New Artificial Intelligence techniques for Computer vision based medical diagnosis

Safaa El Morabit

Thèse de doctorat
Pour obtenir le grade de Docteur de
l'UNIVERSITÉ POLYTECHNIQUE HAUTS-DE-FRANCE
et l'INSA HAUTS-DE-FRANCE

Nouvelles techniques de l'intelligence artificielle pour le diagnostic médical basé sur la vision par ordinateur

Discipline, spécialité selon la liste des spécialités pour lesquelles l'Ecole Doctorale est accréditée :
Electronique- Acoustique et télécommunications

Présenté et soutenu par SAFAA EL MORABIT
Le 11/04/2023, à Valenciennes

Ecole doctorale : Sciences Pour l'Ingénieur (ED PHF)
Laboratoire : IEMN UMR CNRS 8520

JURY

Rapporteurs	
Mme Salima Ouadfel	Professeure, Université Abdelhamid Mehri Constantine 2
M. Yassine Ruichek	Professeur, Université Belfort Montbeliard
Examinateurs	
Mme Raja Elassali	Professeure, ENSA de Marrakech
Mme. Laila Chakour	Docteure, ingénieure, INSA HdF
M. Sébastien JACQUES	Maitre de conférence, Université de Tours
Direction de thèse	
Mme Atika RIVENQ	Professeure, INSA HdF
M. Abdelmalik Taleb-Ahmed	Professeur, UPHF
Président du jury	M. Yassine Ruichek

PhD Thesis
Submitted for the degree of Doctor of Philosophy from
UNIVERSITÉ POLYTECHNIQUE HAUTS-DE-FRANCE
and INSA HAUTS-DE-FRANCE

New Artificial Intelligence techniques for Computer vision based medical diagnosis

Subject: Electronics

Presented and defended by SAFAA EL MORABIT
On 11/04/2023, Valenciennes

Doctoral school: Sciences for Engineers (ED PHF)
Laboratory: IEMN – UMR CNRS 8520

JURY

Reviewers	
MME SALIMA OUADFEL	PROFESSOR, UNIVERSITY ABDELHAMID MEHRI CONSTANTINE 2
M. YASSINE RUICHEK	PROFESSOR, UNIVERSITY BELFORT MONTBELIARD
Examiners	
MME RAJA ELASSALI	PROFESSOR, ENSA MARRAKECH
MME. LAILA CHAKOUR	DOCTOR, ENGINEER, INSA HDF
M. SÉBASTIEN JACQUES	ASSOCIATE PROFESSOR, UNIVERSITY OF TOURS
Thesis directors	
MME ATIKA RIVENQ	PROFESSOR, INSA HDF
M. ABDELMALIK TALEB-AHMED	PROFESSOR, UPHF
Jury president	M. YASSINE RUICHEK

Abstract

The ability to feel pain is crucial for life, since it serves as an early warning system for potential harm to the body. The majority of pain evaluations rely on patient reports. Patients who are unable to express their own pain must instead rely on third-party reports of their suffering. Due to potential observer bias, pain reports may contain inaccuracies. In addition, it would be impossible for people to keep watch around the clock. In order to better manage pain, especially in noncommunicative patients, automatic pain detection technologies might be implemented to aid human caregivers and complement their service. Facial expressions are used by all observer-based pain assessment systems because they are a reliable indicator of pain and can be interpreted from a distance.

Taking into consideration that pain generally generates spontaneous facial behavior, these facial expressions could be used to detect the presence of pain. In this thesis, we analyze facial expressions of pain in order to address pain estimation. First, we present a thorough analysis of the problem by comparing numerous common CNN (Convolutional Neural Network) architectures, such as MobileNet, GoogleNet, ResNeXt-50, ResNet18, and DenseNet-161. We employ these networks in two unique modes: standalone and feature extraction. In standalone mode, models (i.e., networks) are utilized to directly estimate pain. In feature extractor mode, "values" from the middle layer are extracted and fed into classifiers like Support Vector Regression (SVR) and Random Forest Regression (RFR).

CNNs have achieved significant results in image classification and have achieved great success. The effectiveness of Transformers in computer vision has been demonstrated through recent studies. Transformer-based architectures were proposed in the second section of this thesis. Two distinct Transformer-based frameworks were presented to address two distinct pain issues: pain detection (pain vs no pain) and the distinction between genuine and posed pain. The innovative architecture for binary identification of facial pain is based on data-efficient image transformers (Deit). Two datasets, UNBC-McMaster shoulder pain and BioVid heat pain, were used to fine-tune and assess the trained model. The suggested architecture is built on Vision Transformers for the detection of genuine and simulated pain from facial expressions (ViT). To distinguish between Genuine and Posed Pain, the model must pay particular attention to the subtle changes in facial expressions over time. The employed approach takes into account the sequential aspect and captures the variations in facial expressions. Experiments on the publicly accessible BioVid Heat Pain Database demonstrate the efficacy of our strategy.

Keywords: facial expression, pain estimation, CNN (Convolutional Neural Network), transformers, pain detection, Genuine and Posed Pain

Résumé

La capacité à ressentir la douleur est cruciale pour la vie, car elle sert de système d'alerte précoce en cas de dommages potentiels pour le corps. La majorité des évaluations de la douleur reposent sur les rapports des patients. En revanche, les patients incapables d'exprimer leur douleur doivent plutôt se fier aux rapports de tierces personnes sur leur souffrance. En raison des biais potentiel de l'observateur, les rapports sur la douleur peuvent contenir des inexactitudes. En outre, il serait impossible de surveiller les patients 24 heures sur 24. Afin de mieux gérer la douleur, notamment chez les patients avec des difficultés de communication, des techniques de détection automatique de la douleur pourraient être mises en œuvre pour aider les soignants et compléter leur service. Les expressions faciales sont utilisées par la plupart des systèmes d'évaluation de la douleur basés sur l'observation, car elles constituent un indicateur fiable de la douleur et peuvent être interprétées à distance.

En considérant que la douleur génère généralement un comportement facial spontané, les expressions faciales pourraient être utilisées pour détecter la présence de la douleur. Dans cette thèse, nous analysons les expressions faciales de la douleur afin d'aborder l'estimation de la douleur. Tout d'abord, nous présentons une analyse approfondie du problème en comparant de nombreuses architectures CNN (réseau de neurones convolutifs) courantes, telles que MobileNet, GoogleNet, ResNeXt-50, ResNet18 et DenseNet-161. Nous utilisons ces réseaux dans deux modes uniques : autonome et extraction de caractéristiques. En mode autonome, les modèles (c'est-à-dire les réseaux) sont utilisés pour estimer directement la douleur. En mode extracteur de caractéristiques, les "valeurs" de la couche intermédiaire sont extraites et introduites dans des classificateurs tels que la régression à vecteur de support (SVR) et la régression à forêts d'arbres décisionnels (RFR).

Les CNN ont obtenu des résultats significatifs dans la classification d'images et ont connu un grand succès. Plus récemment, l'efficacité des Transformers en vision par ordinateur a été démontrée par plusieurs études. Des architectures basées sur les Transformers ont été proposées dans la deuxième section de cette thèse. Ces deux architectures distinctes ont été présentées pour répondre à deux problèmes distincts liés à la douleur : la détection de la douleur (douleur vs absence de douleur) et la distinction entre la douleur authentique et la douleur simulée. L'architecture innovante pour l'identification binaire de la douleur faciale est basée sur des transformateurs d'images efficaces en termes de données (Deit). Deux bases de données, UNBC-McMaster shoulder pain et BioVid heat pain, ont été utilisées pour affiner et évaluer le modèle formé. La deuxième architecture proposée, repose sur des transformateurs de vision pour la détection de douleurs authentiques et simulées à partir des expressions faciales (ViT). Pour distinguer la douleur authentique de la douleur simulée, le modèle doit accorder une attention particulière aux changements subtils des expressions faciales dans le temps. L'approche employée prend en compte l'aspect séquentiel et capture les variations des

expressions faciales. Les expériences ont été menées sur la base de données BioVid Heat Pain démontrent l'efficacité de notre stratégie.

Mots clés : expression faciale, estimation de la douleur, CNN réseau de neurones convolutifs), transformateurs, détection de la douleur, douleur authentique et douleur posée.

PUBLICATIONS

- El Morabit, S., Rivenq, A., Zighem, M. E. N., Hadid, A., Ouahabi, A., & Taleb-Ahmed, A. (2021). Automatic pain estimation from facial expressions: a comparative analysis using off-the-shelf CNN architectures. Electronics, 10(16), 1926.
- El Morabit, S., & Rivenq, A. (2022, May). Pain Detection From Facial Expressions Based on Transformers and Distillation. In 2022 11th International Symposium on Signal, Image, Video and Communications (ISIVC) (pp. 1-5). IEEE.

ACKNOWLEDGMENTS

I would like to extend my heartfelt gratitude to the following individuals for their support, encouragement, and guidance throughout my academic journey.

First and foremost, I would like to express my deepest appreciation to my thesis advisor Pr Atika Rivenq, for their unwavering support, wisdom, and guidance. Their insightful comments, constructive criticism, and constant encouragement were invaluable in helping me complete this thesis.

I would like to thank my two supervisors, Pr Abdelmalik Taleb-Ahmed and Pr Abdenour Hadid for their invaluable guidance and support throughout my thesis journey. Your unwavering support, expert advice, and insightful feedback have been instrumental in shaping my research and bringing my work to fruition.

I am very grateful to all the members of the jury for their dedication and commitment in carefully evaluating my work, as well as for their genuine interest in my research.

I am also grateful to my colleagues and friends who have provided me with emotional support and encouragement throughout the journey. Your support and encouragement have been instrumental in helping me stay focused and motivated.

Finally, I would like to express my love and appreciation to my family, for their love, encouragement, and unwavering support. Without their constant encouragement and support, this accomplishment would not have been possible.

Thank you all for being an integral part of my journey and for helping me to achieve my academic goals.

Contents

List of Figures

List of Tables

Acronyms

AAM Active Appearance Model

ANN Artificial neural network

AI Artificial Intelligence

AU Action Unit

AUC Area Under the Curve

BCE Binary Cross Entropy

BSIF Binarized Statistical Image Features

BVP Blood Volume Pulse

CA Cumulative Attribute

CNN Convolutional Neural Network

CV Computer vision

DCT Discrete Cosine Transform

Deit Data-eEfficient Image Transformers

DL Deep learning

EEG Electroencephalography

EDA Electrodermal Activity

EMG Electromyography

FACS Facial Action Coding System

FFN Feed Forward Network

FMRI Functional Magnetic Resonance Imaging

FNIRS Functional Nearinfrared Spectroscopy

GA Genetic Algorithm

GSR Galvanic Skin Response

HCRF Hidden Conditional Random Field

HOG Histogram of Oriented Gradients

HR Heart Rate

LBP Local Binary Patterns

LDA Linear Discriminant Analysis

LLR Logistical Linear Regression

LPC Linear Predictive Coding

LPQ Local Phase Quantisation

LSTM Long Short Term Memory

MAE Mean Absolute Error

MFC Mel-Frequency Cepstrum

MFCC Mel Frequency Cepstral Coefficients

ML Machine Learning

MSE Mean Square Error

MTCNN Multitask Cascaded Convolutional Networks

NN Neural Network

NLP Natural Language Processing

PCA Principal Component Analysis

PCC Pearson's correlation coefficient

PSPI Prkachin and Solomon Pain Intensity

RASTA-PLP Relative Spectral Perceptual Linear Predictive

RFR Random Forest Regression

R-GAN Residual Generative Adversarial Network

RGB Red Green and Blue

RMSE Root-Mean-Square Deviation

RNN Recurrent Neural Network

SC Skin Conductance

SGD Stochastic Gradient Descent

SIFT Scale-invariant feature transform

SCL Skin Conductance Level

STE Short-Time Energy

SVM Support Vector Machines

SVR Support Vector Regression

TOP Three Orthogonal Planes

TSD Time series Statistics Descriptors

VAS Visual Analog Scale

ViT Vision Transformer

WSP Weighted Spatio-temporal Pooling

Chapter 1

Introduction

1.1 Thesis context

Pain is a complex phenomenon that is not yet fully comprehended. Pain is generally understood to be "an unpleasant sensory and emotional experience associated with existing or impending tissue damage, or described in terms of such damage" [85]. Nonetheless, the definition of pain is the subject of ongoing debate [8] and basic research continues to expand our understanding of it. As a subjective mental occurrence, pain is experienced differently by each individual [8]. Each individual learns the meaning of the term through early childhood experiences involving damage. This type of pain, referred to as acute pain, serves a protective function by alerting us to potential danger, allowing us to take preventative measures to limit further tissue damage and thereby facilitate the healing process [145]. In most cases, the pain will subside once the injury has healed. Pain that lasts longer than three months is typically classified as chronic or persistent pain. Pain can also be called nociceptive (caused by chemical or mechanical stimulation of sensory nerve fibers), neuropathic (caused by pain to the somatosensory nervous system), and psychogenic (caused, made worse, or lasted longer by mental, emotional, or behavioral factors).

Due to pain, a large number of individuals and the community as a whole face challenging obstacles. This is the most frequent reason why people visit the doctor [110]. The majority of patients (52.2%) who visited an emergency room did so due to pain, while only 34.1% visited for reasons unrelated to pain [29]. According to research conducted in [43], up to 35% of hospitalized patients experience severe pain, and more than 50% experience general discomfort. The incidence of pain was significantly higher than expected among hospitalized patients (83% [157]).

The introduction of innovative therapies has increased the demand for pain relief: Even after being cured of once-fatal conditions such as cancer, HIV/AIDS, and cardiovascular disease, an increasing number of people continue to experience chronic pain. A common side effect of surgical procedures, chemotherapy, and radiation therapy is pain [83]. The economic costs of caring for individuals with chronic pain are substantial [83], and workplace productivity suffers as a result. Chronic pain is more expensive than cancer, heart disease, and HIV/AIDS combined [83].

Experiencing pain is a complicated, individual, and unique phenomenon. Consequently, manual identification and quantification of pain is labor-intensive and objectively challenging [21]. This circumstance requires automated systems, where strong and cost-effective technology solutions can enable individualized and patient-centered

treatment. To allow such an automated method, at least one pain signal, also known as a modality, must be accessible as an input to the system. People's reactions to pain vary [141] since it's both an uncomfortable sensory and an emotional experience. This is why several studies have focused on different approaches to automatic pain evaluation. Behavioral (facial expressions, body movements, vocalization) and physiological (brain activity, cardiovascular activity, skin conductance response) are the key pain signals [141].

Facial expressions derived from video recordings, often in conjunction with head movements, have been the primary focus of the majority of the studies so far. Different physiological signs of pain have attracted attention due to advancements in wearable devices and electrode technology. In addition, the use of two or more pain indicators at once, known as a bi- or multimodal approach, is becoming more popular in research aimed at improving pain assessment performance. Unimodal refers to an evaluation system that only uses one modality to analyze information, whereas bimodal and multimodal refer to pain evaluation systems that use two, three, or more modalities (three or more).

1.2 Motivation and objectives

When it comes to objectively measuring pain and gaining insight into patients' health and enhancing pain treatment, automatic pain assessment is the best model. The field of automatic pain assessment is a dynamic area of multidisciplinary study that draws on insights from medical, psychology, computer science, and engineering. The medical field might benefit greatly from using this evaluation method. For instance, in a clinical context, a patient in pain can be assessed by nurses who will check and measure the severity of the patient's pain multiple times a day using medical devices. However, this medical surveillance is limited, since it is impossible to monitor the patients' pain every minute or second. Within this context, automatic pain assessment aims to construct a temporally dynamic, automated pain intensity identification system by using accurate and authentic pain response patterns that may be recorded.

Automatic pain evaluation consists of pain detection, often known as determining whether a person is feeling pain, pain intensity estimation, and the recognition of genuine and posed pain. In order to achieve this objective, pain-related information is fre-

quently gathered through the use of non-invasive sensor technology, which records the body responses of a person experiencing pain. This can be achieved by employing cameras that catch aspects such as facial expressions, head pose, and posture. Data can also be collected by using microphones to record audio characteristics. Contact-sensor are used to collect physiological signals, such as heart rate and blood pressure. This collected data is then used by machine learning algorithms to either detect pain, estimate pain level or detect posed pain. Figure 1.1 show the several possible data modalities for pain recognition system.

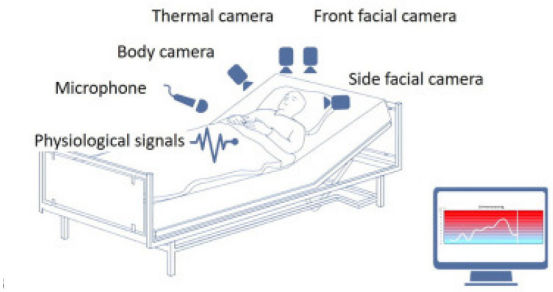

Figure 1.1: Possible modalities and sensors used to collect data in automatic pain recognition system. Figure adapted from [134]

The Facial Action Coding System is a method for decomposing facial movement into discrete measures of muscle change known as Action Units, and is used in the area of pain treatment to quantify experiences of facial discomfort [98]. The PSPI score quantifies pain as a composite of Action Unit intensities and detection scores [132]. Only specialists who have completed extensive training in the Facial Action Coding System (FACS) can conduct manual coding of facial expressions.

A human coder examines each frame of the video to detect and quantify Action Units, along with their associated start and end timestamps. Manually annotating every frame, even in a brief video clip, is a time-consuming and costly operation since videos are typically filmed at 24 to 60 frames per second [132]. To address the limitations of human coding, researchers are developing automatic approaches using ML techniques for encoding facial action units. Researchers can use OpenFace, a program that detects and measures action units in real time, to study pain [9].

Thanks to recent breakthroughs in facial expression analysis, researchers are now able to utilize a vast array of computer vision and machine learning approaches to obtain reliable pain ratings from facial emotions. The early efforts mostly concentrated on identifying pain based on the emotions on people's faces. In the figure 1.2, we present

a general framework for a pain recognition system from facial expressions. The dataset in this case consists of patients' facial images. The preprocessing step consists of cropping and alignment of the subject's face. Then these images are fed into the next block: feature extraction, which is in charge of detecting and extracting pain-related facial expression patterns. The extracted features are passed to the post-processing block. Then the learning module is where the training task is conducted.

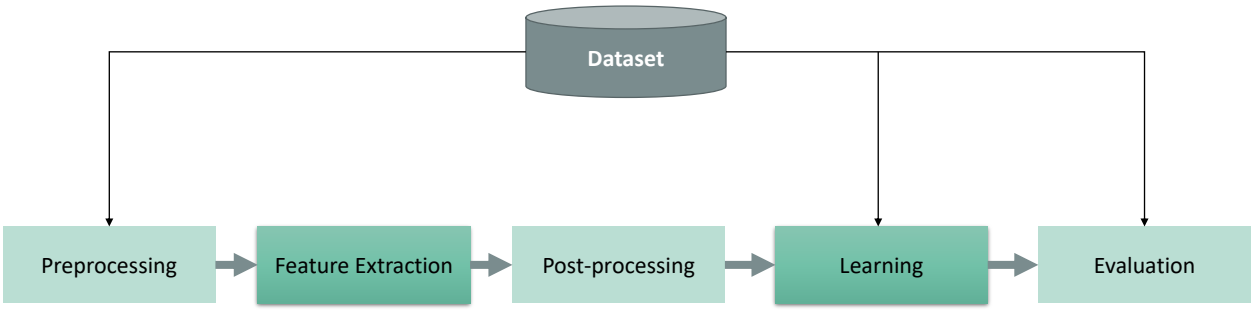

Figure 1.2: General procedures involved in constructing a facial expression-based automated pain detection system.

When it comes to building a classifier, it has become much simpler because of developments in machine learning and, in particular, deep learning. These developments have led to learning representations of the data, which makes it simpler to get information that is both helpful and discriminative. The goal of the combination of numerous non-linear transformations used in deep learning is to generate representations that are both more abstract and, eventually, more discriminative. This composition gives rise to the representations used in deep learning. So, the goal of this thesis is to look into several deep learning algorithms to detect pain, machine learning and, in particular, deep learning.

1.3 Contribution

Through the analysis of facial expressions, our thesis work makes a contribution to the study of pain, and it also makes a contribution to the area of deep learning through the

use of novel techniques. More specifically, the contributions of our thesis are:

- Our thesis's first contribution is a comprehensive analysis of automatic pain intensity assessment from facial expressions using five popular and off-the-shelf CNN architectures. This work studies the effectiveness of the hidden layers in these 5 Off-the-Shell CNN architectures for pain estimation by using features as inputs to two classifiers: SVR (Support Vector Regression) and RFR (Random Forest Regression). The experiments were conducted on the balanced UNBC-McMaster Shoulder Pain Expression Archive Database.

- The second contribution introduces a novel Vision Transformer architecture for the discrimination between Genuine and Posed Pain. In this work, we prove the efficiency of fine-tuning Vision Transformers using a small database, contrary to the use of ViT from scratch on the same database. In addition, we prove the importance of sequential order in time for the discrimination between Genuine and Posed pain.

- The third contribution in this thesis consists of presenting a fine-tuned data-efficient image transformer (Deit) for pain and no pain detection. In this work, we highlight the importance of transformers in the image recognition field in general and in pain tasks more particularly. Also, prove the efficiency of transformers compared to Convolutional Neural Networks (CNN) while studying the discrimination of pain from no pain task.

1.4 Thesis outline

This thesis may be divided into two major parts. Each of the five chapters in the first section provides an overview of the current state of the art in automated pain assessment, as well as a synopsis of the contributions and findings reported in the original articles. The second part covers the main publications that contributed to the development of this thesis.

In the first chapter, which serves as an introduction to the thesis, we discuss the history of the topic being investigated as well as the motivations for conducting the

research. In addition, we outline the contributions made by the thesis and provides a concise summary of the articles. Chapter 2 provides an overview of automatic pain evaluation. This section offers an introduction to pain intensity estimating systems, a description of the publicly accessible pain databases utilized in the experiments performed for this thesis, and a literature assessment of the current state-of-the-art approaches.

Chapters 3 and 4 present the works presented in the original papers. Chapter 3 provides a Comparative Analysis Using Off-the-Shelf CNN Architectures for pain assessment from facial expressions. This work focuses on the extraction of deep features, using CNNs, then training SVR (Support Vector Regression) and RFR (Random Forest Regression) to estimate pain. Chapter 4 introduces Vision Transformers. These transformers were used for two different topics: Detection of Genuine versus Posed Pain and Detection of Pain.

Chapter 5 is the last part of the thesis. It states our conclusion and gives ideas for future research on automated pain evaluation.

Chapter 2

Overview of automatic pain assessment

2.1 Introduction

This chapter focuses on the use of artificial intelligence (AI) and deep learning in the assessment of pain. AI has become an increasingly valuable tool for diagnosing and controlling pain in medical settings as technology has advanced. In this chapter, we will look into various AI applications in facial images, such as emotion identification and medical diagnostics. We will also delve into the pain process and the reactions of patients to it, including biological, physiological, and behavioral reactions.

We will also go through pain datasets, covering the stimulation and labeling methods, as well as the datasets that are currently available for AI-based pain evaluation. Furthermore, we will examine the use of AI and deep learning in pain assessment, including deep learning techniques, training and classification, and computer vision. Finally, we will discuss automatic pain assessment approaches, such as automatic pain recognition and the overall architecture of automatic pain detection systems.

2.2 AI applied facial images

Communication, both verbal and nonverbal, plays a key role in society and is required for many daily tasks. Facial expressions convey a great deal about a person's mental and emotional state, as well as their intentions, and are therefore the most effective form of nonverbal communication [35]. Without speaking, listeners can send a range of information to the speaker through facial expressions and affect the conversation's direction. In [22], the authors claim that the information supplied by a person's face should be given greater weight in cases where it does not correlate with the other method of communication, i.e., spoken words.

