our best shallow model obtained. The performance obtained from these models showed that shallow learning models slightly underperform. It can be inferred that shallow models can be used as effectively as deep models, given that they are preceded by suitable preprocessing, achieving similar performances with theoretically lighter structures.

5.2 Future Works

The area of emotional recognition has been expanding, with numerous studies in recent years, however, there remains a need for additional efforts in this field. In

the following recommendations are provided for advancing the research conducted in this thesis or for enhancing it.

Regarding the research presented in this thesis, it is recommended to develop a model that considers the temporal relationship between frames to provide a more comprehensive analysis. Possible models that could be implemented include LSTM or frame attention networks. Alternatively, a simpler solution could be to calculate a moving average of the outputs generated over consecutive frames, taking into account the gradual changes in emotional state between frames.

Bibliography

[1] *The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology*. Posner, Russell, Peterson. 2005 (cit. on pp. 1, 4).

[2] *Affective neuroscience: The foundations of human and animal emotions*. New York: Oxford University Press, Panksepp. 1998 (cit. on pp. 1, 3).

[3] F. Abdat, C. Maaoui, and A. Pruski. «Human-Computer Interaction Using Emotion Recognition from Facial Expression». In: *2011 UKSim 5th European Symposium on Computer Modeling and Simulation*. 2011, pp. 196–201. DOI: 10.1109/EMS.2011.20 (cit. on p. 1).

[4] *An argument for basic emotions*. Cognition and Emotion 6.3-4, pp. 169–200. DOI: 10 . 1080 / 02699939208411068. 1992 (cit. on p. 3).

[5] *Multimodal Affect Recognition: Current Approaches and Challenges*. Al Osman, Hussein and Tiago Falk DOI: 10.5772/65683. Feb. 2017 (cit. on p. 4).

[6] *Facial expressions of emotion are not culturally universal*. Rachael E. Jack, Oliver G. B. Garrod, Hui Yu and Philippe G. Schyns. 2012 (cit. on pp. 4, 6).

[7] Rafael A. Calvo and Sidney D'Mello. «Affect Detection: An Interdisciplinary Review of Models, Methods, and Their Applications». In: *IEEE Transactions on Affective Computing* 1.1 (2010), pp. 18–37. DOI: 10.1109/T-AFFC.2010.1 (cit. on p. 4).

[8] *Un modello dimensionale delle emozioni: integrazione tra le neuroscienze dell'affettività, lo sviluppo cognitivo e la psicopatologia*. Basile. Aug. 2012 (cit. on p. 5).

[9] *Arousal/valence model*. James Russell and Lisa Feldman Barrett. URL: https://pin.it/5MVKbfn (cit. on p. 5).

[10] *A Directed Acyclic Graph Network Combined With CNN and LSTM for Remaining Useful Life Prediction*. Li, Jialin, Li, He. 2019 (cit. on p. 6).

[11] Dimitrios Kollias and Stefanos Zafeiriou. «Affect Analysis in-the-wild: Valence-Arousal, Expressions, Action Units and a Unified Framework». In: *arXiv preprint arXiv:2103.15792* (2021) (cit. on pp. 6, 39).

[12] *The MuSe 2022 Multimodal Sentiment Analysis Challenge: Humor, Emotional Reactions, and Stress*. Lukas Christ, Shahin Amiriparian, Alice Baird, Panagiotis Tzirakis, Alexander Kathan, Niklas Mueller, Lukas Stappen, Eva Messner, Andreas König, Alan Cowen, Erik Cambria, and Björn Schuller. 2022. Apr. 2022. URL: https://rafd.socsci.ru.nl/RaFD2/RaFD?p=main (cit. on pp. 6, 10, 40, 41).

[13] *A survey on facial emotion recognition techniques: A state-of-the-art literature review*. Canal, Felipe Zago. 2022. URL: https://www.sciencedirect.com/science/article/pii/S0020025521010136 (cit. on p. 7).

[14] T. Ojala, M. Pietikainen, and T. Maenpaa. «Multiresolution gray-scale and rotation invariant texture classification with local binary patterns». In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.7 (2002), pp. 971–987. DOI: 10.1109/TPAMI.2002.1017623 (cit. on p. 8).

[15] *Object Detection Using Local Binary Patterns*. Christos Kyrkou. 2017 (cit. on p. 9).

[16] *Deep Sparse Representation Classifier for facial recognition and detection system*. Cheng, Eric-Juwei et al. In: Pattern Recognition Letters 125, pp. 71–77. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec. 2019.03.006. 2019. URL: https://www.%20sciencedirect.%20com/%20science/article/%20pii/%20S0167865519300868. (cit. on p. 10).

[17] *Facial Action Coding System*. Ekman, P. and W. V. Friesen. 1978 (cit. on p. 10).