As technology progresses, so do our computational capabilities. This means that research into automatic facial detection is improving. Expression analysis and automated recognition have garnered a lot of attention in the field of computer vision research. Human-computer interaction, entertainment, medical applications (pain detection), social robots, interactive video, and behavior monitoring are just some of the many do-

mains that might profit from a system that can understand facial emotions. Some further information on the use of facial images in applications is provided below.

2.2.1 Emotion recognition

The facial expressions of a person reveal a great deal about their mental state and motivations. A person's facial expressions and tone of voice can indicate a variety of emotions, including happiness, sadness, and anger. According to a number of studies, only one-third of human communication consists of words, while the other two-thirds consists of nonverbal cues. The emotional meanings conveyed by facial expressions and other nonverbal components play an important role in interpersonal communication. In recent decades, there has been a growing interest in face emotion research due to its applicability to fields as diverse as perceptual and cognitive sciences, affective computing, and computer animation [67].

Automatic facial expression recognition (FER) has gained traction alongside the rise of AI in fields such as human-computer interface (HCI), virtual reality (VR), augmented reality (AR), advanced driving assistance systems (ADAS), and entertainment. Inputs for FER can come from a variety of sensors, including an electromyograph (EMG), electrocardiogram (ECG), electroencephalogram (EEG), and camera. However, the camera is the most promising type of sensor since it does not require wear and gives the most informative information for FER [67]. Figure 2.1 highlights a classification of Emotion Recognition Methods According to used sensor [110].

2.2.2 Medical diagnosis

There is a growing interest in the use of computer vision for automatic medical diagnosis since it may give objective, non-intrusive data on a patient's health with no disruption to the medical process. The face is a window to the body's internal health, and its features can reflect the presence or absence of certain disorders. Thus, it is of utmost relevance in medical diagnostics to detect facial abnormalities or unusual traits. Different methods have been explored to evaluate these symptoms and give further support to

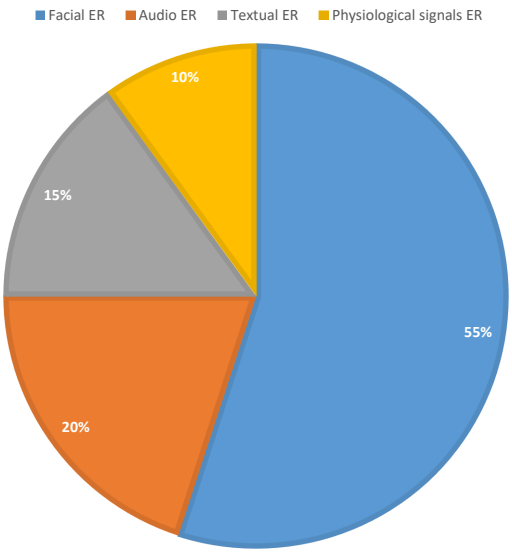

Figure 2.1: classification of Emotion Recognition Methods According to used sensor [110]

the medical community. Despite this, the produced instruments are rarely employed in clinical practice due to ongoing concerns about their dependability as a result of insufficient application and a lack of clinical validation of the methodology. However, efforts are being made to provide robust solutions that are suitable for healthcare settings, including addressing issues such as real-time assessment and patient placement.

In order to aid clinicians in their diagnostic work, computer vision was offered as a method for offering an automated and objective evaluation of facial traits. Face recognition systems have numerous potential applications. Examples include psychiatric diagnosis, facial paralysis detection, and automated pain estimation. Doctors and other medical personnel would do well to inquire about the patient's level of pain in order to properly assess the patient's condition and possibly prescribe painkillers as an initial treatment. Significant research has been conducted on the topic of automatic pain detection. As a result of these studies, a correlation has been established between facial expressions and the use of specific muscles across a broad age range. Because they affect the facial nerves, Bell's palsy [44] and facial paralysis [50] are two disorders that can cause abnormal facial expressions. Movement impairments, such as altered facial expressions, may also be a consequence of brain damage caused by conditions such as stroke [10] and transient ischemic attack. Patients with a variety of mental illnesses, particularly psychotic disorders in which one's perception of reality is distorted, may exhibit peculiar facial expressions.

2.2.3 Other applications

The application of computer vision systems to the study of facial expressions is not limited to medical diagnosis or emotion recognition; rather, it offers a vast array of possible applications. For instance:

- **Age estimation :**

  In the field of face image processing, researchers have devoted close attention to the information communicated by human faces. Researchers have paid close attention to the huge potential for image-based age and age-group estimations in fields like age-invariant face recognition and face verification across age, as well as in commercial and law enforcement settings. There has been a great deal of research on age estimation with the aim of identifying aging trends and changes and determining the best way to describe an aging face for accurate age calculation [3].

  Age estimation is the process of automatically assigning an age or age range to a person's face. One's actual, perceived, guessed, or estimated age may be employed. The genuine age of a person, expressed as a whole number, is the entire number of years from birth [37]. The estimated age of a subject is determined by a machine using facial visual data, whereas appearance and perceived age are dependent on how old a person looks to be. It is commonly acknowledged that a person's apparent age correlates to their actual age, despite the stochastic nature of aging. Visual indicators of appearance age are employed to determine both estimated and perceived age. Few studies on estimating ages and age groups have been published [37]. This may be explained by the fact that guessing an individual's age is not a normal categorization task. Age estimate can be approached as a regression issue, a multi-class classification issue, or a combination of the two, depending on the nature of the task at hand [3].

- **Kinship verification** :

  In the field of computer vision, kinship verification is a novel challenge that attempts to detect, based on face images, whether two people are related. There are various applications for kinship verification, including image annotation, child adoption, social media analysis, etc. But it's hard to tell who someone is related to just by looking at their face because age, gender, and genetics can make a big difference. In the preceding decade, increasingly effective strategies have

emerged[99].

In the past decade, many algorithms have been developed to address the challenges of kinship verification. Most recent reviews [99] focus on the small-sample scenario and only consider four forms of kin relations: father-son, father-daughter, mother-son, and mother-daughter. Now that the FIW database [101] is available, it will be interesting to see how well the most advanced kinship verification algorithms work with larger samples that include more types of family ties.

- **Face Anti-Spoofing :**

Face recognition has developed into an increasingly important security technique as a result of AI's extensive application. Face anti-spoofing is an area of great concern since it protects users from fraud and other sorts of attacks. Face spoofing detection research has been constantly updated and improved since the first method of manually extracting features based on image texture, human-computer interaction, life information, image quality, and depth information. Today, deep learning is used to automatically extract features, along with network updates, transfer learning, feature integration, and generalization of the domain [153] [88].

2.3 Pain Process and patients' Reactions

Pain is an interior, private mental feeling that is totally individual. Pain is not just a sensory phenomenon; it also includes sensory-discriminative, affective-motivating, and cognitive-evaluative aspects [86]: it varies in intensity, location, duration, and quality; it is unpleasant and motivates activity for pain relief; and it is influenced by cognitive factors such as the evaluation of the severity of an injury, distraction, or cultural values [86] [127]. It may be difficult to distinguish between the pain experience, the pain cause (such as tissue damage with nociception), the pain response (verbal and nonverbal displays of pain), and pain assessment (e.g., by a caregiver) [86] [127]. It is possible to identify the source of pain (as in the case of a fracture) and control it by purposeful pain stimulation (as in the case of neurological tests), but it is also possible for the reason to be unknown or nonexistent (especially in chronic pain). It is important to note that elements such as one's thoughts, memories, and surroundings all contribute to the patient's sense of pain.Sometimes, people do not feel pain, which is problematic [150].

The majority of the time, however, individuals do respond visibly to pain, and these reactions are influenced by a variety of personal and contextual factors. This section provides a general review of pain processes and reactions.

2.3.1 Biological Process

The brain and peripheral nervous system both have a role in the perception of pain. In many instances, the process starts when harmful mechanical, thermal, cold, chemical, or inflammatory stimuli activate sensory neuronal circuits. These stimuli activate nociceptors, which are primary sensory neurons with specific surface receptors for sensing nerve impulses. Information concerning a painful event in the periphery may activate both excitatory and inhibitory interneuronal circuits in the spinal cord, resulting in a withdrawal reflex. Diverse supraspinal structures are responsible for the processing of nociceptive input, which eventually results in the perception of pain. As with psychogenic pain, it is conceivable for a person to experience pain without activating the nociceptive pathway [147] [64] [111].

Figure 2.2 from [114] gives a graphical explanation of the pain mechanism stimulated by an intense heat. The figure shows that the action potentials are delivered to the spinal cord through afferent axons. IB4-negative unmyelinated nociceptors synapse in lamina I and outer lamina II, while IB4-positive nociceptors terminate in inner lamina II. Nociceptors provide chemical signals to spinal neurons, which subsequently stretch their axons down the spinal cord and along fiber tracts, eventually terminating in the medulla, the midbrain, and the thalamus, which are responsible for processing the experience of pain. One of the regions of the brain that receive projections from thalamic neurons is the somatosensory cortex [114].

2.3.2 Physiological Reactions

Sympathetic stimulation outflow is the consequence of extensive interactions between the brain regions involved in pain sensation and autonomic regulation [15]. These interactions cause detectable changes in a variety of physiological signals.

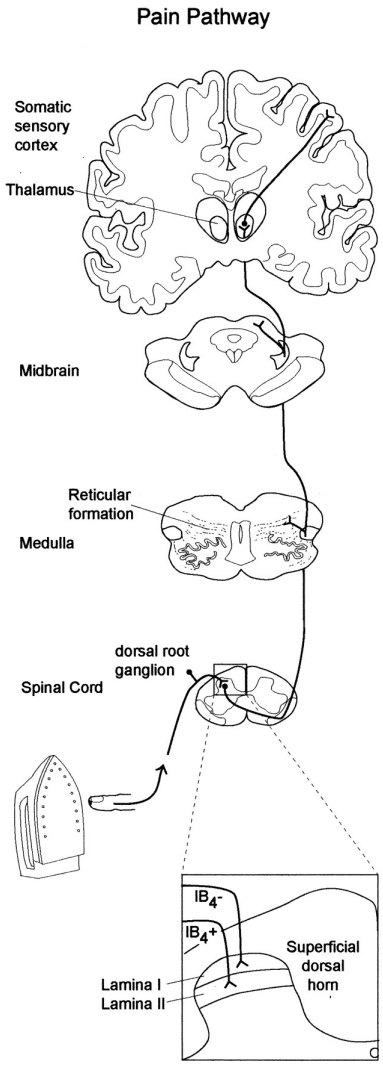

Figure 2.2: Graphical representation of Pain Mechanism circuit by painful stimuli such as high temperatures.

A change in skin conductance [18], which is an autonomically regulated signal, indicates pain presence. Due to the fact that sympathetic excitatory efferent neurons are the only kind of neurons that innervate sweat glands, the increased sympathetic outflow associated with pain causes sweat to be secreted via pores on the skin's surface [15]. The generation of sweat affects the electrical properties of the skin (electrodermal activity (EDA)), resulting in an increase in the skin's electrical conductance until the sweat is reabsorbed or dissipated.

Increased sympathetic activity also results in significant cardiovascular alterations. It alters heart rate, resulting in tachycardia, as well as heart rate variability [121] [4]. Moreover, power spectrum analysis reveals that pain considerably increases the power at low frequencies. In addition, pain raises peripheral vascular resistance and the volume of a stroke. This, in conjunction with the higher heart rate, causes a rise in resting blood pressure [107]. Furthermore, the pupil diameter is also affected by pain experience, due to the pupil dilation reflex [23].

Pain affects the electrical and metabolic activity of brain cortical regions, since pain mechanism includes a complex network of these regions [125]. There are two techniques that show potential for identifying pain response patterns. These techniques are : Electroencephalography (EEG) to identify changes in electrical activity in the cortex, and functional magnetic resonance imaging (fMRI) and functional nearinfrared spectroscopy (fNIRS) to determine changes in brain hemodynamics in response to increased metabolic demand [96].

2.3.3 Behavioral Reactions

Behavioral pain includes reactions to protect the body, and pain expression and communication. Facial expressions, body language, and vocalizations are all examples of behavioral reactions to pain. Furthermore, changes in daily behavior and social interaction are a common consequence of living with chronic pain [145].

Certain facial expressions are consistently associated with pain across a broad range of clinical pain syndromes and experimental pain modalities. As the unpleasant stimulus's intensity increases, so does the amplitude of the facial emotions it generates. The Facial Action Coding System (FACS) is often used to the study of facial expressions due to its capacity to classify facial expressions in terms of elementary Action Units (AUs) based on facial muscle activity. Other instances of pain behavior include paralinguistic vocalizations (crying, moaning, groaning, gasping, and sighing) and voice quality traits (amplitude, timbre, and hesitancy) seen during verbal self-report of pain [30].

Most pain-related activities serve to avoid more injury and alleviate present suffering. Among them are protective responses, clawing, writhing, and guarding. There is no uniformity in the exact patterns of physical exercise identified by the medical com-

munity. A study by Walsh et al [133], observed that body language, such as averting the head or trunk, caressing a body part, bending the knees, or shrugging the shoulders, might express pain. Another study in [142] examined three pain datasets and discovered that during pain, people's heads move differently, at different rates and in different ranges than when they are not in pain. In addition, they found that comparable patterns are more prevalent in the setting of acute pain than chronic pain.

2.4 Pain datasets

Representative data are required for creating and demonstrating the utility of a pain identification system. The following sections discuss the recording and use of data for pain recognition: first, pain stimulation. Second, data labeling and evaluation. Finally, Publicly available datasets.

2.4.1 Pain stimulation

Data from people experiencing pain must be collected in order to build and evaluate pain identification systems. This can be done on patients in clinical settings or on healthy volunteers, as is usual in fundamental and pharmacological research. Anxiety, infirmity, distraction, uncertainty, expectations, sadness, and drugs all have an impact on how patients perceive pain. For experimental design, it is important to use subjects that have minimal bias and closely resemble the population of interest. This is why some datasets contain healthy subjects instead of patients. The intensity, the frequency and duration are controllable in this case.

Pain is a subjective experience that is difficult to measure. Pain can be described as the unpleasant sensory and emotional experience associated with actual or potential tissue damage. The most common methods of pain stimulation are mechanical, thermal, electrical and chemical modalities. Concerning mechanical modality, the patient is typically exposed to a variety of pressures and vibrations in order to assess their sensitivity

thresholds. Thermal modality consists of exposing the person to heat in order to assess their sensitivity thresholds for different temperatures. The electrical method is a process where electrodes are placed on the skin and an electric current is sent through them. Finally, mechanical stimulation is a technique that uses a device that pushes or pulls on the skin to cause pain. Otherwise, many patients in clinical conditions experience chronic pain as a result of disease or damage that is not caused by external stimulation. However, external stimuli or several essential procedures and activities typically intensify pain sensation and reaction. For example, turning, central venous catheter placement, and wound drain removal are all painful clinical procedures in critical care.

2.4.2 Data labeling

To determine ground truth, the methods could be divided into three categories: self-report scales, observer evaluation or research design. Concerning self-report scales, they are considered to be the gold standard to evaluate pain due to the subjective aspect of pain experience. Nonetheless, this method still limited when the patient can not express his feelings. For example, in case of neonates, people with some kind of handicaps, or patients in coma. Regarding the second method, which is observer evaluation, there are various scales adapted to the patients' medical issue. Most of them focus on the facial expressions, body movement and voice. However, this method is inadequate if we consider the fact that not all patients exhibit their feelings, also the expression of pain is different from one patient to another. The widely used observer scale in pain detection is the Prkachin and Solomon Pain Intensity (PSPI). It is calculated by combining the intensities of some facial action units (FACS). The action units (AUs) used to for pain recognition are the following : brow lowerer (AU4), cheek raiser (AU6), lid tightener (AU7), nose wrinkler (AU9), upper lip raiser (AU10) and eyes closed (AU43). The Prkachin and Solomon Pain Intensity score expression is given by the equation 2.1. Figure 2.3 gives a sample from UNBC McMaster shoulder pain database, with facial AUs and their intensity levels used to obtain PSPI score, equal to 12 in this case. The third and last category to determine the ground truth is the study design. It consists on getting pain scale based on preliminary experimental studies.

$$Pain = AU4 + max(AU6, AU7) + max(AU9, AU10) + AU43 \tag{2.1}$$

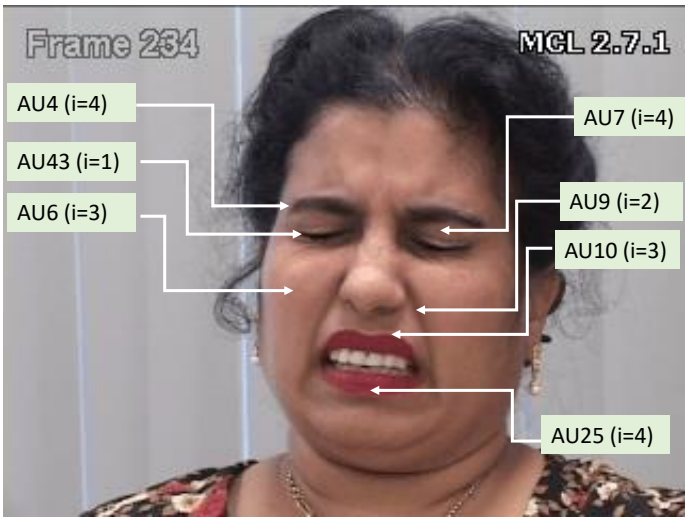

Figure 2.3: Sample from the UNBC-McMaster pain shoulder archive with a PSPI score = 12. The score is obtained from facial AUs mentioned in the target face.

2.4.3 Available Datasets

Data is used to build machine learning models for automated pain recognition.Bellow, examples of available datasets. The table2.1 summarizes and gives more of the available datasets that were used in the literature.

- **UNBC-McMaster Shoulder Pain**

  The lack of representative data is a major impediment to the deployment of a fully functional automatic facial expression detection system. One solution is to narrow the context of the target application so that enough data is available to build robust models with high performance. One such application is automatic pain detection from a patient's face. Researchers from McMaster University and the University of Northern British Columbia captured video of participants' faces (with shoulder pain) while performing a series of active and passive range-of-motion tests on their affected and unaffected limbs on two separate occasions to facilitate this work. The frames in this dataset were coded using AU (Action Unit) by certified FACS (Facial Action Coding System) coders [81]. The UNBC-McMaster Shoulder Pain Expression Archive includes: Temporal Spontaneous Expressions: 200 video sequences capturing patients' spontaneous facial expressions, Manual FACS codes: 48,398 FACS coded frames, Self-Report and Observer Ratings: associated pain self-report and observer ratings at the sequence level and Tracked

Landmarks: 66 point AAM landmarks [81]. Figure 2.4 presents examples of frames in the UNBC-McMaster Shoulder Pain Expression Archive.

Figure 2.4: Examples of some of the sequences from the UNBC-McMaster pain shoulder archive [81]

- **BioVid Heat Pain**

  One of the most fundamental jobs in clinics is assessing acute pain. Normally, doctors rely on the patient's speech. Which is less trustworthy and valid for mentally ill patients. Furthermore, it cannot be used on people that can't express their feelings, such as newborns. However, there are various signs that suggest discomfort. These include facial expressions, psychobiological measures such as heart rate. Therefore, without patient's self report, pain can be assessed with the use of this information.

  To enhance pain assessment methods, particularly automated assessment methods, the BioVid Heat Pain Database studies were experimented on 90 participants [143]. Those participants were subjects to experimentally induced heat pain in four intensities. These four intensities ware triggered 20 times in a random order. The highest temperature was sustained for 4 seconds for each simulation. The time between stimuli was varied between 8 and 12 seconds. The experiment was repeated twice : first, capturing face expression, second, using EMG sensors on the face [143].

  The dataset is divided into five parts: part A, contains short time videos of pain stimulation without facial EMG. Part B, consists of sort videos of pain stimulation with facial EMG. Part C, contains long videos of pain stimulation without facial EMG. Part D, contains posed pain and other emotions. Finally, part E, consists of emotion elicitation with video clips. All these parts contain frontal video and

biomedical signals [143]. Figure 2.5 shows face samples from the BioVid Heat Pain database, and a pain stimulation using a thermo at right arm.

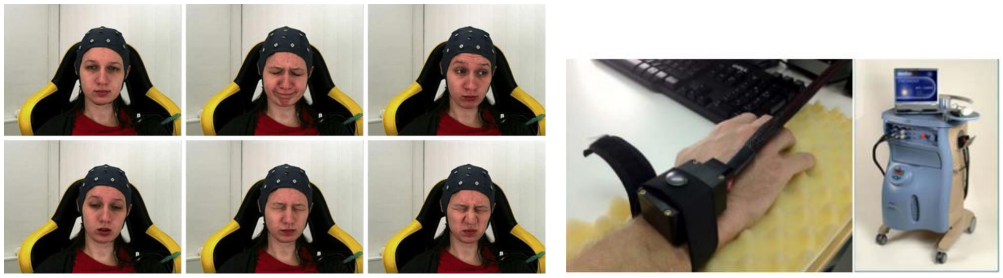

Figure 2.5: Face samples from the BioVid Heat Pain database on the left, and Pain stimulation using a thermo at right arm on the right [143]

- **BP4D-Spontaneous**

  The BP4D-Spontaneous is a 3D video database of spontaneous facial expressions in a broad sample of young adults between 18 and 29 years old. The FACS (Face Action Coding System) was used to provide frame level ground truth. To stimulate the emotions, cold pressor and emotion elicitation were used [154]. The Figure 2.6 gives samples of textured models, shaded models and original 2D videos.

Figure 2.6: Samples of textured models, shaded models, original 2D videos [154]

- **BP4D+**

  The BP4D+ is a multimodal dataset. It consists of 140 participants. Data were collected from a number of facial sensors, including high-resolution 3D dynamic imagery, high-resolution 2D video, and thermal sensing; as well as touch physiological sensors, which included electrical conductivity of the skin, respiration, blood pressure and heart rate. [154] Figure 2.7 represents sample data from a participant.

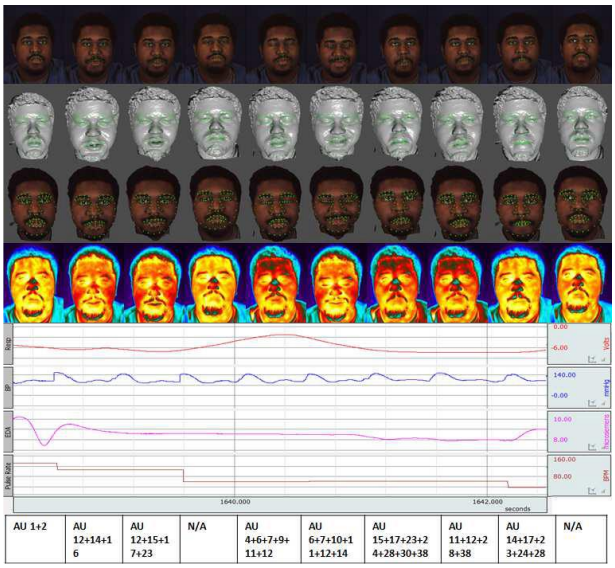

Figure 2.7: Sample data sequences from a participant including original 2D texture (first row), shaded model (second row), textured model (third row), thermal image (fourth row), physiology signal(fifth row: respiration rate, blood pressure, EDA, heart rate) and corresponding action units(last row) [154]

- **MIntPAIN Database**

  The MIntPAIN database was collected by experiencing electrical pain on healthy adults. The dataset contain 20 subjects with stimulated muscular pain. During the data collection session, each individual displayed two assessments, each one with 40 sweeps of pain stimulation. In each sweep, we collected two data points: one for no pain (Label0) and one for pain (Label1-Label4) [48]. Each assessment comprises 80 files from 40 sweeps in total. Each of the 80 trial files comprises three folders containing RGB, Depth, and thermal video frames from a single stimulation. Figure 2.8 presents faces from two subjects for all 5 pain levels for the three different modalities.

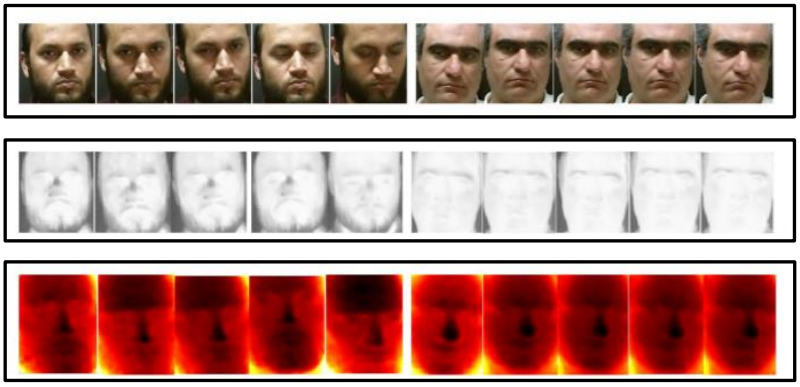

Figure 2.8: Sample data sequences from two participants. The frames in the first row represent RGB faces. The second row contains Thermal faces. Finally, the third row presents Depth faces [48].

2.5 Artificial intelligence and deep learning

In general, artificial intelligence refers to intelligence possessed by a machine as opposed to a natural biological living form. A computer or a specific software on a machine may be considered artificially intelligent if it can sense its environment and behave accordingly to achieve a predetermined objective. Machine learning and deep learning are two popular subfields of artificial intelligence (AI). Both of these subfields execute AI using their own unique forms of learning algorithms. Artificial intelligence is widely used in today's applications, fore instance : object detection in self-driving cars, recommendation methods employed by YouTube and Netflix.

Machine learning (ML) is a branch of artificial intelligence that use algorithms to examine data, learn from it, improve, and then make a decision or prediction about new data. In machine learning, rather than manually generating code with a specific set of instructions to execute a certain task, the machine is trained using data and algorithms to perform the task without being instructed how to do so. Contrary to the conventional method, which consisted of manually developing code with a precise set of instructions to do the operation. Within the discipline of machine learning, there are several types of learning algorithms that may be used to any type of data to accomplish any objective. Examples of typical ML algorithms include: Decision Trees, Support Vector Machines (SVM), K-Nearest Neighbor.

Dataset	Features	Stimuli	Subjects	Classes
UNBC 2011[81]	Facial expression RGB	Natural shoulder pain	129 Shoulder pain patients (63 males, 66 females)	0–16 (PSPI) and 0–10 (VAS)
BioVid 2013[143]	-Video: Facial expression RGB -Biopotential signals (SCL, ECG, sEMG, EEG)	Heat pain at right forearm thermode	90 Healthy	5 (no pain, 4 levels of pain)
BP4D-Spontaneous Database (BP4D) 2014[154]	Facial expression	Cold pressor test with left arm	41 healthy	8 classes of pain
BP4D+ 2016 [154]	Facial expression -EDA, heart rate, respiration rate, blood pressure	Cold pressor test with left arm	141 healthy	8 classes of pain
SenseEmotion 2016 [130]	-Facial expressions: -Biosignals -ECG, EMG -GSR -RSP -Audio	Heat pain	40 heathy	5 (no pain, 4 levels of pain)
EmoPain 2016 [6]	-Audio, Facial expressions -Body movements -sEMG	Natural while doing physical exercises	22 chronic low back pain	2 for face 6 for body behaviors combined: binary
MIntPAIN 2018 [48]	Facial expression: RGB, depth, Thermal	Electrical pain	20 healthy	5 classes (0–4)
X-ITE pain 2019 [144]	-Audio, Facial expressions -ECG, SCL, sEMG (trapezius, corrugator, zygomaticus)	Heat and electrical	134 healthy adults	3 pain levels
COPE 2005 [131]	Facial expression	heel lancing for blood collection	26 neonates	5 pain levels

Table 2.1: Summary of Pain available databases.

Deep learning is a subfield of machine learning that deploys artificial neural networks (ANNs). These algorithms are inspired by the structure and function of the human brain to learn from data. The ANNs were first introduced by McCullouch and Pitts [84]. Below, more details about deep learning.

2.5.1 Deep learning

Deep learning (DL) models, as defined by Goodfellow et al [42], consist of organized layers that receive input from the previous layer and transmit it to the next one. As cited before, deep learning uses artificial neural networks. An ANN is a computer system consisting of a collection of interconnected components known as neurons that are arranged into what we refer to as layers. These layers are linked to each other by weights. Figure 2.9 shows a general architecture of an ANN. The first layer in a network is known as the input layer, the final layer is known as the output layer, and any levels in between are known as hidden layers. The network is said to be deep if it contains several hidden

layers. There are different types of layers. For instance, dense layers convolutional layers, pooling layers, etc. Each link between nodes is assigned a weight. Every weight is a representation of how strongly two nodes are connected to one another. Given that each node is related to all the nodes in the layer before, the sum of these weights is passed then to an activation function. There are several activation functions, each of which transforms data differently and depending on the contexts. The following are some of the most well-known ones: ReLu, Sigmoid and Softmax.

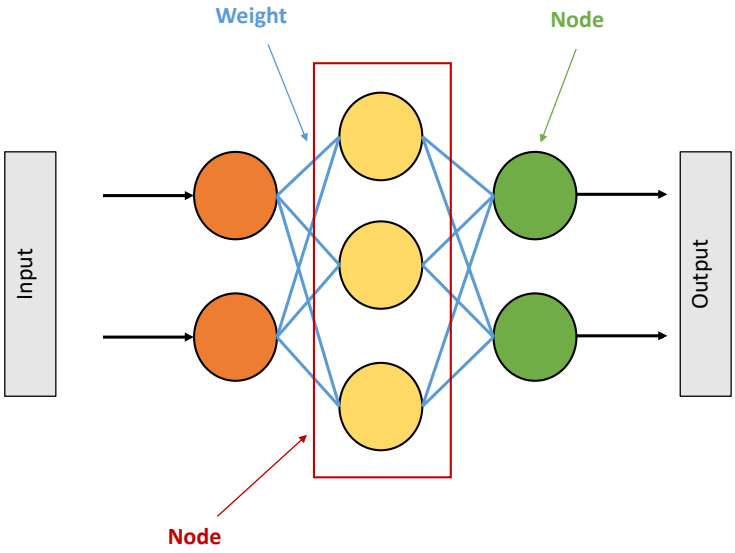

Figure 2.9: An example of an artificial neural network with one hidden layer.

Figure 2.9 presents an example of a convolutional neural network (CNN). The most important layers of a CNN are the convolutional layers, they are also. These layers' main role is to detect patterns in images using filters. An image may contain several patterns. By patterns, we mean edges, shapes, colors, etc. Therefore, one filter is able to detect one type of patterns. The network gets more complex and sophisticated when it gets deeper. In further layers, the filters may be able to recognize particular items like eyes, ears, hair, rather than simple shapes and edges. Convolution involves sliding the filter over the image's height and width to compute the dot product of each filter element with the input at each pixel. An illustration of this convolution procedure is shown in Figure 2.10. By convolving the filter with the green component of the input image, we can determine the first entry of the activation map (highlighted in Figure 2.10). With this procedure repeated for each pixel in the input picture, the

activation map is produced. Activation maps are created for each filter in the convolutional layer, and they are stacked along the depth axis to produce the final volume of the convolutional layer's output. All the activation map's nodes might be interpreted as the neuron's output. Consequently, each neuron is linked to a tiny local region in the input picture, and the size of the region corresponds to the size of the filter. In an activation map, all the neurons share the same settings. Because of the convolutional layer's local connection, the network is driven to train filters that have the best response to a local area of the input. The initial convolutional layers capture the low-level features (e.g., lines) of images, while the later layers extract the high-level features (e.g., shapes and specific objects) [93].

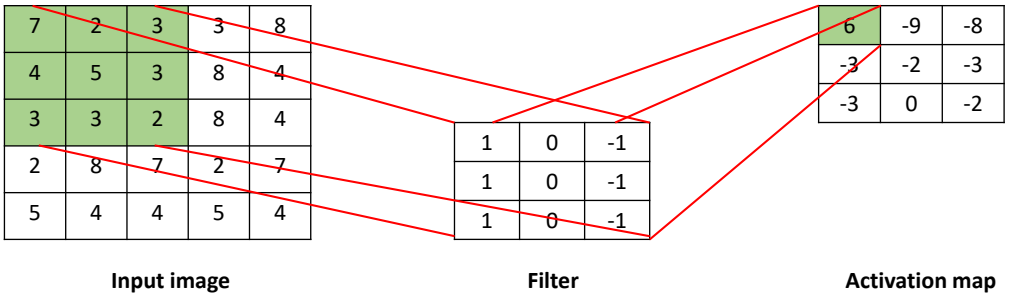

Figure 2.10: Example of convolutional process

Convolutional layers' feature map output is limited by the fact that it retains the specific input feature locations. In other words, the feature map will change depending on how the feature is positioned in the input picture. Minor adjustments to the original image, such as cropping, rotation, and shifting, can generate different feature maps of the same image. Therefore, the common approach used to avoid this problem is the use of a pooling layer. So, following the convolutional layer comes a new layer called the pooling layer. Particularly, after the application of a nonlinearity (e.g. ReLU) to the feature maps generated by a convolutional layer. Implementing the layers of a convolutional neural network often involves adding a pooling layer after the convolutional layer, and this process may be repeated several times in a single model. Pooling entails choosing a pooling process, similar to applying a filter on feature maps. The size of the pooling operation or filter is less than the size of the feature map. The pooling process typically performs two functions: Average Pooling and Maximum Pooling (or Max Pooling). The first one consists of calculating the average value of each patch of the feature map. The second one, calculates the maximum value for each patch of the feature map [93]. Figure 2.11 shows an example of max-pooling operation.

Concerning the fully connected layer, each neuron is linked to every neuron in the

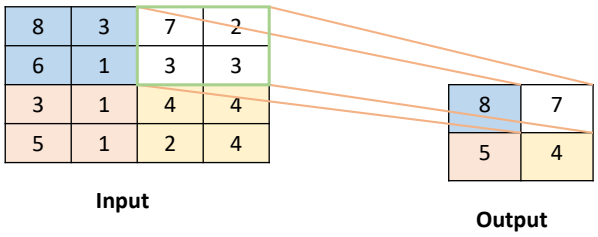

Figure 2.11: Example Max-pooling operation

layer underneath it. This layer represents the final, high-level classification rationale. Its size is proportional to the number of classes, and each neuron represents the likelihood that the input belongs to that class. This layer examines the preceding layer to identify which attributes relate most likely to certain classifications.

2.5.2 Training and classification

The training of a neural network is its most crucial step. Before being trained, a CNN has no preexisting knowledge or understanding; its weights and filter values are completely arbitrary. While training, the computer will need to adjust these settings to reduce loss and improve classification precision. At each iteration, the training algorithm executes a series of steps. Four steps: Forward Propagation, backward pass, loss function calculation, and weight value update.

In neural network development, the quality of the training data is critical. There are three sets of images, the first one is the training set, the second one is the validation set and the third one is the test set. The first set is utilized to train the model. The model will be trained repeatedly on the same data from the training set, enabling it to gain a better knowledge of the data's features with each epoch. Separate from the training set, the validation set is a collection of data used to verify our model during training. Information gleaned from the validation process might be used to change and fine-tune the hyperparameters we've set. Furthermore, a validation set is required to determine whether our model has been overfit to the training set. After training, the model will be evaluated using the test set. The test set is distinct from both the training set and the validation set.

The neural network receives training data in batches. In this context, "batch size" refers to the number of images that CNN processes before updating the parameters. Each epoch represents the time at which all batches have traversed the network successfully. Training may go as long as required by modifying the number of epochs (the number of times training data is employed throughout the training process) or by applying alternative ending conditions. At the start of each epoch, all training data are combined and presented to the model in new batches. Each of the training phases will be described in further detail below.

- **Forward Propagation:** We feed data into the model while training an artificial neural network. This data is propagated through the model via forward propagation, where we continually calculate the weighted sum of the preceding layer's activation output with the relevant weights and then feed this total to the subsequent layer's activation function.
- **Optimization Algorithm:** An optimization procedure is used to get the optimal values for the weights. The selected algorithm is often referred to as an optimizer. Stochastic gradient descent, or SGD for short, is the most well-known optimizer. SGD's objective is to minimize a defined function, often known as a loss function. Therefore, SGD updates the model's weights to bring this loss function as close to its minimum value as possible.
- **Loss function:** With each iteration, SGD attempts to bring the loss function down to a minimum by adjusting the network's weights. Therefore, SGD updates the model's weights to bring this loss function as close to its minimum value as possible. There are several loss functions. For instance, Cross Entropy and Mean Square Error (MSE). The choice of the loss function depends on the objective of the model.
- **Backpropagation :** When the forward propagation reaches the output layer, the loss function is calculated and the gradient descent then works to minimize this loss. To achieve this minimization, gradient descent first determines the gradient of the loss function and then adjusts the network weights to account for it. To achieve this minimization, gradient descent first determines the gradient of the loss function and then adjusts the network weights to account for it. Gradient descent employs backpropagation to do the actual gradient computation.

2.5.3 Computer vision

Computer vision stands out among the numerous fields where artificial neural networks have been extensively implemented and proven useful. Computer vision (CV) is concerned with providing computers the ability to analyze and understand visual data such as videos and images. Simply expressed, this is the method of giving computers vision skills comparable to human eyes. Much of the analysis and processing of digital visual data is automated with the help of computer vision. Figure 2.12 presents some computer vision problems that have benefited greatly from the use of deep learning algorithms.

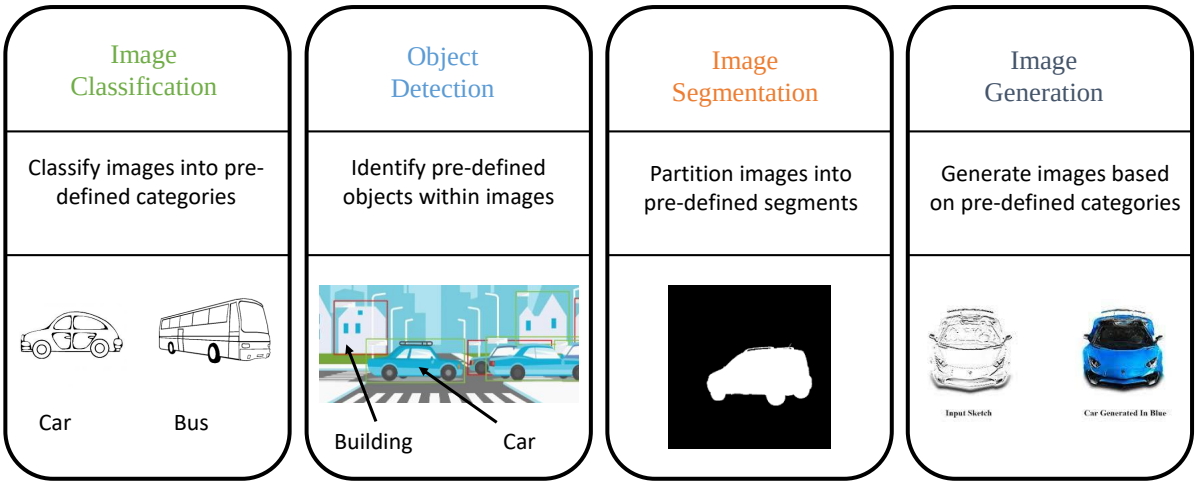

Figure 2.12: Examples of computer vision tasks

Computer vision consists of three fundamental steps: first, image acquisition. Video, photographs, or 3D technologies may be used to gather images in real-time, even in massive quantities, for the purpose of analysis. Second, image processing. Deep learning models automate a substantial portion of this procedure; yet, the models are often trained using thousands of tagged or otherwise pre-identified images. Finally, image understanding. In this concluding interpretive step, an object is either identified or classified.

2.6 Overview of automatic pain assessment approaches

In the first part of this section, we will provide an overall summary of the existing studies for automated pain identification. Next, we provide an overview of the present techniques for the analysis of facial expressions in order to estimate pain from the face.

2.6.1 Automatic Pain Recognition

With the increasing need for regular and consistent monitoring of pain in clinical settings and at home, automated analysis of pain is a new topic of artificial intelligence study. There are a number of newly developed approaches for automating the identification of pain by analyzing behavioral or physiological pain signals, or both. These approaches were further divided into the two categories shown in Figure 2.13: single model techniques and multimodal pain analysis.

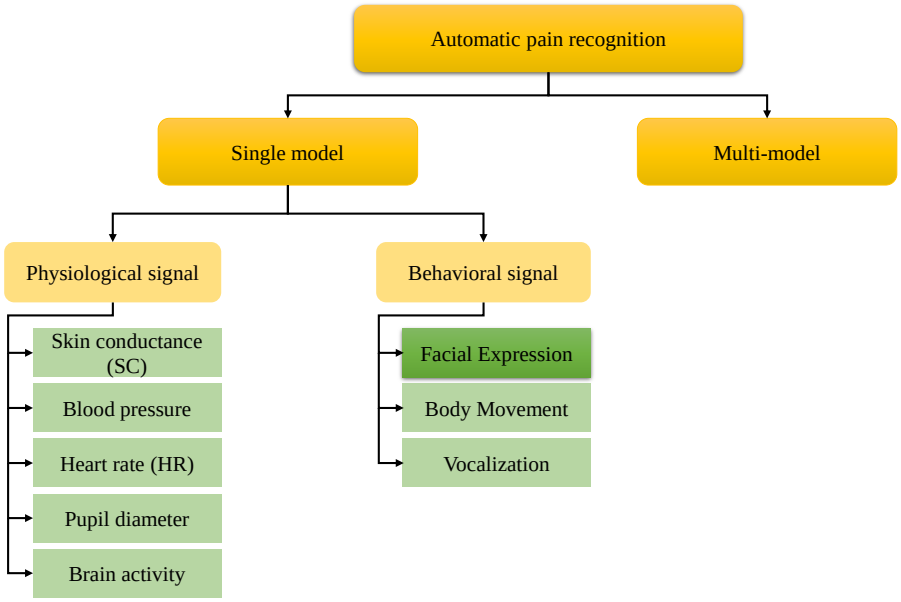

Figure 2.13: Diagram of Automatic Pain assessment methods in two blocks

A basic pain-recognition structure depends just on a single parameter to determine

an individual's pain degree. Physiological and behavioral signs are taken into account. Firstly, to assess pain using physiological measurements is to extract pain-related features from the physiological responses of patients. Variations in vital signs (such as a rapid heartbeat) and brain hemodynamic activity are two examples of these physiological responses (see Figure 2.13). For instance, Electrocardiogram (ECG), Blood Volume Pulse (BVP), and Skin Conductance Level (SCL) are some of the physiological indications that an external electrical input might induce. Based on these various physiological signals, Y. Chu et al [28] devised a new approach for identifying the amount of pain. Their dataset was generated by gathering information from (six-subjects for 7-days). The training set comprised 75% of the samples, whereas the testing set comprised the remaining 25%. They utilized a genetic algorithm (GA) to extract features and principal component analysis (PCA) to minimize the number of extracted features (PCA). They ultimately opted to use linear discriminant analysis (LDA) as a classifier, and then compared its performance to that of KNN and SVM. In Lopez-M. et al [76] study, a method for continuously assessing pain intensity with high temporal accuracy based on skin conductance autonomic data was developed and assessed using the BioVid Heat Pain Dataset [12]. In order to gather measures that more correctly characterize the activity of the sympathetic sudomotor nerve, the LSTM-NN approach begins by deconvoluting the signal into its tonic and phasic components. Then, features extracted from overlapping windows within the deconvolved data are fed into a regression LSTM recurrent neural network. Level 4 reflects the pain tolerance threshold in the dataset. They also explored metrics based on the variability of the heart rate using a point process approach, but found the skin conductance data to be significantly more accurate.

The second type of pain analysis using single model, is the one based on behavioral measures. Which is the examination of pain based on behavioral indicators such as facial expression, vocalizations, physical movements, changes in interpersonal interactions, mental state change, and activity patterns. This article discusses the methods currently available for decoding behavioral data into classification-relevant features. In this section, we focus on approaches that used vocalizations and body movement. Facial expressions' based approaches will be detailed in the next section. In the work of Olugbade et al [92], they aim to explore if pain levels can be recognized by assessing body movement quality during two functional physical activities (sit-to-stand and complete trunk flexion). Using the feature optimization and machine learning methods, it was possible to automatically identify between those with low level pain, those with high level pain, and control participants during physical activity. Support Vector Machines yielded 94% accuracy for complete trunk flexion and 80% accuracy for the sit-to-stand transition. The most promising results came from feature set optimization methods, with Support Vector Machines yielding 94% accuracy for complete trunk flexion and 80% accuracy for the sit-to-stand transition. Due to the association between depression and pain, the authors included depression scores to a standard questionnaire in order to better differentiate between healthy controls and those with pain when uti-

lizing Random Forests. In this study, the authors used the EmoPain dataset. In regard to the employment of vocalizations, this modality was rarely utilized alone. However, the utilization of vocalizations signals includes newborn cries. Vempada et al [92] presented a time-domain method for recognizing distress cries. The recommended method was evaluated using 120 cry corpuses, including 30 from pain, 60 from hunger, and 30 from a wet diaper (30 corpuses). The age of newborns can range from 12 to 40 weeks. Each corpus was recorded using a Sony digital recorder with a 44.1 kHz sample rate. At the stage of feature extraction, two features were computed: 1) The average square of the sample values across a suitable time frame, known as the short-time energy (STE), and 2) The duration of the sobbing section's pauses. A portion of these characteristics were applied in the development of SVM, while the remaining were used to evaluate its performance. Pain cries were recognized by 83.33%, hunger cries by 27.78%, and wet diapers by 61.11%. An overall rate of 57.41% was found for recognition.

The second category of automated pain recognition approaches is multi-model techniques (Figure 2.13). Using this strategy for pain detection, scientists attempt to utilize many physiological and behavioral signs. Combining facial and physiological signals is a novel multi-model approach to pain identification (ECC and SC) proposed in [78]. In this work, the major purpose is to personalize pain estimation, by categorizing patients into various profiles instead of considering each individual. Then, the authors moved to a technique known as multitask NN (MT-NN), which employs a system in which distinct profiles are connected with distinct activities. Their research utilized the publicly available BioVid Heat Pain database. Better performance was seen for high clusters (C = 4), indicating that additional research utilizing a higher number of clusters is required. Kächele et al [60] combined videos of people's faces with physiological indicators such as electromyography, electrocardiography, and skin conductance. Implementing the concept of employing the system in an adaptable manner to evaluate unidentified individuals based on unlabeled data, their technique entails evaluating unidentified individuals. Consequently, they relied on the BioVid Heat Pain database and applied a multi-stage ensemble-based classification algorithm. NNs (Neural Networks) were trained with three distinct inputs: the prediction of one-to-one classifiers, the continuous pain-level estimation of the regressor, and the variance of a bagged ensemble of random forests. Then, by using a random regressor, we may employ NNs to calculate sample confidence. After testing numerous input combinations for the NN, the best correlation coefficient and RMSE values were 0.183 and 0.347, respectively. Individual variation in response to painful stimuli demonstrated that investigating adaptation is challenging and requires additional research.