[18] Jacob Whitehill and Javier Movellan. «Automatic facial expression recognition for intelligent tutoring systems». In: *Proceedings of the CVPR Workshop on Human Communicative Behavior Analysis* (June 2008). DOI: 10.1109/CVPRW.2008.4563182 (cit. on p. 11).

[19] *Cos'è il machine learning?* Oracle. 2019. URL: https://www.oracle.com/it/artificial-intelligence/machine-learning/what-is-machine-learning/#:~:text=Il%5C%20Machine%5C%20Learning%5C%20(ML)%5C%20%5C%C3%5C%A8,che%5C%20imitano%5C%20l'intelligenza%5C%20umana. (cit. on p. 11).

[20] *The Effects of User Features on Twitter Hate Speech Detection*. Unsvag, Elise and Björn Gambäck. 2018 (cit. on p. 16).

[21] *Support Vector Machine Algorithm*. Javapoint. URL: https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm (cit. on p. 16).

[22] *Multi-layer Perceptron in TensorFlow*. Javapoint. URL: https://www.javatpoint.com/multi-layer-perceptron-in-tensorflow (cit. on p. 17).

[23] *DeepLearning series: Convolutional Neural Networks*. Cavaioni. 2018. URL: https://medium.com/machine-learning-bites/deeplearning-series-convolutional-neural-networks-a9c2f2ee1524 (cit. on pp. 19, 20).

[24] *Improving neural networks by preventing co-adaptation of feature detectors*. Hinton, Geoffrey E. et al. 2012 (cit. on p. 20).

[25] *Deep Residual Learning for Image Recognition*. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. 2015. URL: https://arxiv.org/abs/1512.03385 (cit. on p. 21).

[26] *Optimized Deep Convolutional Neural Networks forIdentification of Macular Diseases from OpticalCoherence Tomography Images*. Ji, Huang. 2019. URL: https://www.researchgate.net/publication/331364877_Optimized_Deep_Convolutional_Neural_Networks_for_Identification_of_Macular_Diseases_from_Optical_Coherence_Tomography_Images (cit. on p. 22).

[27] *Very Deep Convolutional Networks for Large-Scale Image Recognition*. Karen Simonyan, Andrew Zisserman. 2014. URL: https://arxiv.org/abs/1409.1556 (cit. on p. 22).

[28] *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam. 2017. URL: https://arxiv.org/abs/1704.04861 (cit. on p. 23).

[29] *LONG SHORT-TERM MEMORY*. Sepp Hochreiter, Jurgen Schmidhuber. In Neural Computation 9(8):1735. 1997. URL: https://www.bioinf.jku.at/publications/older/2604.pdf (cit. on p. 23).

[30] *Multi-modal Continuous Dimensional Emotion Recognition Using Recurrent Neural Network and Self-Attention Mechanism*. Licai Sun, Zheng Lian, Jianhua Tao, Bin Liu. 2020 (cit. on p. 23).

[31] *Continuous prediction of spontaneous affect from multiple cues and modalities in valence-arousal space*. IEEE Transactions on Affective Computing, vol. 2, no. 2, pp. 92–105. 2011 (cit. on p. 25).

[32] *An Emotion Recognition Model Based on Facial Recognition in Virtual Learning Environment*. Yang, D. et al. In: Procedia Computer Science 125, pp. 2–10. DOI: 10.1016/j.procs.2017.12.003. 2018 (cit. on p. 30).

[33] *Emotion Recognition from Facial Expressions using Hybrid Feature Descriptors*. Kalsum, Tehmina et al. In: IET Image Processing 12. DOI: 10.1049/iet-ipr.2017.0499. 2018 (cit. on p. 30).

[34] *Comparison of Viola-Jones Haar Cascade Classifier and Histogram of Oriented Gradients (HOG) for face detection*. Rahmad, Cahya et al. In: IOP Conference Series: Materials Science and Engineering 732, p. 012038. DOI: 10.1088/1757-899X/732/1/012038. 2020 (cit. on p. 30).

[35] *Preprocessing Techniques to Improve CNN based Face Recognition System*. Raghavan, Jayanthi and Majid Ahmadi In: pp. 1–20. DOI: 10.5121/csit.2021.110101 2021 (cit. on p. 30).

[36] *Py-Feat: Python Facial Expression Analysis Toolbox*. Jin Hyun Cheong, Eshin Jolly, Tiankang Xie, Sophie Byrne, Matthew Kenney Luke J. Chang. 2019 (cit. on p. 30).

[37] *BlazeFace: Sub-millisecond Neural Face Detection on Mobile GPUs*. Valentin Bazarevsky Yury Kartynnik Andrey Vakunov Karthik Raveendran Matthias Grundmann. Google Research 1600 Amphitheatre Pkwy, Mountain View, CA 94043, USA. 2020 (cit. on p. 30).