In addition, Thiam et al. [123] used the publicly available BioVid Heat Pain dataset (Part A) to study many CNN topologies. The signal modalities utilised were electronic data acquisition (EDA), electrocardiography (ECG), and electromyography (EMG). They

conducted experiments with both 1D and 2D network input. Multiple designs for deep fusion were also presented and evaluated. Their respective binary classification results were 84.57% and 84.40%. However, C. Wang et al. [135] use deep learning to identify characteristic chronic pain behaviors. Their primary focus is on analyzing the defensive actions taken by LBP sufferers when they carry out five different exercises. They drew on a preexisting database of sEMG and mobility data (emo-pain data set). Two different types of recurrent neural networks, stacked LSTM and dual-LSTM, were suggested in this research. They calculated the angles and energies of Mockup data based on just five activities. Corrected sEMG data was applied to raw data on muscle activity to eliminate noise and improve readability. The information was divided using a sliding window. They ran separate tests for each activity to find the optimal window size, and then analyzed and selected that size based on how well it overlapped with the others at a set overlapping ratio of 75%. Therefore, they concluded that a window of duration equal to three seconds is optimal for detecting the vast majority of actions. They also discovered that the best results may be achieved by combining the two enhancement strategies. When everything was said and done, their LSTM Networks performed better than a regular neural network by a mean F1 score of 0.815. However, there will be a performance drop as a result of the generalization, and this is an issue that must be addressed.

2.6.2 General Architecture of Automatic Pain Detection System

In most instances, automatic pain identification systems are multi-step, with pain recognition occurring at each stage (input, processing, and output). These steps are frequently implemented using machine learning and computer vision techniques. Beginning with the input step, which describes data collection, Following this, the system entered the processing phase, where feature selection and extraction were performed feature-by-feature. In addition to computing the accuracy rate and receiving the final result, the output stage is responsible for picking the right classifier and obtaining the final result. In order to give insight on the general architecture of the automated pain identification system, each of these phases is described in detail below and in Figure 2.14.

The initial phase of the automatic pain identification system focuses on the individual's physiological and behavioral responses to pain, which are discussed previously in 2.6.1. Digital cameras, cellphones, and microphones collect data on non-physical signals (such as behavioral patterns) and contact sensors (such as wristbands, caps, spectacles, t-shirts, and rings). Next, due to the fact that pain information, once acquired, may not be in a standard format or may be affected by noise, occlusion, and other environmental

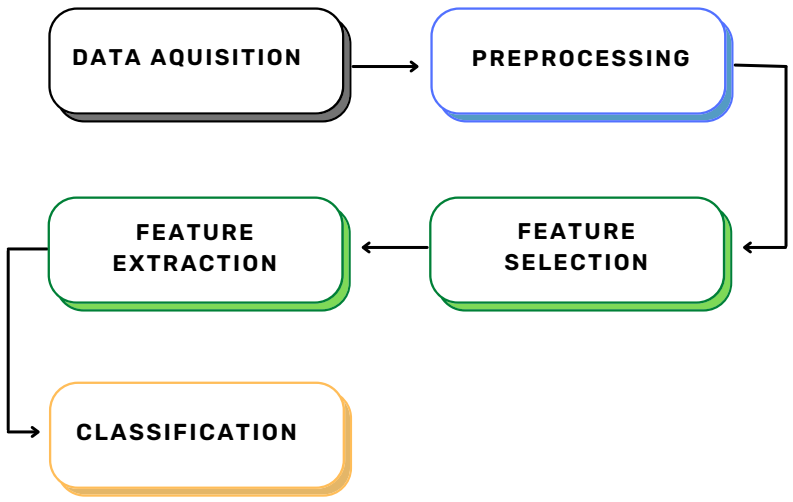

Figure 2.14: General Architecture of Automatic Pain Detection System

factors, preprocessing is an essential step in the automatic pain recognition technique. Before extracting features from the data, it must be normalized and enhanced.The, the most significant phase of an automatic pain recognition system is believed to be the extraction of features. It acts similarly to zooming in on the general feature space, collecting the features that are most closely related, and generating a new space with fewer dimensions that is entirely distinct from the initial space. At this stage, the approach utilized to identify pain will determine the effectiveness of the automatic pain detection system. When the input data is too large to be processed in its entirety, feature selection is the process of evaluating a subset of the original features. The selected features hold the essential information from the input data, allowing the ideal goals to be attained with less training time, less dimensionality, and less overfitting by using only the essential information from the input rather than the complete original data. Finally, upon completion of the Automatic Pain Recognition System, a classifier is employed to sort the data. After the pain-relevant elements have been added to the input, it now begins to determine whether pain is present. Here, any ML approaches, such as K-Nearest Neighbors, Support Vector Machine, Decision Trees, Logistic Regression, and Random Forest, as well as the most popular classification models, can be utilized (Binary, Multi-class, Multi-label, and Imbalanced).

2.6.3 Automatic Pain Recognition from Face

Since facial expressions are such a reliable indicator of pain, all observer-based pain assessment techniques utilize them. Since the discovery of algorithms for automatically evaluating facial expressions, computer vision researchers have sought to apply this knowledge to the problem of automatically identifying pain based on facial expressions. Specifically, this debate focuses on cutting-edge facial expression-based pain recognition techniques. The process flow for pain identification using facial expressions is depicted in Figure 2.15. These techniques' input data could be categorized as frames, time windows, or entire videos. Also, whether it contains RGB or thermal and depth images. In many methods, feature extraction is still an essential step. These features can be divided into three main categories: geometric, textural, and learned. The learning methods are the final step in the facial expression-based pain recognition pipeline. We will list methods in the order in which they were developed.

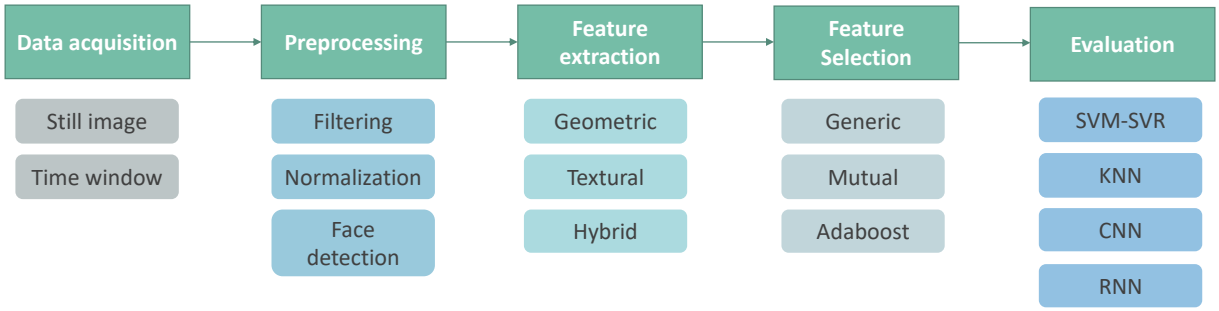

Figure 2.15: Pain recognition from facial expressions general pipeline. It includes examples of used methods in each step.

2.6.3.a Input Data

So far, the great majority of pain identification methods have evaluated facial expression-containing camera photos. As mentioned in Section 2.3, facial expressions play an important role in transmitting pain to others; thus, the majority of early research on automatic pain identification focused on this modality. The introduction of the first public

database for pain identification, the UNBC-McMaster database, which contains facial photos with comprehensive annotation but no other modalities, strengthened this trend. The approaches will be divided into two groups depending on the input data type: the approaches that used frames and studies that used time windows.

Chen et al. [27] used spatial features for pain estimation using frames from the UNBC-McMaster dataset. In the work of Egede et al. [34], the frames of the same dataset were used. The same features have been extracted from the UNBC-McMaster dataset in the works: [77] [102][136]. Works that used the BioVid database for pain estimation from facial expressions only, we can cite Werner et al [139] that used spatial features to train a Random Forest model. In another work by Yang et al [151], they employed spatial features extracted from both UNBC-McMaster and BioVid datsets' frames. Concerning approaches that used times windows, we can cite Lo Presti et al [97] that used Hankel of Haar & Gabor to extract features from time windows using the UNBC-McMaster dataset. In the study by Thiam et al [122] several features were extracted from the video channel and put into a hierarchical fusion architecture in an attempt to increase the overall efficiency of the system. More approaches are cited in table reefff.

2.6.3.b Feature extraction

Images and sequences of images of the face are used to extract facial features that are utilized to characterize the facial shape and appearance, or the changes to these characteristics caused by facial emotions. These qualities may be roughly classified as either spatial or spatio-temporal. The face's shape and texture are examples of spatial characteristics that may be utilized to characterize an image in a definite way. In a sequence of frames, facial shape and appearance may be observed to change over time, and these differences are represented using spatio-temporal properties. As seen above, the spatial features take data frames as input, while the spatio-temporal features take time windows. These features consist of Geometric and textural features. Below, approaches categorized into three groups based on features type: *Geometric* features, *Textural* features and *Hybrid* features (fusion of both geometric and textural features).

Geometric features characterize the form of the face using point-based shape description techniques. They specify the locations of points on the head, including the eyes, eyebrows, cheeks, nose, lips, chin, and/or facial border. Geometric characteristics might be either the specific positions of these facial feature points or higher-order features such as distances and angles between them. Rarely were geometric characteristics

employed alone. Meng and Bianchi-Berthouze [87], Ghasemi et al. [40], Aung et al. [6], Liu et al. [73] and Lopez-Martinez et al. [77] used the facial landmark positions. Romea-Paredes et al. [103] used facial landmark distances. In addition to facial landmark distances, Niese el al. [91] used also angles. However Zafar and Khan [152] added facial landmarks positions.

Concerning spatio-temporal features, they describe changes in spatial features over time. In this case, the geometric features derived from a series of images were summed up using mathematical and statistical procedures. Lo Presti and La Cascia [74] used Hankel matrices based on facial landmark positions and distances. Tsai et al. [126] employed statistical features from sequence of facial landmark distances and quadratic polynomial coefficients of mouth shape.

Textural features describe the appearance of the face and facial features. Textural characteristics are creases and folds that occur on or around the facial structure. Intensity of individual pixels is one example of a texture feature used in the literature; other examples include manually developed or learned features. Widely used examples of hand-crafted textural feature descriptors include Gabor filters, Local Binary Patterns (LBP), and the Histogram of Oriented Gradients (HOG). Brahnam et al. [20], Gholami et al. [41] used pixel intensities. Littlewort et al. [72] [71], Roy et al. [104], Sikka et al. [112] used Gabor filters. Brahnam et al. [19] and Aung et al. [6] used the Discrete Cosine Transform (DCT). The Local Binary Pattern (LBP) features were used in [90] [25] [106]. Chen et al. [26] used the Histogram of Oriented Gradients (HOG) around facial landmarks.

Furthermore, three Orthogonal Planes (TOP) were used to extract spatio-temporal textural characteristics like LBP-TOP and HOG-TOP, with one of the planes covering the temporal dimension that encompasses an ordered succession of frames throughout time. In this study, Yang examined the efficacy of several spatio-temporal textural characteristics, including LBP-TOP, LPQ-TOP, BSIF-TOP, and their combinations. Chen et al. [26] used the HOG from Three Orthogonal Planes (HOG-TOP). Kaltwang et al. [62] employed the LBP from Three Orthogonal Planes (LBP-TOP). Yang et al. [151] used a combination of LBP-TOP, LPQ-TOP and BSIF-TOP. Zhou et al. [156], Egede et al. [34] and Tavakolian and Hadid [119] used deep learned spatio-temporal features.

Hybrid features present the combination of both Geometric and textural features. The use of multiple features involves the employment of feature fusion. Which is either an early or late fusion. For spatial hybrid features, Hammal et al. [47] and Hammal and Kunz [46] used facial landmarks distances, nasal root wrinkles and context variable. Zhao et al. [155] used facial landmark positions with LBP and Gabor filters.

These positions were combined with pixel intensities to extract features in the works of Ashraf et al. [5], Lucey et al. [80] [81] [82]. Egede et al. [34] used the facial landmark positions, distances, angles and HOG features. However, Werner et al. [143] [140] used statistical features from sequence of head pose, facial landmark distances and mean gradient magnitudes. Kachele et al. [61] used LBP-TOP, statistical features from facial distances.

Table 2.2 presents a summary of spatial and spatio-temporal features extracted from facial images for automatic pain detection.

2.6.3.c Recognition models

Deep learning is used directly to evaluate the level of pain based on facial expressions. In Martinez et al. [77] study, estimating pain using the values reported by individuals on a visual analog scale (VAS) is one way of reducing these differences. Their training program consists of two parts. Estimating Prkachin and Solomon pain intensity (PSPI) from face images is the initial task that is learned using recurrent neural networks (RNNs). The individual VAS was then calculated based on the output of the concealed conditional random fields (HCRFs). Compared to methods that weren't personalized, a single-sequence test did the best, and its score was the highest of any other method.

Recent research [102] utilized deep learning to extract pain-related information from facial expressions. The three-step method they employ is as follows: Initially, convolutional neural networks are used to extract features from VGG Faces (CNNs). A LSTM is then trained using the feature map's output. A special RNN utilized for pain estimation in binary form (pain, no pain). Their area under the curve (AUC) performance of 93.3% was the greatest of all previous studies that evaluated their methods on the UNBC-McMaster dataset. In addition to being appropriate to their specific goal, this method may also be applied to the larger challenge of emotion recognition in faces. When their model was used with the Cohn Kanade Plus database of facial expressions, it achieved good results (AUC = 97.2%).

Similarly, Egede, Valstar, and Martinez [34] developed a pain estimating model that used both machine-learned and hand-crafted characteristics. Their motivation was the difficulty of accumulating a large enough data set for pain estimation to utilize deep learning effectively. So, scientists gave a CNN an image of a face and instructed it to acquire particular characteristics based on data collected manually. Their appearance,

geometry, and movement are their features. A linear regression model based on both sets of data was then used to classify individuals' pain reports. Their root-mean-square error (RMSE) and Pearson correlation (CORR) were better than those of cutting-edge methods.

In Wang et al. [136] presented an alternative solution to the problem of deep learning on restricted data sets. They refined a smaller pain dataset using the WebFace dataset, which comprises 500,000 photos of human faces. As a further step, they utilized a regression loss that was regularized with the center loss to adapt the scenario as if it were a regression problem. Instead of evaluating their performance with unbalanced data, different metrics were offered. Weighted measures (mean absolute error (MAE): 0.389, mean squared error (MSE): 0.804, and Pearson's correlation coefficient (PCC): 0.651) and newly proposed metrics (both 0.804 and 0.651) demonstrate that this technique outperformed state-of-the-art methods (weighted MAE 0.991, weighted MSE 1.720).

In contrast to previous methods, the cumulative attribute (CA) methodology was utilized in [58] as an efficient method for correcting the imbalance of data in datasets for pain measurement. Two stages of a CNN with cumulative features are utilized in this study. A trained CNN outputs the cumulative attribute vector in the initial step. Training the regression model to obtain the actual result is the second part. Their research examined a data set of pain estimations and determined the age. In CA–CNN trials, their pain estimation findings were more precise than in earlier tests. In addition to employing a CA layer trained with a log-loss function, it significantly outperforms a CA layer trained with Euclidean loss. Their solution is advantageous since it leverages the CNN framework without any additional annotations. However, constructing an annotated dataset for pain evaluation is essential for solving the vast majority of classification challenges.

In the work proposed by Haque et al. [48] created a new database including RGB, depth, and thermal (RGBDT) images of the face for detection of pain levels in sequences. Their technique to elicitation differs from earlier datasets produced by electrically stimulating healthy individuals. Twenty participants participated in the gathering of data to establish pain recognition based on five degrees of discomfort (0 for no pain and 4 for severe pain). This approach proposes a model that uses spatio-temporal features and deep learning. First, they preprocessed the video frames by cutting just the facial area according to the approach they had previously presented in [14] for RGB photos. Then, they cropped additional depth and thermal pictures using homography matrix codes. Following this, they used deep learning based on two distinct methodologies for separate modalities or their integration. The suggested approach is mostly based on two phases. First, they used 2D-CNN for extraction of frame information and detection of pain. LSTM was then utilized to investigate the temporal relationship between frames

and sequence-level pain recognition.

2.6.4 Conclusion

To summarize, AI and deep learning have the potential to revolutionize pain assessment in medical settings. AI-based systems can accurately identify and manage pain in patients by using facial images and other related data. This chapter has looked at various applications of AI in pain assessment, such as emotion recognition, medical diagnosis, and automatic pain recognition from the face.

However, more research is required to optimize these systems and ensure their dependability and accuracy. We can expect to see more innovative and effective approaches to pain assessment in the future as AI and deep learning continue to advance. The following chapter delves into deep learning architectures proposed for estimating pain level from facial expressions.

	Approach	Temporal information	Features	Objective
	Geometric Features			
Spatial Features	Meng and Bianchi-Berthouze [87]	no	Facial landmark positions	pain no pain
	Aung et al. [6]	no		pain no pain
	Liu et al. [73]	no		continuous pain intensity estimation
	Lopez-Martinez et al. [77]	yes		continuous pain intensity estimation
	Ghasemi et al. [40]	yes		classification of pain intensity
	Romea-Paredes et al. [103]	no	Facial landmark distances	continuous pain intensity estimation
	Niese el al. [91]	no	Facial landmark distances and angles	pain and emotions
	Zafar and Khan [152]	no	Facial landmark distances, angles and positions	pain no pain
Staptiotemporal Features	Lo Presti and La Cascia [74]	yes	Hankel matrices based on facial landmark positions and distances	pain in sequence
	Tsai et al. [126]	yes	statistical features from sequence of facial landmark distances and quadratic polynomial coefficients of mouth shape	classification of pain intensity
	Textural Features			
Spatial Features	Brahnam et al. [20]	no	pixel intensities	pain no pain
	Gholami et al. [41]	no		pain no pain
	Littlewort et al. [72]	yes	Gabor filters	genuine vs posed pain
	Roy et al. [104]	no		classification of pain intensity
	Sikka et al. [112]	yes		pain is sequence
	Brahnam et al. [19]	no	Discrete Cosine Transform (DCT)	pain no pain
	Aung et al. [6]	no		pain no pain
	Nanni et al. [90]	no	Local Binary Pattern (LBP)	Pain states' classification
	Chen et al. [25]	no		pain no pain
	Rudovic et al. [106]	no		pain intensity estimation
	Chen et al. [26]	yes	Histogram of Oriented Gradients (HOG) around facial landmarks	pain in sequence
Staptiotemporal Features	Chen et al. [26]	yes	HOG from Three Orthogonal Planes (HOG-TOP)	pain in sequence
	Kaltwang et al. [62]	yes	LBP from Three Orthogonal Planes (LBP-TOP	continuous pain intensity estimation
	Yang et al. [151]	yes	combination of LBP-TOP, LPQ-TOP and BSIF-TOP	pain no pain
	Zhou et al. [156]	yes	deep learned spatio-temporal features	continuous pain intensity estimation
	Egede et al. [34]	yes		continuous pain intensity estimation
	Tavakolian and Hadid [119]	yes		continuous pain intensity estimation
	Hybrid Features			
Spatial Features	Hammal et al. [47]	yes	facial landmarks distances, nasal root wrinkles and context variable	pain and emotions
	Hammal and Kunz [46]	yes		pain and emotions
	Zhao et al. [155]	yes	facial landmark positions with LBP and Gabor filters	continuous pain intensity estimation
	Ashraf et al. [5]	no	facial landmark positions and pixel intensities	pain no pain
	Lucey et al. [80]	no	facial landmark positions, distances, angles and HOG features	pain no pain
	Egede et al. [34]	yes		continuous pain intensity estimation
Staptiotemporal Features	Werner et al. [143]	no	statistical features from sequence of head pose, facial landmark distances and mean gradient magnitudes	continuous pain intensity estimation
	Kachele et al. [61]	yes	LBP-TOP, statistical features from facial distances	continuous pain intensity estimation

Table 2.2: Summary of spatial and spatio-temporal approaches classified according to features type. The use of temporal information in the input and the objective of each approach are also highlighted in the table.

Chapter 3

Automatic Pain Estimation from Facial Expressions : A Comparative Analysis Using Off-the-Shelf CNN Architectures

3.1 Introduction

The human face is a rich source for non-verbal information regarding our health [7]. Facial expression [16] can be considered as a reflective and spontaneous reaction of painful experiences. Most previous studies on facial expression are based on the Facial Action Coding System (FACS), which describes expressions by elementary Action Units (AUs) based on facial muscle activity. More recent works have mostly focused on the recognition of facial expressions linked to pain using either conventional machine learning (ML) or deep learning (DL) models. The studies showed the potential of conventional ML and DL models for pain estimation, especially when the models are trained on large fully annotated datasets, and tested under relatively controlled capturing conditions. However, the accuracy, robustness, and complexity of these models remain an issue when applied to real-world pain intensity assessment.

This chapter provides a comprehensive analysis on automatic pain intensity assessment from facial expressions using Off-the-Shell CNN architectures, including MobileNet, GoogleNet, ResNeXt-50, ResNet18, and DenseNet-161. The choice of these CNN architectures is motivated by their good performance in different vision tasks, as shown in the ImageNet Large Scale Visual Recognition Challenge [68]. These architectures have been trained on more than a million images to classify images into 1000 object categories. We use these networks in two distinct modes: stand-alone mode or feature extraction mode. In stand-alone mode, the networks are used for directly assessing the pain. In feature extraction mode, the features in the middle layers of the networks are extracted and used as inputs to classifiers, such as SVR (Support Vector Regression) and RFR (Random Forest Regression). We perform extensive experiments on the benchmarking and publicly available database called UNBC-McMaster Shoulder Pain Expression Archive Database [81], containing over 10,783 images. The extensive experiments showed interesting insights into the usefulness of the hidden layers in CNN for automatic pain estimation from facial expressions.

The main contributions of this chapter include:

- We provide a comprehensive analysis on automatic pain intensity assessment from facial expressions using 5 popular and Off-the-Shell CNN architectures.
- We compare the performance of these different CNNs (including MobileNet, GoogleNet, ResNeXt-50, ResNet18, and DenseNet-161).
- We study the effectiveness of the hidden layers in these 5 Off-the-Shell CNN ar-

chitectures for pain estimation by using features as inputs to two classifiers: SVR (Support Vector Regression) and RFR (Random Forest Regression).

- We provide extensive experiments on a benchmarking and publicly available database called UNBC-McMaster Shoulder Pain Expression Archive Database [81], containing 10,783 images.

The rest of the chapter is organized as follows. Section 3.3 presents the different CNN architectures that are considered in this work, as well as our proposed framework. Section 3.4 describes the conducted experiments and the obtained results. Finally, Section 3.5 draws some conclusions and remarks.

3.2 Related Work

Automatic pain recognition from facial expressions has been widely investigated in the literature. The first task is the detection of the presence of pain (a binary classification). Some other works are not limited to binary classification, but are mainly focused in assessing the pain level intensity. These works commonly use the Prkachin and Solomon's Pain Intensity metric (PSPI) [98]. This can be calculated for each individual video frame, after coding the intensity of certain action units (AU) according to the Facial Action Coding System (FACS). Below, we review some existing works.

Several approaches have been proposed for pain recognition as a binary classification problem, aiming at discriminating between pain versus no pain expressions. For instance, Chen et al. [27] proposed a new framework for pain detection in videos. To recognize facial pain expression, the authors used Histogram of Oriented Gradients (HOG) as frame level features. Then, they trained a Support Vector Machine (SVM) classifier. Lucey et al. [81] addressed AUs (Action Units) and pain detection based on SVMs. They detected the pain either directly using image features or by applying a two-step approach, where first AUs are detected, and then this output is fused by Logistical Linear Regression (LLR) in order to detect pain.

Other approaches focused on the estimation of pain level. Many of them are based on variants of machine learning methods. For instance, Lucey et al. [82] used SVM to classify three levels of pain intensity. In another study, four pain levels were identified using SVM classifier to estimate the pain intensity by Hammal and Cohn [45].

Chen et al. [27] proposed a framework for pain detection in videos, exploring spatial and dynamic features. These features are then used to train an SVM as a frame-based pain event detector. Recent work by Tavakolian et al. [120] also used a machine learning framework, namely a Siamese network. The authors proposed a self-supervised learning to estimate pain. In this work, the authors introduced a new similarity function to learn generalized representations using a Siamese network. The learned representations are fed into a fine-tuned CNN to estimate pain. The evaluation of the proposed method was done on two datasets (the UNBC-McMaster and the BioVid), showing very good results.

The recent works have shown considerable interest in automatic pain assessment from facial patterns using deep learning algorithms. Transfer learning was adopted by various image classification works. For instance, Bargshady et al. [12] propose to extract feature using the pre-trained CGG-Face model. Their approach consists of a hybrid deep model, including two-stream convolutional neural networks related to long short-term memory (CNN-BiLSTM). A pre-trained convolutional neural network (CNN) (VGG-Face) and long short-term memory (LSTM) algorithm were applied to detect pain from the face using the MIntPAIN dataset by Haque et al. [48]. In this work, a hybrid deep learning approach is employed. Actually, the combination of a CNN and an RNN allowed the use of spatio-temporal information of the collected data for each of the modalities (RGB, Depth, and Thermal). In this study, fusion strategies (early and late) between modalities were employed to investigate both the suitability of individual modalities and their complementarity.

Another study also extracted facial features using a pre-trained VGG-Face network [102]. These features are integrated into an LSTM to deploy the temporal relationships between the video frames. The study of Tavakolian et al. [119] aimed to represent the facial expressions as a compact binary code for classification of different pain intensity levels. They divided video sequences into non-overlapping segments with the same size. After that, they used a Convolutional Neural Network (CNN) to extract features from each segment. From these features, they extracted low-level and high-level visual patterns. Finally, the extracted patterns are encoded into a single binary code using a deep network. In reference [11], the authors used two different Recurrent Neural Networks (RNN), which were pre-trained with VGGFace-CNN and then joined together as a network for pain assessment. Recent work of Huang et al. [57] proposed a hybrid network to estimate pain. In this paper, the authors proposed to extract multidimensional features from images. They extracted three type of features : spatio-temporal features using 3D convolutional neural networks (CNN), spatial features using 2D CNN, and geometric information with 1D CNN. These features are then fused together for regression. The proposed network was evaluated on the UNBC McMaster Shoulder dataset. Table 3.1 summarizes the mentioned state-of-the-art works, the corresponding methods,

Table 3.1: Summary of some previous works on pain estimation.

Approach	Method	Metrics	Database
Rodriguez et al., 2017 [102]	Features: VGG-16 Model: LSTM	AUC 93.3% Accuracy 83.1% MSE 0.74	UNBC-McMaster Shoulder Pain
Haque et al., 2018 [48]	Features: VGG-face Model: LSTM / FF, DF	Mean frame Accuracy 18.17%	MIntPAIN
Tavakolian et al., 2018 [119]	Features: CNN Model: deep binary encoding network	MSE 0.69 PCC 0.81	UNBC-McMaster Shoulder Pain
Bargshady et al., 2019 [11]	Features: VGG-face Model: RNN	MSE 0.95 Accuracy 75.2%	UNBC-McMaster Shoulder Pain
Tavakolian et al., 2020 [120]	Unsupervised learningModel: Siamese network	MSE 1.03 PCC 0.74	UNBC-McMaster Shoulder Pain andBioVid
Bargshady et al., 2020 [12]	Feature: VGG-faceModel: EJH-CVV-BiLSTM	AUC 88.7% Accuracy 85% MSE 20.7 MAE 17.6	UNBC-McMaster Shoulder Pain 10783 images
Huang et al., 2021 [57]	Features: spatiotemporal, spatial features andgeometric information Model: Hybrid network	MAE 0.40 MSE 0.76 PCC 0.82	UNBC-McMaster Shoulder Pain
Tavakolian et al., 2019 [118]	3D deep architecture SCN (Spatiotemporal Convolutional Network)	MSE 0.32 PCC 0.92	UNBC-McMaster Shoulder Pain
Lucey et al., 2011 [81]	Features: shape, SAPP, CAPP Model: SVM	AUC 83.9%	UNBC-McMaster Shoulder Pain
Hammal et al., 2012 [45]	Features: log-normal filters Model: SVM	Recall 61% F1 57%	UNBC-McMaster Shoulder Pain
Chen et al., 2017 [27]	Features: HOG, HOG-TOP Model: SVM	Accuracy 91.37% F1 Score 0.542	UNBC-McMaster Shoulder Pain
Lo Presti et al., 2017 [97]	Features: Hankel of Haar and GaborModel: AdaBoost	Accuracy 74.1%	UNBC-McMaster Shoulder Pain

performances, and used databases.

3.3 Proposed Framework for Studying Pain Assessment Using Deep Features

Automatic pain recognition from facial expressions is a challenging problem that has attracted a significant attention from the research community. This chapter provides a comprehensive analysis on the topic by comparing some popular and Off-the-Shell CNN (Convolutional Neural Network) architectures. We start by showing the importance of feature extraction. We present then the framework used in this study. Finally, we present the used CNN architectures.

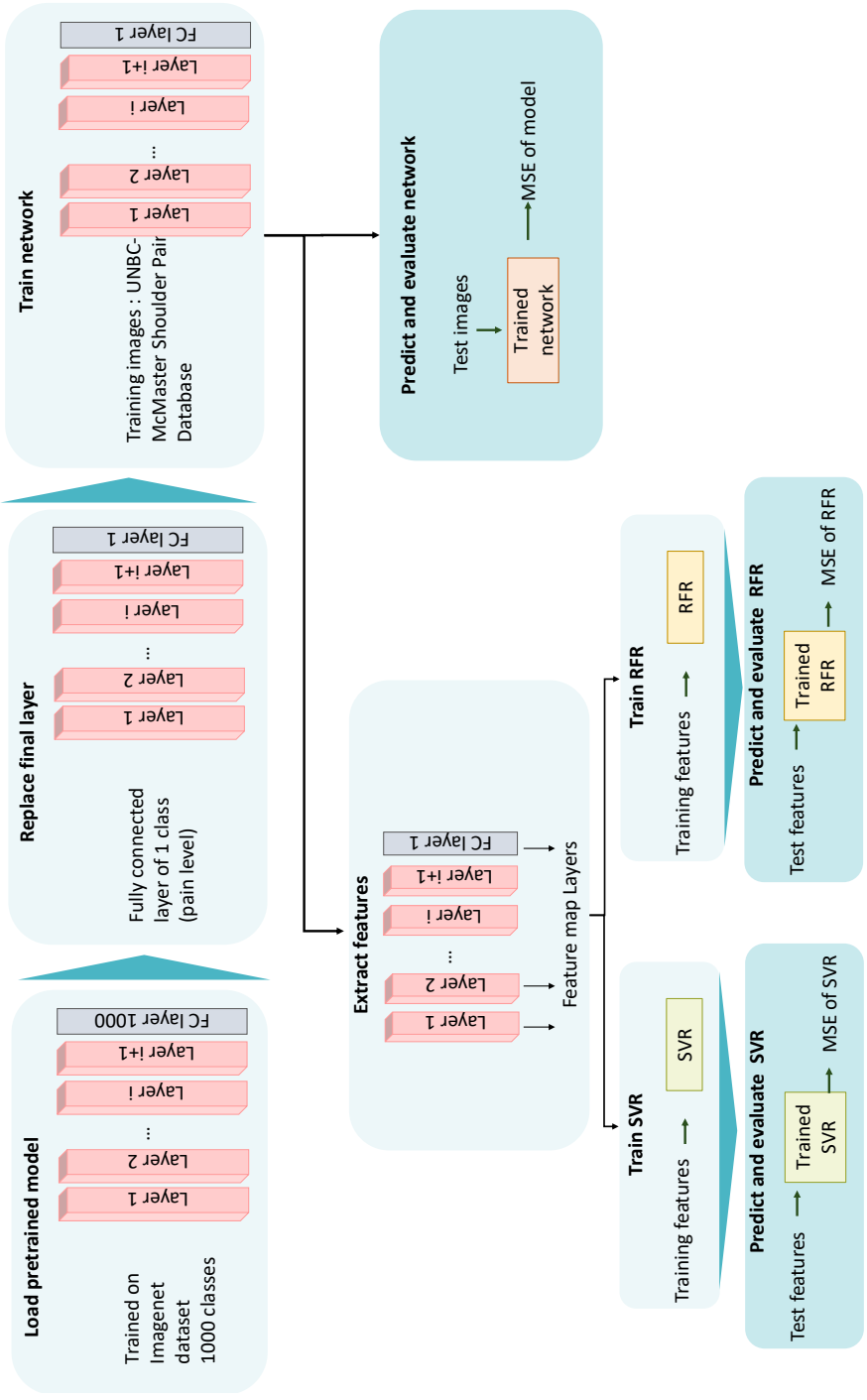

Figure 3.1: Proposed framework for pain estimation.

3.3.1 Feature extraction

Features represent a piece of relevant information to solve computational tasks concerning a specific application. In computer vision, features represent the region of image with important information. These features might be specific shapes in the image, like edges, points, blobs or objects. In pain studies, feature extraction is a commonly used method. Moreover, features can be split into three categories: generic features, hand-crafted features and learned features. Generic features are based on ideas that proved success in other domains, and are used for pain tasks. For example, Local Binary Pattern (LBP)[1] and mel-frequency cepstrum (MFC)[75]. Hand-crafted features are manually designed features by expert. These extracted features should be robust to the variances in the objects. Examples are Scale-invariant feature transform (SIFT)[79] and Histogram of Oriented Gradients (HOG)[138]. Concerning learned features, they are obtained automatically from a machine learning algorithm. We find most of the deep learning approaches in this category.

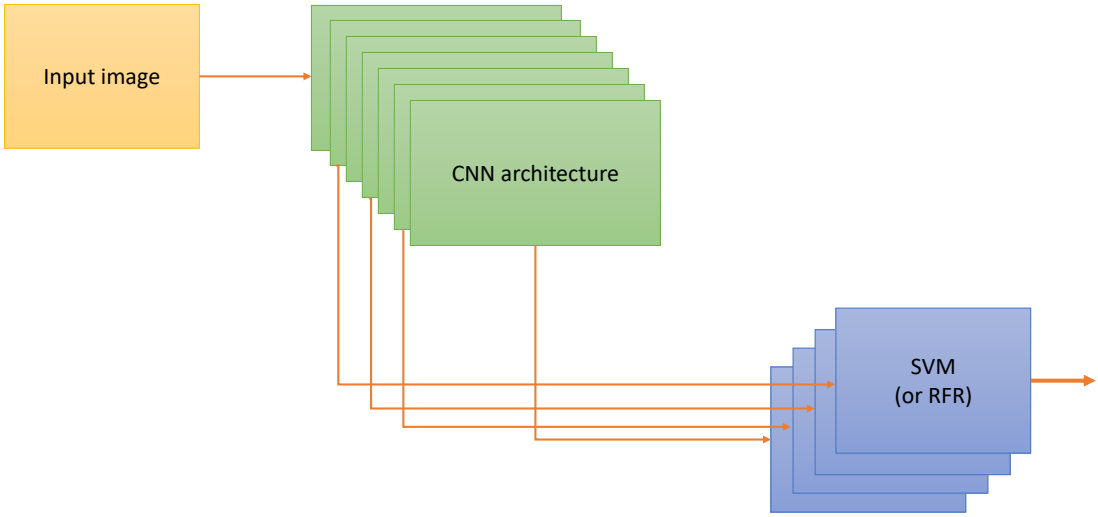

Figure 3.2: Feature extraction from multiple inner layers of Convolutional Neural Networks (CNNs). Then these features are used individually to train separate Support Vector Machines (SMMs).

There are different approaches for automatic pain recognition. Each one depends on special input modality and sensors. Therefore, features depend on approach category. First, for camera-based approaches, different features to detect facial pain expressions are used. Such as, Gabor, Local Binary Pattern (LBP), Histogram of Oriented Gradients (HOG), Discrete Cosine Transform (DCT), facial distances in 2D and other deep features. Second, for audio approaches, mostly the Mel Frequency Cepstral Coefficients (MFCC) are the used features. Other features may include Linear Predictive Coding (LPC) coefficient and Relative Spectral Perceptual Linear Predictive (RASTA-PLP). Third, for contact-Sensor approaches that contain for example EDA, ECG or sEMG signals, there is a variety of Time series Statistics Descriptors (TSD) features.

In the case of our study, we used facial expressions as an input. We extracted deep features from Convolutional Networks (CNNs). The features are extracted from different layers of a CNN and then individually trained on separate SVMs and RFRs as shown in figure 3.2. The CNN architectures considered in this work are MobileNet-v2, GoogleNet, ResNeXt-50, ResNet18, and DenseNet-161. More details about the framework and the used CNN architectures are presented in the sections below.

3.3.2 Framework presentation

Recently, transfer learning is increasingly applied for feature extraction [116], especially in computer vision [2]. It consists of adopting prior knowledge that has been previously learned in other tasks. Our studied models are also heavily based on transfer learning and fine-tuning. Our methodology consists of exploring popular and Off-the-Shell CNN architectures, including MobileNet, GoogleNet, ResNeXt-50, ResNet18, and DenseNet-161. We use these networks in two distinct modes: stand-alone mode or feature extraction mode. In stand-alone mode, the models are fined-tuned and used for directly estimating the pain. In feature extraction mode, the features from 10 different layers are extracted and used as inputs to SVR (Support Vector Regression) and RFR (Random Forest Regression) classifiers. The model automatically extracts the most discriminative facial features from the training data. These features do not necessary have a clear visual interpretation, but they represent parts of the face that are more important for pain discrimination.

As mentioned above, the recognition models used in this study are the Support Vector Regression (SVR) and Random Forest Regression (RFR). These models represent an essential component in the architecture after the feature extraction. Their role is to map the features to the latent pain state. We briefly review Support Vector Regression (SVR)

and Random Forest Regression (RFR) below.

- **Support Vector Regression :**

  Support Vector Regression is a supervised learning algorithm used for regression problems. It uses the same principles as Support Vector Machine (SVM) with some modifications. Considering the regression problem that consists of finding a function that approximates mapping from an input domain to real numbers. The figure below presents an example of how SVR works. In the figure 3.3, the two red lines are the decision boundaries and the green one is the hyperplane. The idea behind SVR is to consider the points that are withing the decision boundary lines. Therefore, the best fit line is the hyperplane with a maximum number of points.

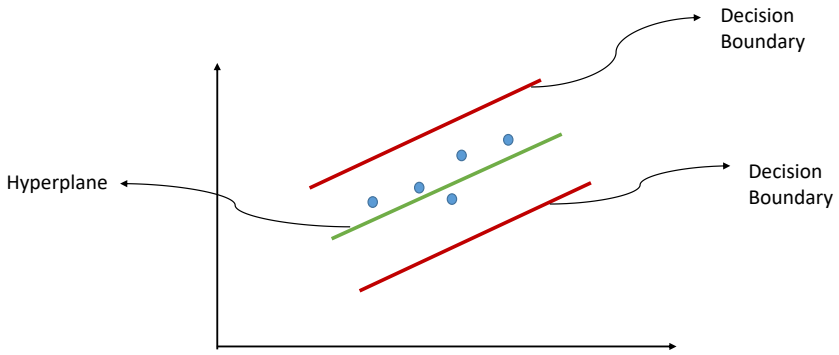

Figure 3.3: Illustrative example of Support Vector Regression (SVR)

Considering the pain intensity estimation during validation is expressed by

$$y = f(x, \theta) \tag{3.1}$$

where θ is the parameter of f and y is the ground truth value. Moreover, we consider a pair $\{y_i, x_i\}$ of ground truth and label used by our SVR to learn. Therefore, the model learns the parameter $\theta = \{w, b\}$ by solving the following optimization problem [155].

$$\begin{gathered} min(\frac{1}{2}) \parallel W^2 \parallel + \gamma \sum (\eta_i^+ + \eta_i^-) \\ s.t. W^T \phi(X_i) + b - y_i (\leq \epsilon + \eta_i^+) \\ y_i - W^T \phi(X_i) - b (\leq \epsilon + \eta_i^-) \\ \eta_i^+, \eta_i^- \geq 0, \forall i \end{gathered}$$

where $\phi : X \mapsto F$ is a mapping from inputs to features, ϵ is a constant that stands for the maximum deviation allowed for a prediction to be considered correct and γ is a constant balancing between the regularization and regression loss.

- **Random Forest Regression :**

  Random Forest Regression is an approach for supervised learning that belongs to the family of ensemble learning. The result is the mean of all the separate decision trees. The technique is based on decision trees, where each internal node reflects a characteristic of the input data and the leaf node represents the result. The decision tree is a straightforward yet effective classification and regression technique.

  Random Forest Regression is a mathematical process based on the concept of decision trees. Each decision tree in the forest is trained using a random subset of both the input data and the features. The objective is to minimize the prediction variance, which is accomplished by averaging the predictions of all decision trees. The final forecast is generated by averaging all the different forecasts.

  Consider, formally, a dataset having N occurrences and M characteristics. Before training each decision tree, the Random Forest Regression algorithm randomly selects n instances (n<N) and m features (m<M) from the dataset. Then, for each decision tree, it identifies the split point that maximizes variance reduction and recursively divides the dataset into smaller subsets until a stopping requirement is met. The ultimate output of the algorithm is the mean of all the individual forecasts from the decision trees, which is calculated as follows:

$$y_{pred} = \frac{1}{T} \times \sum_{i=1}^{T} (y_i) \tag{3.2}$$

  Where T is the number of decision trees in the forest, y_i is the output of the i-th decision tree, and y_{pred} is the final predicted output.

  We used Random Forest Regression technique in our study because it is well-known for its resistance to overfitting, its high level of accuracy, and its capacity to manage huge datasets that contain a significant number of characteristics.

The considered 5 models were originally trained on the Imagenet dataset, containing 1000 classes [32]. We adapted these models to our task of pain estimation by replacing the final layer in each network with a fully connected layer with one class instead of 1000 classes. Figure 3.1 illustrates our proposed methodology.

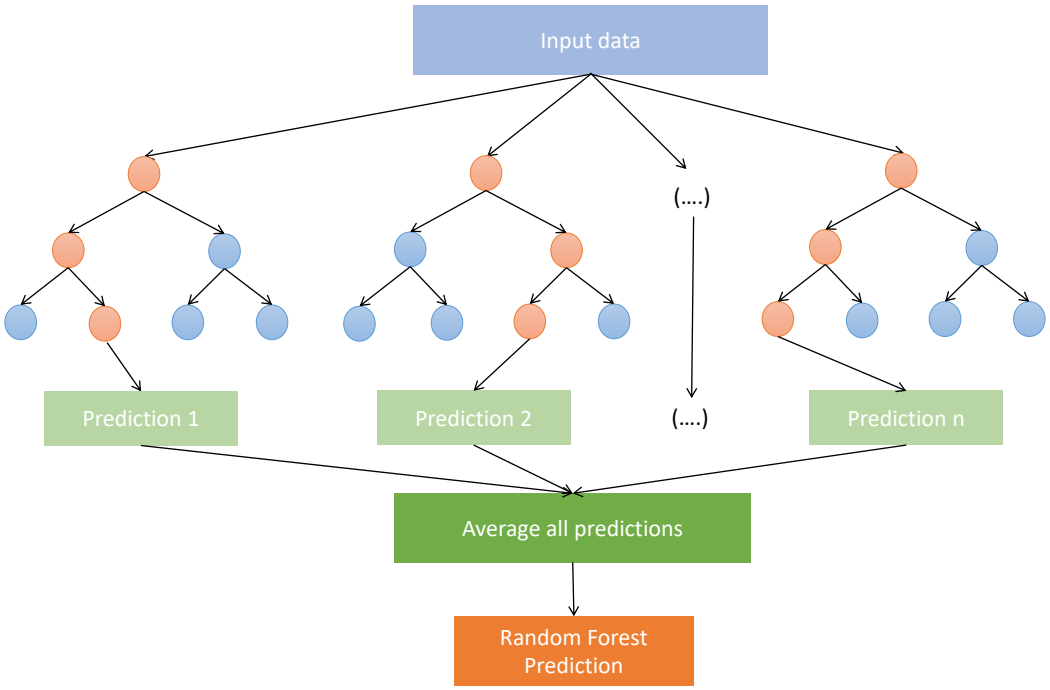

Figure 3.4: Random Forest Regression structure relying on ensemble learning

3.3.3 Used CNN architectures

As seen above, there are various types of features. Our proposed method relies on Convolutional Neural Networks (CNNs). Below, we give a short description of each of the five CNN architectures that we used in our experiments. Which are : MobileNet_v2, GoogleNet, ResNeXt-50, ResNet18, and DenseNet-161.

- **MobileNet_v2** [109]: MobileNet_v2 is a lightweight CNN architecture that is designed to run efficiently on mobile devices. It uses depth-wise separable convolutions to reduce the computational complexity of the model. The depth-wise separable convolution consists of two steps: a depth-wise convolution that applies a single convolution filter to each input channel, and a point-wise convolution that combines the output of the depth-wise convolution. This architecture is designed to reduce the number of parameters and the computation required for each layer, making it suitable for real-time applications with limited computational resources. The architecture of MobileNet_v2 (Table 3.3)contains a series of layers that are organized in a linear structure, starting with a series of convolutional layers followed by multiple depth-wise separable convolution layers and ending with a fully connected layer.

Input	Operator	Output
$h \times w \times k$	1×1 conv2d , ReLU6	$h \times w \times (tk)$
$h \times w \times (tk)$	3×3 dwise s = s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times (tk)$	linear 1×1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

Table 3.2: Bottleneck residual block transforming from k to k' channels, with stride s, and expansion factor t, height h, and width w [109].

Input	Operator	t	Output Channels	Repeat	Stride
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1×1	-	1280	1	1
$7^2 \times 1280$	avgpool 7×7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1×1	-	k	-	

Table 3.3: MobileNet_v2 architecture [109].

- **GoogleNet** [115]: GoogleNet, also known as Inception-v1, is a CNN architecture that was developed by Google in 2014. It uses a combination of convolutional and pooling layers, as well as an auxiliary classifier, to improve the performance of the model. The architecture of GoogleNet is based on the concept of Inception modules (Figure 3.5), which are a combination of convolutional and pooling layers that are designed to extract features at different scales. The Inception module is followed by a series of convolutional layers, and an auxiliary classifier is added to the architecture to improve the performance. GoogleNet has a larger number of parameters compared to MobileNet_v2, but still considered a lightweight model. Figure 3.5a presents the naive version of Inception layer. Szegedy et al. [115] found that this naive form covers the optimal sparse structure, but it does it very inefficiently. To overcome this problem, the authors proposed the form shown in Figure 3.5b. It consists of adding 1×1 convolutions [115].