[38] *Facial Action Units Intensity Estimation by the Fusion of Features with Multi-kernel Support Vector Machine*. Zuheng Ming, Aurélie Bugeau, Jean-Luc Rouas, Takaaki Shochi. 11th IEEE International Conference on Automatic Face, Gesture Recognition Conference, and Workshops. 2015 (cit. on p. 30).

[39] *Deep Region and Multi-label Learning for Facial Action Unit Detection*. Kaili Zhao, Wen-Sheng Chu, Honggang Zhang, School of Comm., Info. Engineering, Beijing University of Posts, and Telecom., Beijing China Robotics Institute, Carnegie Mellon University, USA. 2016 (cit. on p. 31).

[40] *Facial Expression Recognition by Fusing Gabor and Local Binary Pattern Features*. Yuechuan Sun and Jun Yu. IN MMM 2017, Part II, LNCS 10133, pp. 209–220, 2017. DOI: 10.1007/978-3-319-51814-5 18. 2017 (cit. on pp. 31–33).

[41] *Frame attention networks for facial expression recognition in videos*. Debin Meng, Xiaojiang Peng, Kai Wang, Yu Qiaou. 2019 (cit. on p. 32).

[42] *Estimation of continuous valence and arousal levels from faces in naturalistic conditions*. Toisoul, A., Kossaifi, J., Bulat, A. et al. IN Nat Mach Intell DOI: https://doi.org/10.1038/s42256-020-00280-0. 2021 (cit. on p. 32).

[43] *Facial expression recognition from near-infrared videos*. Guoying Zhao, Xiaohua Huang, Matti Taini, Stan Z. Li, Matti Pietikäinen. 2011 (cit. on pp. 33, 34).

[44] *A Multi-modal Approach for Emotion Recognition of TV Drama Characters Using Image and Text*. Lee, Jung-Hoon, Hyun-Ju Kim and Yun-Gyung Cheong. In: pp. 420–424. DOI: 10 . 1109 /BigComp48618.2020.00-37. Feb. 2020 (cit. on p. 35).

[45] *A multimodal emotion recognition method based on facial expressions and electroencephalography.* Tan, Ying et al. In: Biomedical Signal Processing and Control 70, p. 103029. ISSN: 1746-8094. DOI: https://doi. org/10.1016/j.bspc.2021.103029. 2021. URL: https://www.sciencedirect.com/science/article/pii/S1746809421006261. (cit. on p. 35).

[46] *Emotion recognition with facial expressions and physiological signals.* Zhong, Boxuan et al. In: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–8. DOI: 10.1109/SSCI.2017.8285365. 2017 (cit. on p. 35).

[47] *Hybrid Mutimodal Fusion for Dimensional Emotion Recognition.* Ziyu Ma, Fuyan Ma, Bin Sun, Shutao Li. Oct. 2021. URL: https://arxiv.org/pdf/2110.08495.pdf (cit. on p. 36).

[48] *ViPER: Video-based Perceiver for Emotion Recognition.* Lorenzo Vaiani, Moreno La Quatra, Luca Cagliero, Paolo Garza. 2022. URL: https://dl.acm.org/doi/pdf/10.1145/3551876.3554806 (cit. on p. 37).

[49] *The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression.* Lucey, Cohn, Kanade et al. 2010. URL: https://www.jeffcohn.net/wp-content/uploads/2020/02/CVPR2010_CK2.pdf.pdf (cit. on p. 39).

[50] *The Karolinska Directed Emotional Faces.* Lundqvist, Flykt Öhman. 1998. URL: https://www.kdef.se/home/aboutKDEF.html (cit. on p. 39).

[51] *Presentation and validation of the Radboud Faces Database.* Langner, O., Dotsch, R., Bijlstra, G., Wigboldus, D.H.J., Hawk, S.T., van Knippenberg, A.
Cognition Emotion, 24(8), 1377—1388. DOI: 10.1080/02699930903485076. 2010. URL: https://rafd.socsci.ru.nl/RaFD2/RaFD?p=main (cit. on p. 39).

[52] *Facial expression recognition from near-infrared video sequences.* M. Taini, G. Zhao, S. Z. Li and M. Pietikainen, 19th International Conference on Pattern Recognition, Tampa, FL, USA, 2008, pp. 1-4, doi: 10.1109/ICPR.2008.4761697. 2008 (cit. on p. 40).

[53] *The 'Trier Social Stress Test'–a tool for investigating psychobiological stress responses in a laboratory setting.* Kirschbaum C, Pirke KM, Hellhammer DH.Neuropsychobiology;28(1-2):76-81. doi: 10.1159/000119004. PMID: 8255414¦ 1993 (cit. on p. 41).