- **ResNeXt-50** [149]: ResNeXt-50 is a CNN architecture that was developed by

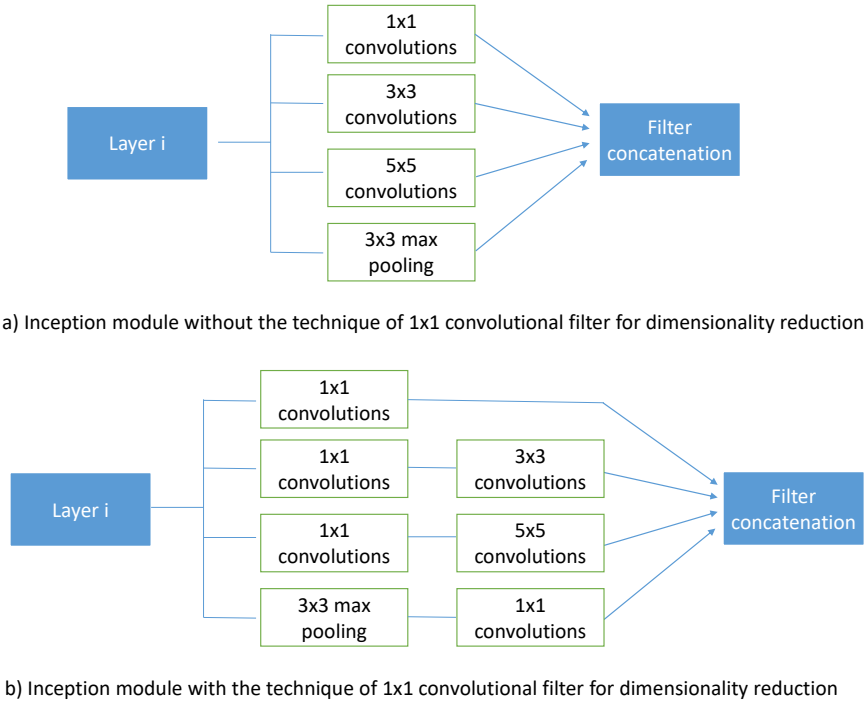

Figure 3.5: Inception module [115].

Facebook in 2016. It uses a new type of building block, called the ResNeXt block, which improves the performance of the model by increasing the depth and width of the network. The ResNeXt block is based on the concept of grouped convolutions, where multiple convolutional filters are applied to the input data in parallel, and the output is concatenated. This architecture is designed to increase the number of filters applied to the input data, which increases the expressive power of the model. The ResNeXt block is followed by a series of convolutional layers, and the architecture ends with a fully connected layer. The architecture of ResNeXt-50 is given in Table 3.4. In our present work, we consider ResNeXt-50 architecture constructed with a template of cardinality = 32 and bottleneck width = 4d.

- **ResNet18** [49]: ResNet18 is a CNN architecture that was developed by Microsoft in 2016. It uses a new type of building block, called the residual block, which improves the performance of the model by making it easier to optimize. The residual block is based on the concept of identity shortcuts, where the input data is added to the output of the convolutional layers. This architecture is designed to increase the depth of the network while maintaining the performance of the model. The residual block is followed by a series of convolutional layers, and the architecture

ends with a fully connected layer. The details of the architecture of ResNet18 are given in Table 3.4.

Layer Name	Output Size	Resnet18	ResNeXt-50 (32×4d)
conv1	112×112	7×7, 64, stride2	7×7, 64, stride2
conv2	56×56	3×3 max pool, stride 2 $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	3×3 max pool, stride 2 $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128, C = 32 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256, C = 32 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
conv4	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512, C = 32 \\ 1 \times 1, 1024 \end{bmatrix} \times 2$
conv5	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 1024 \\ 3 \times 3, 1024, C = 32 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
average pool	1×1	7×7 average pool	global average pool
fully connected	1000	512×1000 fully connections	2048×1000 fully connections
softmax	1000		

Table 3.4: ResNet18 and ResNeXt-50 architectures [49] [149].

- **DenseNet-161** [56]: DenseNet-161 is a CNN architecture that was developed by Gao Huang in 2016 and looks to overcome the problem of CNNs when they go deeper. This is because the path for information from the input layer until the output layer (and for the gradient in the opposite direction) becomes too large.It uses a new type of building block, called the dense block, which improves the performance of the model by increasing the depth and width of the network. The dense block is based on the concept of feature reuse, where the output of the previous layer is concatenated with the input of the next layer. This architecture is designed to increase the number of filters applied to the input. In our work, we used the pre-trained architecture of DenseNet-161 on the ImageNet challenge database [68]. The architecture of DenseNet-161 is illustrated in Table 3.5.

The five CNN architectures covered in this chapter, MobileNet v2, GoogleNet, ResNeXt-50, ResNet18, and DenseNet-161, are all well-known options for image classification and computer vision tasks. MobileNet v2 uses depth-wise separable convolution to reduce

computational complexity. GoogleNet uses a combination of convolutional and pooling layers and an auxiliary classifier. ResNeXt-50 uses a new type of building block called the ResNeXt block. ResNet18 uses a new type of building block called the residual block. Each design has its own advantages and disadvantages, and in the next chapter we will conduct experiments to demonstrate which one of them is more suitable for our task.

Layer Name	**Output Size**	**DenseNet-161**
Convolution	112×112	$7 \times 7 \times$ conv, stride2
Pooling	56×56	7×7 max pool, stride 2
Dense Block 1	56×56	$\begin{bmatrix} 1 \times 1\text{conv} \\ 3 \times 3\text{conv} \end{bmatrix} \times 6$
Transition Layer 1	56×56	1×1 conv
	28×28	2×2 average pool, stride 2
Dense Block 2	28×28	$\begin{bmatrix} 1 \times 1\text{conv} \\ 3 \times \text{conv} \end{bmatrix} \times 12$
Transition Layer 2	28×28	1×1 conv
	14×14	2×2 average pool, stride 2
Dense Block 3	14×14	$\begin{bmatrix} 1 \times 1\text{conv} \\ 3 \times 3\text{conv} \end{bmatrix} \times 36$
Transition Layer 3	14×14	1×1 conv
	7×7	2×2 average pool, stride 2
Dense Block 4	7×7	$\begin{bmatrix} 1 \times 1\text{conv} \\ 3 \times 3\text{conv} \end{bmatrix} \times 24$
Classification Layer	1×1	7×7 global average pool
	7×7	1000D fully connected, softmax

Table 3.5: DenseNet-161 architecture [56].

3.4 Experimental Analysis

In the experimental analysis section of this article, we present a comprehensive evaluation of various CNN architectures for automatic pain recognition from facial expressions. We compare the performance of popular models such as MobileNet, GoogleNet, ResNeXt-50, ResNet18, and DenseNet-161 in both stand-alone mode and feature extrac-

tor mode. The experimental data used in this study is taken from the UNBC-McMaster Shoulder Pain database, and the results and discussion of our findings provide valuable insights into the usefulness of the hidden CNN layers for automatic pain estimation. This section is divided into three subsections: experimental data, experimental setup and results and discussion.

3.4.1 Experimental Data

The problem with unbalanced databases is that they can cause bias in the training and evaluation of machine learning models, as the model may be more likely to predict the majority class, even when the input belongs to a minority class, leading to poor performance in recognizing the minority class, and a lower overall accuracy of the model. This can lead to a situation where the model is not useful in real-world scenarios, where the minority class is important. Unbalanced datasets can also cause an overfitting problem, where the model becomes too specialized to the majority class, and fails to generalize to the minority class. In many real-world applications, it is important to have a balance in the dataset to ensure that the model can perform well on all classes.

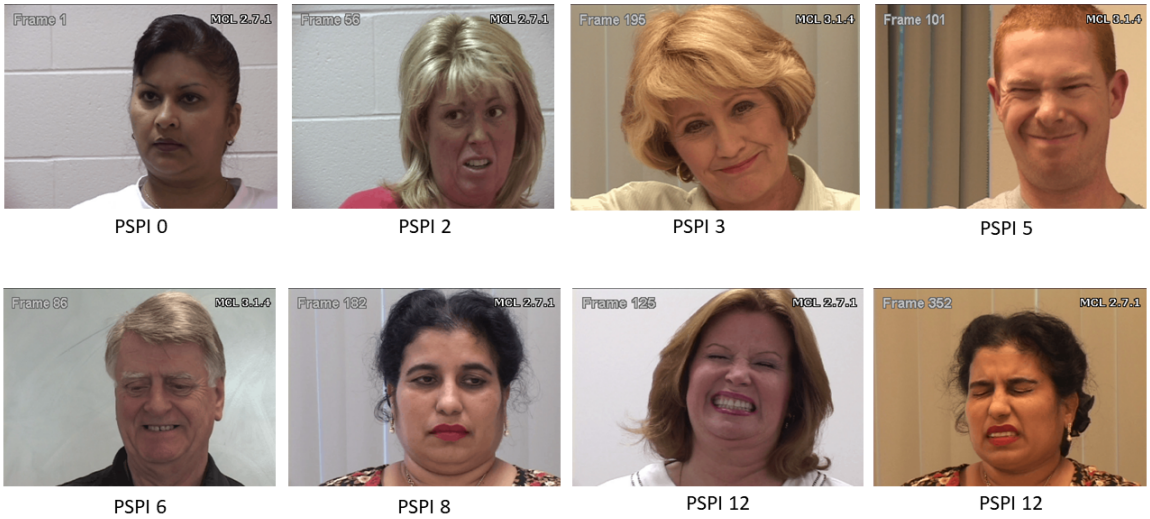

Figure 3.6: Sample images from UNBC McMaster dataset [81].

In our experimental analysis, we considered the benchmark and publicly available

UNBC McMaster dataset. The dataset is composed of 200 sequences of 25 subjects, with a total of 48,398 images. Figure 3.6 shows some images indicated by PSPI. The Prkachin and Solomon Pain Intensity (PSPI) [98] represents the scale for facial expression, which is associated to the UNBC McMaster dataset. The problem of unbalanced datasets is also present in the UNBC McMaster database. As illustrated in Figure 3.7, more than 80% of the database has a PSPI score of zero (meaning "no pain"). To overcome this imbalanced data problem, we balance the databaset applying under resampling technique to decrease the no-pain class. The same protocol is used by Bargshady et al. in Reference [12]. While every sequence starts and ends with a no-pain score, we excluded those two parts for every sequence. As a result, a total of 10,783 images were obtained and used in the experiments. Figure 3.7 illustrates the amount of PSPI for each pain level after data balancing.

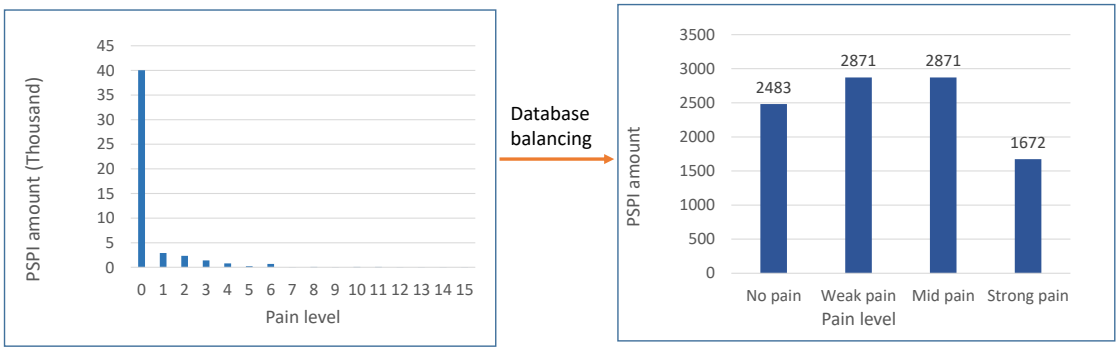

Figure 3.7: Amount of PSPI for each pain level on both balanced and imbalanced UNBC McMaster dataset.

3.4.2 Experimental setup

For evaluation, we used the Leave-one-subject-out-cross-validation. As the balanced database has the same number of subjects, we obtain 25 feature vectors for each layer. We used data augmentation to further increase the size of the dataset. In all the experiments, we used the MSE (Mean Square Error) as the loss function, and Adam as the optimizer. Moreover, we fixed the number of epochs to 200. This setup was kept similar for all the models through all experiments. To measure the performance of the models, we calculate the Mean Square Error metric. Mean Squared Error (MSE) is a commonly used metric for evaluating the performance of regression models. It is a measure of the

difference between the predicted value and the true value of a variable, and it is calculated as the average of the squared differences between the predictions and the true values. In other words, is defined as the sum of squares of prediction errors which is ground-truth (y) value minus predicted value (y') and then divided by the number of data points (N). The formula of the MSE is defined in Equation (3.3).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - y_i')^2. \tag{3.3}$$

The MSE is always a non-negative value, with zero indicating a perfect match between the predicted values and the true values. The lower the MSE value, the better the model's performance. MSE is a commonly used metric in regression problems because it is sensitive to outliers, penalizes large errors, and is differentiable, making it suitable for optimization algorithms. MSE is also a natural choice when the errors are Gaussian; it can be used with any types of data, and it is easy to interpret since it has the same units as the response variable.

3.4.3 Results and Discussion

The UNBC-McMaster Shoulder Pain Database was utilized as the dataset for our experiments, as previously discussed. Each of the five CNN models, MobileNet v2, GoogleNet, ResNeXt-50, ResNet18, and DenseNet-161, had the images from the dataset fed into their feature extraction pipelines. As an evaluation metric, Mean Squared Error (MSE) was utilized to assess the performance of the models. In two distinct modes, the MSE results were reported: standalone mode and feature extractor mode.

3.4.3.a Results and Comparisons

In standalone mode, the entire network (i.e., the model) is utilized to directly estimate the image's pain level. This mode allows us to evaluate the performance of the model based on its ability to directly predict pain levels.

In feature extractor mode, features are extracted from 10 different layers of pretrained CNN models and used as inputs for Support Vector Regression (SVR) and Random Forest Regression (RFR) classifiers. This mode permits the evaluation of the model's

performance based on its ability to extract useful features from the images, which can be fed to other classifiers to predict the pain level. This mode is useful when the objective is to use pre-trained models as a feature extractor in a larger pipeline, as opposed to directly making predictions with the model.

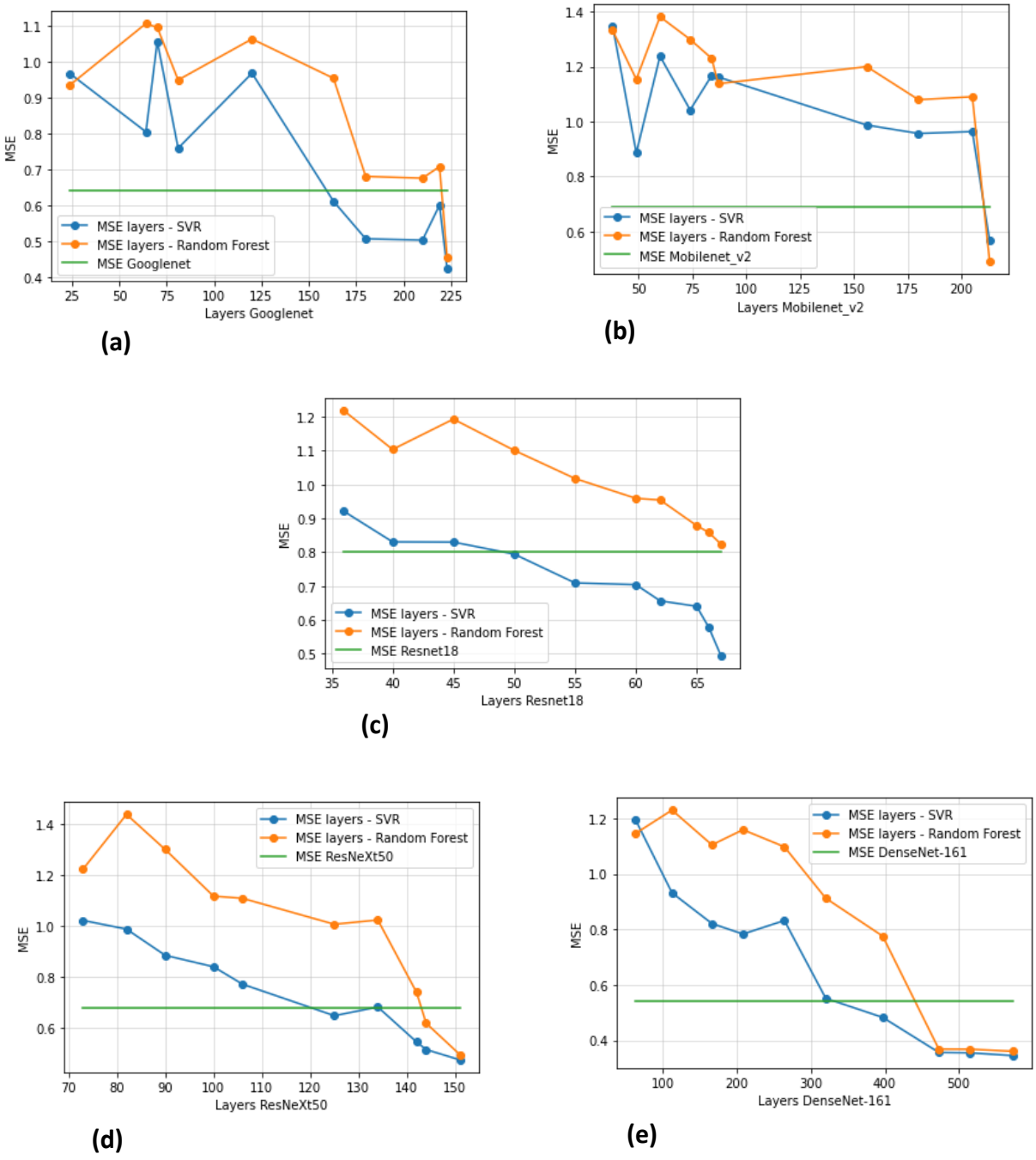

Figure 3.8: MSE (Mean Square Error) of pain estimation of each model ((**a**) GoogleNet, (**b**) MobileNEt, (**c**) ResNet18, (**d**) ResNeXt-50, (**e**) DenseNet-161) and their corresponding 10 layers when used as inputs to SVR and RFR.

Figure 3.8 shows the MSE obtained for each model evaluated after training and testing on the UNBC-McMaster shoulder database. The figure also shows the MSE of both SVR and RFR classifier for each of the 10 layers.

The results for Mobilenet_v2 show that the performance of the model when using feature extraction with the Support Vector Regression (SVR) and Random Forest Regression (RFR) classifiers is not consistently better than the performance of the stand-alone model. When analyzing the results of the feature extraction, it was noticed that there were some oscillations in the first layers, with two minimums being reached when using both SVR and RFR, however, these minimums were not always observed in the same layers. These minimums were not significantly better than the performance of the stand-alone model. However, after layer 100, the performance of the model when using feature extraction with SVR and RFR classifiers becomes smoother and converges quickly to the minimum on the last layer. This later achieves results that are better than the stand-alone model for both SVR and RFR. Therefore, when using Mobilenet_v2, the use of feature extraction is more relevant when using the features of the last layer in conjunction with SVR or RFR classifiers. This indicates that the last layers of Mobilenet_v2 contain more informative features for the task of pain estimation, and the use of these features with a different classifier can give better results.

The results for Resnet18 show a smooth behavior, with the performance of the model when using feature extraction with Support Vector Regression (SVR) and Random Forest Regression (RFR) classifiers converging to a minimum value that is reached in the last layer. When using Random Forest Regression (RFR) classifier, the results of feature extraction do not provide better results than the stand-alone model. This can be seen by the fact that the value achieved by the last layer is almost the same as the stand-alone model. On the other hand, when using Support Vector Regression (SVR) classifier, the results of feature extraction are better than the stand-alone model, especially after layer 50. This indicates that the later layers of Resnet18 contain more informative features for the task of pain estimation, and the use of these features with SVR classifier can give better results than using the stand-alone model. In conclusion, it is interesting to extract features from layers of Resnet18, especially the last one, and use SVR classifier, rather than using only the stand-alone model. This suggests that the later layers of Resnet18 contain more informative features that can be used to improve the performance of the model for the task of pain estimation when using SVR classifier.

The results for Googlenet are more intricate, with different minimums being observed throughout the layers, including the last ones. This indicates that the performance of the model when using feature extraction with Support Vector Regression (SVR) and Random Forest Regression (RFR) classifiers is not consistently better than the performance of the stand-alone model. When analyzing the results of the feature

Method	MSE
Siamese network [120]	1.03
Joint deep neural network [11]	0.95
HybNet [57]	0.76
LSTM [102]	0.74
CNN [119]	0.69
SCN [118]	0.32
EJH-CVV-BiLSTM * [12]	0.20
ResNet18-SVR *	0.49
ResNeXt-50-SVR *	0.47
MobileNet_v2-SVR *	0.46
GoogleNet-SVR *	0.42
DenseNet-161-SVR *	0.34

Table 3.6: Comparative analysis using state-of-the-art methods on the UNBC-McMaster database. The indication (star *) precises data balancing is used.

extraction with SVR classifier, it was noticed that the performance of the model becomes better than the stand-alone model after layer 160. This suggests that the later layers of Googlenet contain more informative features that can be used to improve the performance of the model for the task of pain estimation when using SVR classifier. On the other hand, when using Random Forest Regression (RFR) classifier, the results of feature extraction do not provide better results than the stand-alone model, except for the last layer. Googlenet's results are complex, with layer-specific performance. After layer 160, the SVR classifier enhances model performance, but the Random Forest Regression (RFR) classifier only improves the last layer. This shows that the latter layers of Googlenet contain more useful features that can be utilized to enhance the model's pain estimation performance when using SVR classifier and the last layer when using RFR classifier.

In the last layer for ResNeXt-50, the model's performance using feature extraction with Support Vector Regression (SVR) and Random Forest Regression (RFR) classifiers converges to a minimum value. When assessing the results of feature extraction using the SVR classifier, it was discovered that, beyond layer 134, the performance of the model surpasses that of the standalone model. This indicates that the subsequent layers of ResNeXt-50 have more useful features that may be used to enhance the performance of the model when estimating pain using the SVR classifier. Similarly, when utilizing the Random Forest Regression (RFR) classifier, the results of feature extraction are superior to those of the standalone model, particularly beyond layer 144. This suggests that the latter layers of ResNeXt-50 include more informative features for the task of

pain assessment, and that the combination of these features with the RFR classifier can produce better results than the standalone model. In conclusion, the ResNeXt-50 findings demonstrate that the addition of SVR and RFR classifiers improves the results of the final layer compared to the results of the standalone model. This means that the later layers of ResNeXt-50 have more useful features that can be used to improve the model's performance when estimating pain with SVR and RFR classifiers.

The performance results for DenseNet-161 demonstrate a steady evolution, with the model's feature extraction performance converging to a minimum at the final layers. The model demonstrates a linear behavior beginning at layer 470, with RFR and SVR obtaining comparable performance. Prior to layer 470, SVR typically outperforms RFR. In general, starting at layer 400, the feature extraction mode offers better results than the standalone model. This shows that later layers of DenseNet-161 include more informative characteristics that can be utilized to improve pain estimation performance. Both RFR and SVR are efficient classifiers in this situation, but SVR tends to produce somewhat better results than RFR before layer 470.

As shown in Table 3.8, the best performance in terms of Mean Squared Error (MSE) was achieved by utilizing features from the final layers of the CNN models as inputs to the Support Vector Regression (SVR) classifier. In addition, when examining the final layers of each model, the results indicate that SVR outperforms Random Forest Regression (RFR) classifier. DenseNet-161 was the model with the best performance among those we employed, and it was also the model with the deepest network. This observation is interesting, as it suggests that deeper networks may have an advantage in this task. It is also important to note that the performance of the studied models appears to increase with the total number of layers, which may indicate the significance of having a more complex architecture for this task.

For comprehensive analysis, we have also compared the results of our models against the results of the state of the art on the UNBC-McMaster database. The comparative results are given in Table 3.6. The results show that best performances are indeed obtained when extracting features from the last layer of existing pre-trained architectures. The models using features from the last layers and SVR classifier seem to perform better than some previous works of Bargshady et al. [11], Rodriguez et al. [102], and Tavakolian et al. [120, 119].

Although our obtained results are significantly better than many state-of-the-art methods, our results are still not optimal. In fact, Tavakolian et al. [118] reported a very low MSE of 0.32 using a Spatio-temporal Convolutional Network. Moreover, Bargshady et al. [12] achieved an MSE of 0.20.

In addition to the Mean Square Error (MSE) metric, we also used the Confidence Prediction Error (CPE) to evaluate the results of our experiments. CPE is a measure of the accuracy of a model's predictions, specifically in the context of regression problems. It is calculated as the absolute difference between the predicted value and the true value, expressed as a percentage of the true value. CPE is a useful metric because it allows us to evaluate the accuracy of a model's predictions in relation to the true values. A low CPE value indicates a high degree of accuracy for the model's predictions, while a high CPE value indicates a low degree of accuracy. In our experiments, CPE was calculated for each of the five CNN models (MobileNet_v2, GoogleNet, ResNeXt-50, ResNet18, and DenseNet-161) in both stand-alone mode and feature extractor mode (Figure 3.9). The results were used to compare the performance of the different models and to determine which layers of the CNNs were most informative for the task of pain estimation. The results obtained from CPE helped us to understand the performance of the models and to identify the best model for the task.

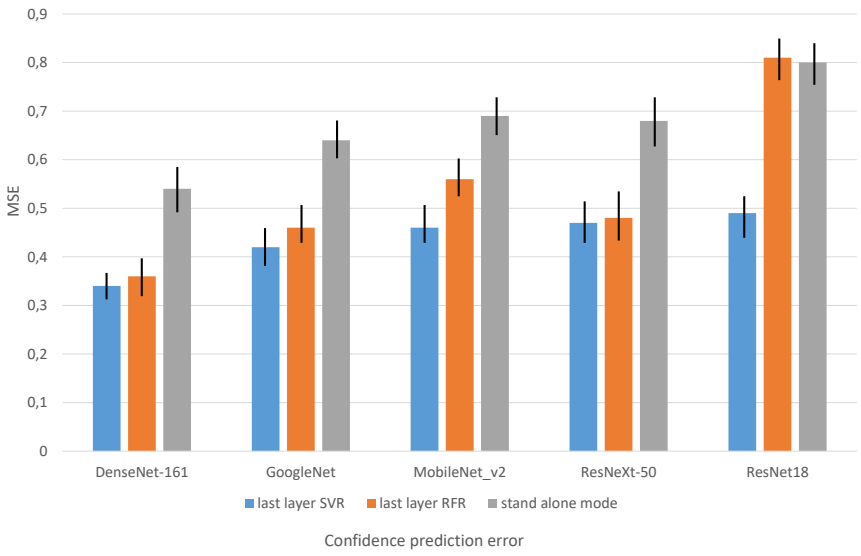

Figure 3.9: Confidence prediction error (CPE) used to evaluate the performance of the standalone mode, and feature extraction mode from the last layer using SVR and RFR for each of the five models.

In the Figure 3.9, the results of the CPE computation for each model are displayed. In

general, the feature extraction mode combined with SVR or RFR results in a lower CPE value for all models than the standalone option. This suggests that feature extraction can increase the predictability of a model.

CPE values for DenseNet-161 are 0.044 for feature extraction and SVR, 0.045 for feature extraction and RFR, and 0.047 for stand-alone operation. CPE values for GoogleNet are 0.046 for feature extraction and SVR, 0.047 for feature extraction with RFR, and 0.045 for standalone operation. CPE values for MobileNet v2 are 0.047 for feature extraction and SVR, 0.046 for feature extraction with RFR, and 0.043 for stand-alone operation. The CPE values for ResNeXt-50 are 0.047 for feature extraction and SVR, 0.047 for feature extraction and RFR, and 0.044 for stand-alone operation. The CPE values for ResNet18 are 0.047 for feature extraction and SVR, 0.037 for feature extraction with RFR, and 0.037 for standalone mode.

With a CPE value of 0.037, it is obvious that the feature extraction mode in combination with RFR provides the best results for ResNet18. This shows that the combination of feature extraction and RFR can improve the accuracy and confidence of predictions for this particular model. In contrast, the CPE values for the other models are comparable when the feature extraction mode is paired with either SVR or RFR; consequently, the choice between the two approaches would depend on other considerations, such as computing cost and implementation complexity.

In conclusion, the CPE calculation findings indicate that the feature extraction mode can increase confidence in the model's predictions. The combination of feature extraction and RFR appears to provide the highest performance for ResNet18, although the selection between SVR and RFR would depend on other factors.

3.4.3.b Computational Analysis

When contrasting the efficiency of various models, it is essential to take into account the amount of computation that is required by each model. ResNet18 has the most parameters out of all the models, as can be seen in Table 3.7; however, it also has the shortest training time. This makes it the most desirable model. This is probably due to the fact that its architecture is not as complicated as the architecture of the other models. DenseNet-161 on the other hand, which achieved the best results, had the highest number of parameters (30 million) and required more than 6 days for training. This demonstrates how deep learning models have a trade-off between accuracy and the amount of computation they require. Both MobileNet v2 and GoogleNet have a suitable amount of time for training and a reasonable number of parameters, making them appropriate for use in applications that have restricted access to computational resources.

The results were calculated based on a total of 100 images taken from the UNBC-McMaster Shoulder dataset. The average test time was determined using these results. According to the findings, each of the models was able to calculate an estimate of the level of pain in a relatively short period of time, with the average time taking anywhere from a few milliseconds to a few seconds for each image. These findings illustrate the potential of these deep learning models for estimating pain intensity in clinical applications in real time.

It is important to note that the results were obtained using an NVIDIA Quadro RTX 5000 GPU, and that the training and testing were carried out using Leave-one-subject-out cross-validation on the balanced UNBC-McMaster Shoulder dataset. Additionally, it is important to note that the results were obtained using the Leave-one-subject-out cross-validation method. The Adam optimizer was used throughout the training process, which lasted for a total of 200 iterations. When comparing the computational costs of various models, it is essential to take into consideration the mentioned factors because the results may be impacted by them.

Model	Total Parameters (Million)	Required Time for Train (Hours)	Average Time for Test (Second)
MobileNet_v2	3.4	43.33	0.59
GoogleNet	7	44.16	0.68
ResNet18	11	40.83	0.35
ResNeXt-50	25	90.55	0.83
DenseNet-161	30	157.22	2.84

Table 3.7: Comparative analysis regarding the computation costs (number of parameters, training time, and average test time) of different models.

3.4.3.c Case Study

The performance of the five models in feature extraction and SVR mode is shown in Figure 3.10 at the frame level for a single video. The figure displays the ground truth of the frames of the video and the predicted pain values for each model. From the figure, we can observe that all the models follow the shape of the ground truth, but the curve of DenseNet-161 is the closest to the ground truth among the models. On the other hand, ResNet18 has the farthest curve from the ground truth, indicating that its performance is not as good as the other models.

An interesting observation from the figure is that when the pain level is equal to zero, the curves of all the models overlap, suggesting that the models have a similar performance when there is no pain. However, when the pain level is at its maximum, the curves are not overlapping, indicating that the models have different behaviors in this case.

The results from this use case are consistent with the previous findings that showed DenseNet-161 as the best model in terms of performance. This highlights the importance of considering the models' behavior at different levels of pain intensity, as they may perform differently.

In conclusion, the figure provides useful insights into the performance of the models in feature extraction and SVR mode and confirms the previous finding that DenseNet-161 is the best model among the considered models.

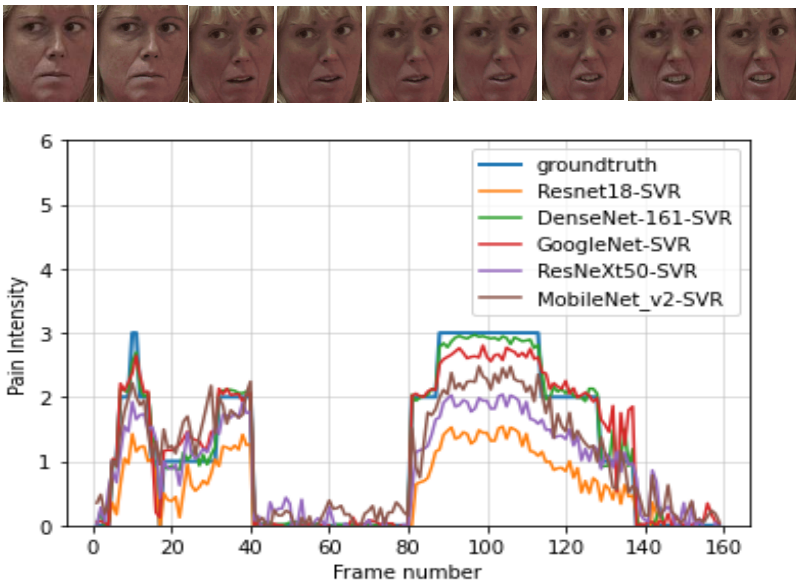

Figure 3.10: An example of continuous pain intensity estimation using all the considered CNN architectures on a sample video from the UNBC-McMaster database.

3.5 Conclusion

In this work, we conducted a comprehensive analysis on pain estimation by comparing some popular and Off-the-Shell CNN architectures, including MobileNet [109], GoogleNet [115], ResNeXt-50 [149], ResNet18 [49], and DenseNet-161 [56]. We used these networks in two distinct modes: stand-alone mode or feature extractor mode. Features were extracted from 10 different layers. Predictions were done with SVR (Support Vector Regression) and RFR (Random Forest Regression) classifiers. The results are given in terms of Mean Square Error. We conducted an evaluation with balanced data from UNBC-McMaster Shoulder Pain Database [81]. The database contains 10783 images and consists of 4 pain levels (no pain, weak pain, mid-pain, and strong pain). The obtained results indicated the importance of feature extraction from the last layers of pre-trained architectures to estimate pain. Most of the used architectures achieved significantly better results compared to many state-of-the-art methods.

Chapter 4

Vision Transformers for Pain Detection and Discrimination between Genuine and Posed Pain

4.1 Introduction

Pain has many different expressions, depending on the individual. Moreover, it is difficult to estimate pain from facial expressions, as there are many factors that can affect an individual's ability to express pain. Being able to easily detect pain would be a great asset for doctors and other medical professionals. However, in this chapter, we first propose a Transformers-based architecture that demonstrated the ability to detect genuine and posed pain from facial expressions; second, we propose another Transformers-based method to deal with the pain and no pain issue. We demonstrated the effectiveness of transformers in two areas: the classification of pain and no pain, and the differentiation of genuine and posed pain from facial expressions. Transformers have an advantage over other architectures because they can learn global dependencies. This is important for tasks like pain classification, where the global context is important to understanding whether someone is in pain or not. The transformer can learn this sort of global context and make better predictions as a result.

4.2 Related work

4.2.1 Genuine versus Posed Pain

Facial expressions have a considerable impact on human social interactions. In recent research, facial expressions have been used to understand social interactions [59]. However, since facial expressions are a mirror of emotions [143], they might not reflect our true feelings. Therefore, distinguishing genuine from posed emotions is a relevant study. In the case of our research, we are aiming to detect genuine pain from a posed one. Moreover, this presents an important task in some medical and criminal applications [17].

In the early study of Hill *et al.* [53], the authors highlighted the difference in facial actions in terms of frequency, type, and intensity between genuine and posed expressions. In fact, posed pain expressions showed different temporal patterns than real pain expressions. The authors of this study worked on a dataset that contains 40 low back patients' facial expressions. These patients were videotaped raising their legs in order

to capture real pain and then pretending to be hurt in order to capture fake pain. Later work by Bartlett *et al.* [13] used features of 20 Action Units (AU) to train a non-linear SVM for the classification of spontaneous and posed pain. The method was evaluated on a self-collected database.

Another study by Littlewort *et al.* [71] distinguished between posed and genuine pain using Support Vector Machines (SVM). The SVM was trained on a database composed of 26 adults videotaped under three experimental conditions: baseline, genuine pain, and posed pain. Genuine pain conditions consist of cold pressure pain. The findings of this study were compared to naive human discrimination of genuine from posed questions. The obtained results of the method outperform those of humans. Tavakolian *et al.* [117], published the most recent work on the task of distinguishing genuine from posed pain. In this paper, the authors propose a Residual Generative Adversarial Network (R-GAN) method to distinguish real from fake pain expressions by magnifying the subtle changes in faces. This method captures and encodes the appearance and dynamics of a specific video into an image map, using Weighted Spatio-temporal Pooling (WSP). To evaluate their approach, the authors used three databases. First, the UNBC-MCMaster [81] contains videos of genuine pain. Second, the BioVid Heat Pain Database [143] contains videos of both genuine and posed pain. Finally, the STOIC database [105] which contains only posed pain expressions.

4.2.2 Pain recognition

There is a considerable amount of literature on automatic pain recognition from facial expressions. First, studies that focus on the detection of the presence of pain (pain or no pain). Other approaches work on the estimation of pain level. These studies propose handcrafted methods, machine learning, or deep learning architectures. Chen et al. [27] proposed a novel architecture for detecting and locating joint pain event in video. The authors extracted features by applying a histogram of oriented gradients (HOG) and using them as an input to a support vector machine (SVM). They used the UNBC McMaster Shoulder Pain [81] dataset.

In their work, Laduona Dai et al [31] suggested a technique for pain identification using facial expressions. The process entails extracting action units (AUs) using OpenFace 2.0 software. These AUs are used to represent facial expressions that indicate pain, such as wrinkles or elevated eyebrows. The authors then train a support vector machine (SVM) model using the intensity values of the collected AUs. The purpose of the model is to make a prediction of a binary output based on the AUs. Five-fold cross-validation was used to assess the model's performance during training and testing on data from

the UNBC-McMaster Shoulder Pain Expression Archive. The findings indicated that the model's accuracy was 85%. However, the study also revealed a shortcoming of the method, since all facially-moving data were identified as pain movements. This limitation is mostly attributable to the fact that all no-pain samples show motionless, neutral faces, making it more difficult to distinguish between pain and no-pain samples.

Bargshady et al [11] proposed a hybrid method by joining a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN). They first used VGGFace to extract deep features from images of the UNBC McMaster Shoulder Pain [81] dataset. In this paper, the authors aim to classify pain into four categories: no pain, weak pain, mild pain; and strong pain. One of the state-of-the-art approaches that uses deep learning with deep features is the work of Haque et al [48]. The authors used CNN to extract deep features, which they then fed into a long short-term memory network [55] (LSTM). They evaluated early and late fusion strategies for the recognition of pain levels. This study was trained using the Multimodal Intensity Pain [55] (MIntPAIN). This database consists of 20 adults with stimulated electrical pain. In a recent work by Karamitsos et al [63], the authors proposed a novel Convolutional Neural Network (CNN) for automatic pain detection from facial expressions. The proposed CNN consists of a modified version of VGG16 [113] model. They conducted experiments using the UNBC McMaster Shoulder Pain dataset [81].

4.2.3 Transformer models

Transformers are a type of machine learning model designed for processing sequences. They were mentioned for the first time in the paper [129]. Based on self-attention mechanisms, transformers have gained fairly widespread use. The original paper focused on natural language processing [129]. This is where transformers have been most commonly used. But they can be applied to other types of data as well, such as images. Moreover, transformers in computer vision have proved promising results in different fields, for instance, image recognition, object detection, and segmentation [65]. Some of these transformers achieve and outperform state-of-the-art results by relying only on self-attention and without the use of convolutional neural networks (CNNs). In fact, Transformer allows parallelization for sequential data, contrary to CNNs.

Several authors have attempted to use Transformer architectures for vision tasks. Some methods used transfer learning from the vision transformer model (ViT) [33] for zero-shot anti-spoofing issues [39]. In this work, the authors applied fine-tuning of a pre-trained vision transformer and achieved state-of-the-art performance. Another

work studied the classification of Covid-19 from chest images based on vision transformer models [38]. In this work, the ViT model [33] was used and compared to CNN models. It has been shown with evaluation results that ViT performs better than CNNs. An additional method using Transformer models for deepfake detection [51]. In this study, the authors provided a vision Transformer model with distillation methodology [124]. The use of Transformer models gave a robust model for deepfake detection that outperformed state-of-the-art results.

4.3 Detection of Genuine versus Posed Pain from Facial Expressions using Vision Transformers

An important issue in computer vision and facial expression understanding, is the ability to distinguish spontaneous expressions from fake ones. The high resemblance between the two states (Genuine and Posed expressions) makes this issue a challenging yet crucial field of research [94]. As we can see in the late work presented in the paper [69]. Posed facial expressions are distinguished by their intensity, duration, and configuration [59]. Therefore, the existing work on differentiating genuine facial expressions from posed ones can be classified into four categories. First, consider muscle movement (Action Units (AU)). For instance, a recent work [100] used the AlexNet model on 12 AUs intensities to obtain the features. Second, spatial patterns are based, as in the work of Van Der Geld *et al.* [128] for smile expression, and they analyze tooth display, position, and smile width from a dental perspective. Third, texture-feature-based methods. Here, we can cite the study of Tavakolian *et al.* [117] and Littlewort *et al.* [71] on pain expression. Finally, hybrid methods that combine different classes of features [108] [70].

In this study, we propose a Vision Transformer model to capture the subtle changes in facial expressions. We use the fine-tuning of the pre-trained Vision Transformer model [33]. Since the ViT determines the relationship between patches, we propose that those patches be represented by the frames captured from a single video of the database. In this case, ViT will compute the relationship among the frames of a video. Therefore, the represented changes in facial movements will be detected. Fig. 4.2 illustrates the proposed architecture. We start by taking as input an image that consists of concatenated frames. This input image is split into non overlapping patches (in our case, each patch is a frame from the video). The patches are then flattened into a vector form and fed to the ViT as a sequence.

To the best of our knowledge, this is the first study to use Vision Transformer (ViT) to differentiate between genuine and posed pain. The main contributions of this study are the following:

- Present a Vision-based Transformer architecture for detection of Genuine versus Posed pain.
- Prove the efficiency of fine-tuning Vision Transformers using a small database, contrary to the use of ViT from scratch on the same database.
- Prove the importance of sequential order in time for the discrimination between Genuine and Posed pain.

4.3.1 Proposed Vision Transformer for genuine and posed pain differentiation

In this work, in order to detect genuine and posed pain from facial expressions, we propose transfer learning from a pre-trained Vision Transformer. Our study provides a framework based on the architecture of the Vision Transformer (ViT) [33].

At first, Transformers were proposed for the Natural Language Processing (NLP) task by Vaswani *et al.*[129]. These models broke multiple NLP records and pushed the state of the art. These models are attention-based encoder-decoder types. However, CNNs [36] have been incredibly successful and have achieved important results in image classification. In the paper by Dosovitskiy *et al.* [33], they proved that the reliance on CNNs is not necessary anymore, and that the use of a pure Transformer can perform very well, and even outperform CNNs results. A Vision Transformer requires partitioning an input image into patches of the same shape. These patches might overlap or not, depending on the user's subject. Therefore, every patch is a small color image with RGB (Red, Green, and Blue) channels. Thus, a patch is an order of three tensors. The next step is to vectorize the patches. Which means reshaping these tensors into vectors. After that, a dense layer is applied to those vectors, and then the positional encoding vector is added. Then those vectors are fed to the Transformer layers [33]. After this latter, a Multi-Layered Perceptron MLP was added for classification, and it consists of a fully connected layer.

The Vision Transformer models are trained on large datasets, and this helped boost the performance of ViT so that it outperformed the state-of-the-art CNNs models. In-

deed, training a ViT from scratch is very computational and needs a large dataset. Nonetheless, fine-tuning the ViT does not require expensive computational resources and also takes advantage of the powerful ViT since it has been trained on large datasets. In this paper, we use a pre-trained ViT which we adapt to our use case, to study the transferability of the ViT for Genuine versus posed pain issues.

4.3.1.a The concept of attention

The concept of attention in artificial neural networks refers to the ability of a model to selectively focus on certain parts of the input data while processing it. Attention mechanisms allow models to weigh and combine different parts of the input data dynamically, based on their relevance or importance to the task at hand.

There are several different types of attention mechanisms that have been developed for use in neural networks. One of the most well-known is the self-attention mechanism, which operates by projecting the input data into a higher-dimensional space, where the data points are represented as vectors. The model then calculates the pairwise dot-products between these vectors, which represent the similarity between each pair of data points. These dot-products can be calculated using the following equation:

$$dot - product = x_i^T x_j \tag{4.1}$$

Where x_i and x_j are the vectors representing the i-th and j-th data points, respectively.

These dot-products are then normalized using a softmax function, which converts them into a set of weights that represent the importance of each data point with respect to the others. The softmax function is defined as follows:

$$softmax(x_i) = \frac{exp(x_i)}{sum(exp(x_j)} \tag{4.2}$$

Where x_i is the i-th dot-product and x_j is the $j - th$ dot-product.

These weights are then used to linearly combine the input data points, producing a weighted sum that is used as the output of the self-attention layer. The weighted sum can be calculated using the following equation:

$$output = sum(w_i \times x_i) \tag{4.3}$$

Where w_i is the weight for the i-th data point and x_i is the i-th data point.

Attention mechanisms have been shown to be effective at capturing long-range dependencies and relationships in data, and are an important component of many modern neural network architectures. They have been widely used in natural language processing tasks and are also beginning to be applied to computer vision tasks as well.

In the field of computer vision, attention mechanisms have been used to allow models to focus on certain parts of the input data (e.g., images) that are relevant or important for the task at hand. This can be particularly useful for tasks that involve input data with complex structures or long-range dependencies, such as object detection or image segmentation.

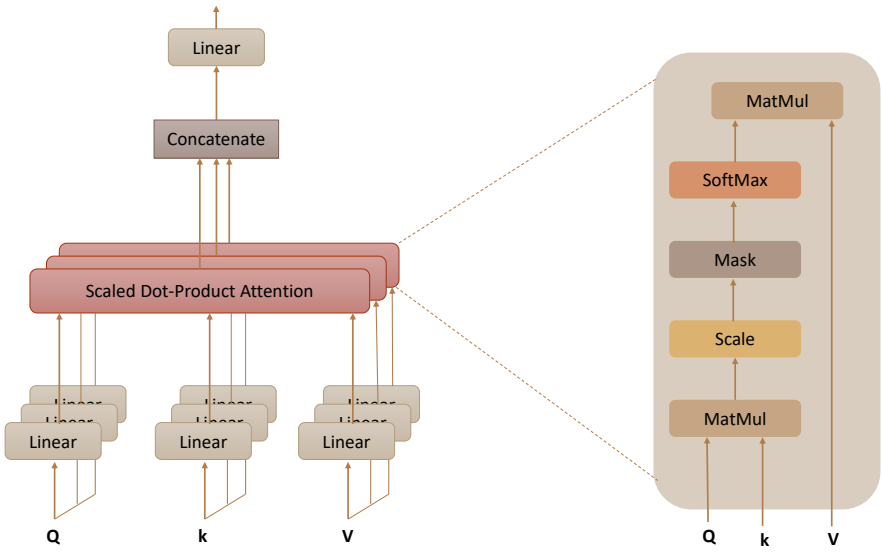

Figure 4.1: Multi-Head self Attention and self attention head.

4.3.1.b The transformer attention mechanism

There are several different types of self-attention mechanisms that have been used in transformers and other neural network architectures.

One type of self-attention is called dot-product attention, which operates by calculating the dot-products between the input data points, as described above. This type of attention is relatively simple to implement, but can be limited in its ability to capture more complex relationships in the data.

Another type of self-attention is called multi-head attention. It involves dividing the input data into multiple "heads" and performing self-attention independently on each head. The self-attention mechanism calculates the dot-products between the input data points, normalizes them using the softmax function, and produces a weighted sum of the input data points as described in the previous messages.

To perform multi-head attention, the input data is first divided into H heads, where H is the number of heads. The self-attention mechanism is then applied independently to each head, producing H weighted sums of the input data points. These H weighted sums are then concatenated and transformed using a linear layer, producing a single combined output. The combined output of the multi-head attention can be calculated using the following equation:

$$output = linear\_transform(concat(head_1, head_2, ..., head_H)) \tag{4.4}$$

Where head_i is the weighted sum for the i-th head and linear_transform is a linear transformation (e.g., a fully-connected layer) that transforms the concatenated heads into the final output.

Multi-head attention has several advantages over single-head attention. First, it allows the model to attend to multiple parts of the input data at the same time, which can be useful for tasks that require a global understanding of the input data. Second, it allows the model to learn multiple different attention patterns, which can be useful for tasks with complex relationships in the input data. Finally, multi-head attention can improve the expressiveness of the model, as it allows the model to learn more complex combinations of the input data.

Multi-head attention is a key component of the transformer architecture, which has been highly successful in natural language processing tasks. It has also been applied to computer vision tasks, where it is used in the vision transformer architecture.

4.3.1.c Proposed model

In this work, we proposed an architecture based on the ViT model. The new architecture for the differentiation between Genuine and Posed Pain is called LinViT. The first step in this method is the input image pre-processing stage, as can be seen in Fig 4.2. The process is called image to patches. Since a Transformer can only process a sequence of tensors given an image, we split it into patches (in Fig 4.2 we use nine patches to give an example). After the input image has been converted to patches, those patches are

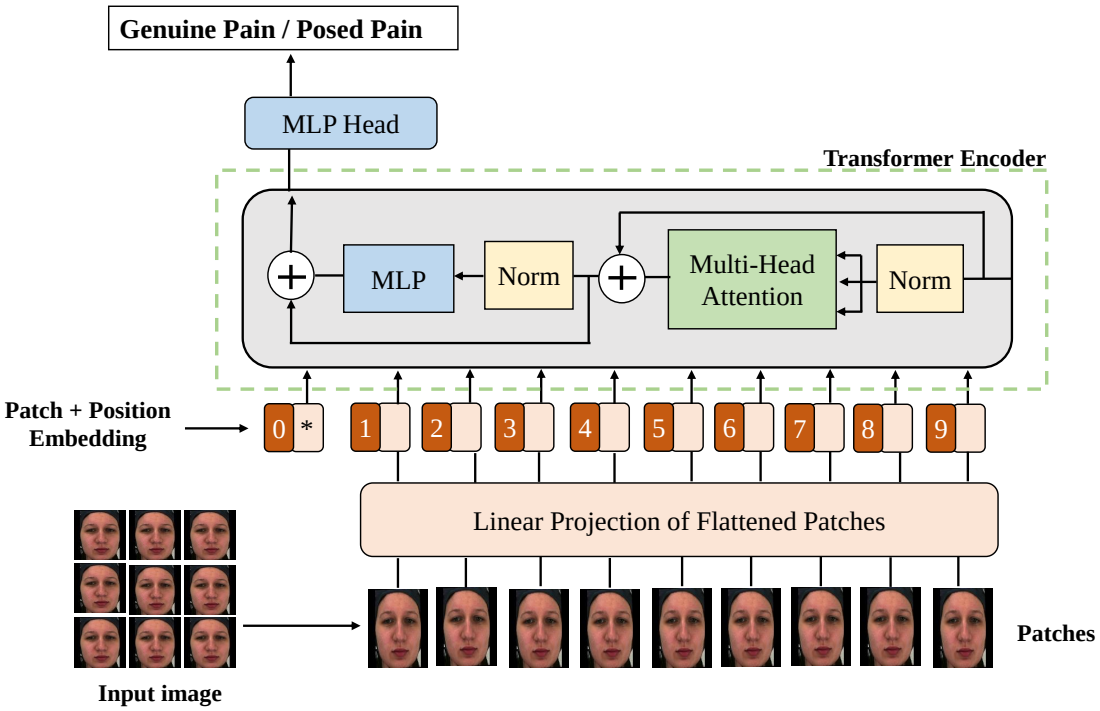

Figure 4.2: The architecture of Vision Transformer ViT for the detection of Genuine versus Posed Pain.

converted to vectors using a linear projection process. It happens at the input stage of the transformer. All patches are arranged in sequence from top left to bottom right. All patches go through a linear layer to produce vectors. The next step is adding position embedding. The vision transformer introduces a learnable class embedding or token(*). This token is also assigned by a position embedding(0).

One of the main components of the architecture is the Transformer Encoder. The first step of this encoder is the Normalization layer (Norm). It is used to normalize the output of the previous layers, which helps to stabilize the training process and improve the overall performance of the model. Then, there is the Multi-Head Self-Attention. This element allows the model to attend to different parts of the input image, which helps it to better understand the overall structure of the image. The Multi-Head Self-Attention used in our method is shown in Fig 4.3. In this figure, we see a traditional self-attention mechanism with the addition of Projection blocs. The main of these projections is to have a lower dimension, so the inner products will not be expensive. This linear attention reduces the complexity from $O(n^2)$ to $O(n)$ [137]. The resultant linear transformer matches the performance of conventional Transformer models while consuming signif-

icantly less memory and time. This linearity is adapted from the Linformer paper [137].

The skip connections are used in the Encoder. This element helps to make the model more robust to changes in the input image, by allowing the output of the layer to be added to the input of the same layer. This allows the model to learn the residual changes between the input and the output. In addition to the components mentioned earlier, the encoder in our architecture also includes a Multi-Layer Perceptron (MLP). The MLP is a type of neural network that consists of multiple fully connected layers. The role of the MLP is to learn a non-linear function that maps the output of the multi-head self-attention layer to a new representation that is more suitable for the task at hand. In other words, the MLP component is responsible for learning more complex interactions and representations from the self-attention layer. The outputs of the MLP component are then passed to the feed-forward network for further processing.

In this work, we train the proposed model LinViT using binary cross-entropy loss BCE (4.5). Where N is the number of images being predicted, p_i is the probability that the model will predict Genuine Pain. Concerning $1 - p_i$, it represents the probability of class 0, if the model predicts Posed Pain. The ground truth is represented by y_i. It's 1 if the image presents Genuine Pain and 0 for Posed Pain. Finally, the model is trained with 10×10 patches to match our input images, which will be detailed in the following section.

$$Logloss = -\frac{1}{N}\sum(y_i \times log(p_i) + (1 - y_i) \times log(1 - p_i)) \tag{4.5}$$

4.3.2 Experiments

Throughout this section, we detail the used database, also we will highlight the parameter setting and configuration needed for training, then we will present the baseline methods.

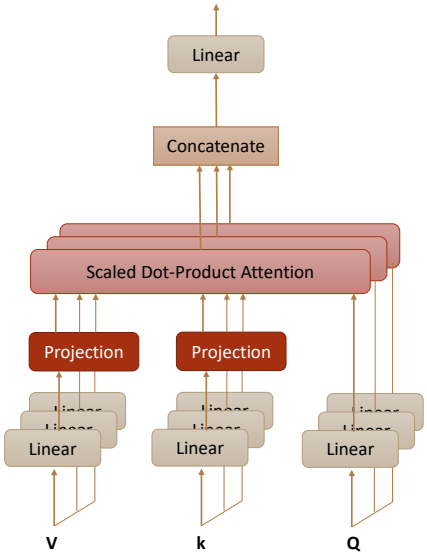

Figure 4.3: The addition of the Projection blocks that allow the reduction of Keys and Queries dimension[137].

4.3.2.a Database and pre-processing

The aim of our study is to differentiate between genuine and posed pain. To the best of our knowledge, the only database that contains both states (genuine and posed pain) is the publicly available BioVid Heat Pain Database [143]. Therefore, to evaluate the performance of our architecture, the experiments are executed on this database. Moreover, the BioVid Heat Pain Database [143] database was collected to help advance works on pain assessment. It concerns 90 participants that were subject to 4 intensities of induced heat pain. The database consists of four parts. In our study, we will be using the two parts: A with genuine pain and D with posed pain. Concerning part A, the useful information in our case is the frontal video. It contains 87 subjects, 5 classes (no pain and the four intensities of pain), and each subject has 20 samples per class, which makes it 8,700 samples. Every sample lasts 5.5 seconds. Concerning part D, the information we'll be interested in is also the frontal videos of the subjects. It contains 1 minute posed pain videos of 90 subjects.

As seen before, the BioVid Heat Pain Database [143] contains frontal videos. Since we will be using ViT, we convert the videos to frames. To focus on the face changes, and avoid any disturbance from the background, we used the Multitask Cascaded Convolutional Networks [148] (MTCNN) as a face detector. Then, we align the detected face and crop it to a resolution of 100×100. For part A of the database that contains 5.5 seconds videos, we capture the frames of each video and get rid of the first and last 8 frames because it doesn't contain expression of pain. Therefore, for each video, we get

a total of 100 frames. Concerning part D with posed pain, we do the same thing and capture frames from each video and get rid of the first and last frames, to get in the end a total of 1500 frames per video.

Pre-processing:

In the context of using the LinViT architecture for image processing, the sequence aspect of the architecture is leveraged by concatenating multiple frames of a video together into one big image. This allows the model to detect the temporal relationship between the frames, as the frames are processed together in a sequential order.

For example, in the described implementation of Part A, the frames are captured at a resolution of 100×100 pixels each. These frames are then concatenated together to create one big image of size 1000×1000 pixels. This image contains all the frames of a single video in a sequential order.

In Part D, each video is represented by 15 images containing captured frames. This means that the video is divided into 15 different segments, and each of these segments is represented by an image of captured frames in the sequential order. These big images have the same resolution as before:1000×1000 pixels, and of course, it contains 100 frames of 100×100 pixels each.

This approach of concatenating the frames of a video together allows the LinViT to better understand the temporal relationship between the frames and make more accurate predictions. It also allows the model to learn temporal patterns in the video that could be missed if the frames were processed independently. In Fig. 4.4, we illustrate the different steps before the used preprocessing.

4.3.2.b Implementation details

To train the LinViT, a specific image size and patch size were used. The image size is 1000×1000 pixels and the patch size is 100×100 pixels. As mentioned earlier, each patch corresponds to a single frame of a video, allowing the model to learn the temporal relationship between frames. The training and testing processes were conducted on an NVIDIA Quadro RTX 5000 GPU machine with 32GB of memory. The LinViT model was designed to classify two classes, so it has two output neurons.

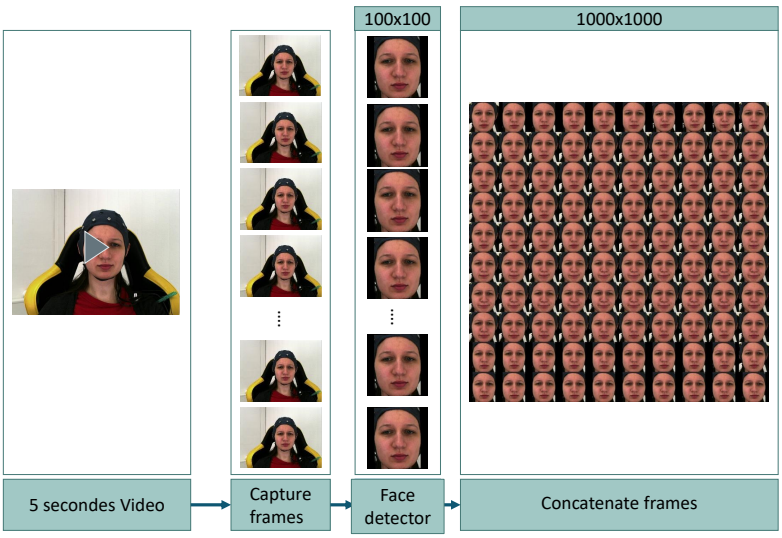

Figure 4.4: Prepossessing of the BioVid Heat Pain Database.

The batch size used during the training process is 64, meaning that the model is updated with 64 samples at a time. The optimizer used is the standard Adam Optimizer [66] with an initial learning rate of 0.01. The learning rate is then reduced over time using a step-based method. The training process was done for 30 epochs, which means that the model was trained on the same dataset for 30 times. After the training process, the best model is selected based on the minimum loss during the validation phase.

For classification, we used the binary cross-entropy loss BCE to supervise our model. The LinViT architecture was implemented in the Pytorch library [95]. We tested on the public available BioVid Heat Pain Database [143] and the accuracy of the LinViT model was measured and compared with state-of-the-art methods. This comparison was done to evaluate the performance of the LinViT model against existing methods in the field and to see if it can achieve similar or better results.

4.3.2.c Baseline methods

To compare our method with state-of-the-art methods, we implemented two baseline methods : Recurrent neural network (RNN) and Long short-term memory [55] (LSTM). Since our method deals with sequential data, we have chosen two methods that include

time in the aspect too.

RNN to detect Genuine versus Posed Pain (RNN-GPP): On account of the fact that temporal dynamic behavior is necessary in our study, RNNs are a potential choice for this task. RNNs are a class of neural networks that allow previous outputs to be used as inputs while having hidden states. In other words, the computation takes into account historical information. RNNs are an architecture that captures relationships between the inputs within several time steps by learning to conserve relevant information.

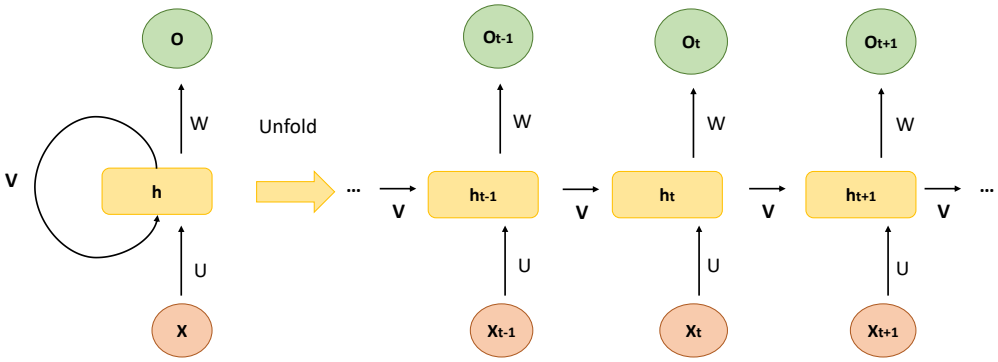

Figure 4.5: General Recurrent Neural Networks architecture.

RNNs are presented as a class of neural networks that operate well with sequential data. In actuality, this kind of network analyzes the input sequence one element at a time and keeps track of a hidden state vector that acts as a repository for previous data. Recurrent neural networks acquire theability to carefully preserve pertinent data in order to detect dependencies over a range of time steps.

The key components associated with recurrent neural networks include memory, a non-linear activation function, and a non-linear transfer function. The memory stores the hidden state of the network, and the hidden state is modified by applying the non-linear transfer function. Hidden states are modified sequentially by applying an activation function to the hidden state at each time period to form a new hidden state. In addition, weights associated with synaptic connections are adjusted according to a learning rule. Therefore, recurrent neural networks are able to learn to process sequential data that is dependent on the current and past states.

Figure 4.5 outputs the fully recurrent neural networks (FRNN) architecture that connects all neurons' outputs and inputs. This is the most general neural network topology, since all other topologies may be replicated by setting some connection weights to zero to imitate the lack of connections between particular neurons. As can be seen in the Figure 4.5, the hidden neurons i the RNN are linked over time by a feedback connection. The element of the current sequence, x_t, and the hidden state, h_{t-1}, which was recovered from the previous time step, are given to it as inputs at time t. The output of the network, O_t, is then computed after updating the hidden state to h_t. U is the weight matrix that, like in a traditional neural network, connects the input and the hidden layers. V presents the weight matrix for the recurrent transition, which connects one hidden state to the following. W displays the weight matrix for the transition from hidden to output.

LSTM to detect Genuine versus Posed Pain (LSTM-GPP): The LSTM [55] models are a variant architecture of RNNs. They were designed to mitigate the vanishing and exploding gradient problem. Thus, they can learn short and long term dependencies. LSTM networks have an internal mechanism called gates that can regulate the flow of information. These gates can learn which data in a sequence is relevant, then decide to keep it or to throw it away. Following this mechanism, the LSTM network learns to use important information to make predictions.

The LSTM architecture is divided into three parts: forget gate, the second part is known as the input gate and the last one is the output gate. As shown in the Figure 4.6, an LSTM has a hidden state, just like a straightforward RNN, with h_{t-1} standing for the hidden state of the prior timestamp and h_t for the hidden state of the present timestamp. Additionally, LSTMs have a cell state that is denoted by the timestamps C_{t-1} and C_t, which stand for the prior and current timestamps, respectively. Below, more details about the three gates in LSTMs' architecture.

- **Forget Gate:** The initial step in an LSTM network cell is to choose whether to keep or discard the data from the preceding timestamp. The Forget gate equation is given below by the equation 4.6.

$$f_t = \sigma(X_t \times U_f + h_{t-1} \times W_f) \tag{4.6}$$

  Where:

  X_t: the timestamp's current input.

  h_{t-1}: the previous timestamp's hidden state.

  U_f: weight associated with the input.

W_f: the hidden state-related weight matrix.

A sigmoid function is then applied to it. This will result in f_t being a number between 0 and 1. This f_t is then multiplied by the previous timestamp's cell state, as shown in Figure 4.6, in the Forget gate. The network will forget everything if ft is equal to 0, but nothing if ft is set to 1.

- **Input Gate:** The value of the new information carried by the input is measured by the input gate. The input gate's equation is shown below.

$$i_t = \sigma(X_t \times U_i + h_{t-1} \times W_i) \tag{4.7}$$

Where:

X_t: the timestamp's current input.

h_{t-1}: the previous timestamp's hidden state.

U_i: input weight matrix.

W_i: Weight matrix of the input associated with the hidden state.

The new information can be expressed as follows :

$$N_t = \tanh\left(X_t \times U_c + H_{t-1} \times W_c\right) \tag{4.8}$$

The new information required to be passed to the cell state is now a function of a hidden state at timestamp t-1 and input x at timestamp t. Tanh is the activation function in this case. The tanh function causes the value of new information to range between -1 and 1. If N_t is negative, information is subtracted from the cell state; if N_t is positive, information is added to the cell state at the current timestamp. However, the N_t will not be directly added to the cell state. Here is the updated equation.

$$C_t = f_t \times Ct - 1 + i_t \times N_t \tag{4.9}$$

$Ct - 1$ represents the cell state at the current timestamp, while the other variables are previously determined values.

- **Output Gate:** depending on what is needed, it can allow the cell to forget or remember its former condition. Here is the equation of the Output gate below, which is similar to the previous two gates.

$$o_t = \sigma(X_t \times U_o + h_{t-1} \times W_o) \tag{4.10}$$

Because of the sigmoid function, its value will also be between 0 and 1. We will now use o_t and tanh of the updated cell state to determine the current hidden state. As illustrated below.

$$H_t = o_t \times \tanh(C_t) \tag{4.11}$$

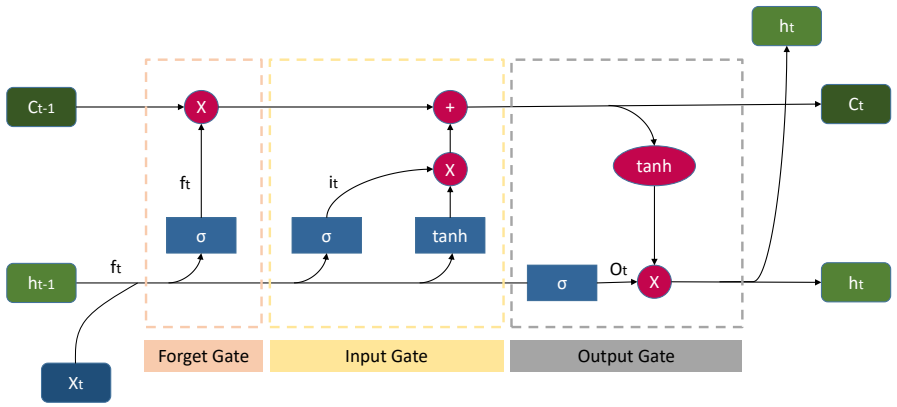

Figure 4.6: Long Short Term Memory architecture.

For both RNN and LSTM for Genuine versus Posed Pain methods, we indeed applied the pre-processing detailed in the second paragraph of the subsection 4.4.1. Therefore, the captured frames from the BioVid Heat Pain Database [143], are fed to the model respecting the sequential order in the video. Also, the input size is 100×100, and the last layer is adapted for the classification of two classes.

4.3.3 Performance analysis

In this section, we will analyze the performance of the LinViT model. We will start by comparing the results of the LinViT model with the results of the baseline methods. This comparison will provide an understanding of how well the LinViT model performs compared to the existing methods in the field. Next, we will study the impact of using an unordered database on the performance of the LinViT model. This analysis will provide an understanding of how the order of the frames in the video affects the performance of the model.Finally, we will conduct a computational analysis to evaluate the efficiency

and scalability of the LinViT model. This analysis will provide an understanding of the computational resources required to train and run the model and how it can be optimized for real-world applications.

Overall, the objective of this section is to evaluate the performance of the LinViT model in terms of accuracy, robustness, and computational efficiency. This will provide a comprehensive understanding of the capabilities and limitations of the LinViT model, and help to identify areas for improvement.

4.3.3.a Comparison with baseline methods

The experiments for evaluating the performance of the proposed LinViT method were conducted on the BioVid Heat Pain Database [143]. This database contains videos of individuals experiencing both genuine and posed pain, making it an ideal dataset for evaluating the ability of the LinViT model to distinguish between the two types of pain. The performance of the LinViT model was evaluated using accuracy as the primary metric and compared to the results of several baseline methods. These baseline methods were selected to represent the current state-of-the-art in the field of pain detection and provide a benchmark for the LinViT model.

In order to ensure a fair comparison, we carefully justified the choice of pre-processing method applied to the database. This pre-processing step is important to ensure that the database is prepared in a way that is suitable for the LinViT model and that the results are comparable to the baseline methods.

To evaluate the impact of the temporal relationship on the Vision Transformer's performance and its ability to discriminate between Genuine and Posed pain, we performed experiments using disordered sequences. In these experiments, the order of the frames in the video was randomly shuffled to disrupt the temporal relationship between the frames. This allows us to understand how the temporal relationship between the frames affects the performance of the LinViT model and the discrimination of the two pain types.

In this work, we propose a novel approach for representing videos of the BioVid Heat Pain Database [143] by means of an image. This image is composed of the concatenation of captured frames from the video, arranged in the same order as the original video. This approach enables the detection of temporal relationships between frames,

Table 4.1: Comparison of the performance of our proposed model for the differentiation between Genuine and Posed Pain while varying the size of the input image.

Pre-processing detail	Frame size	Input size	Accuracy (%)
One video is captured into three big images and each image is a concatenation of 36 frames	100×100	600×600	81.07
One video in captured into one big image and each image is a concatenation of 81 frames	80×80	720×720	84.78
One video in captured into one big image and each image is a concatenation of 100 frames	100×100	1000×1000	85.13

which is crucial for the task of distinguishing between genuine and posed pain.

We conducted a series of experiments to evaluate the performance of our proposed LinViT model on this task. Three different configurations of the LinViT model were tested, each with a different number and size of frames concatenated into the input image.

The first experiment involved capturing 36 frames from each video and concatenating them together to form three images per video. These images were then preprocessed using the MTCNN [148] to detect faces, resulting in frames of 100×100 pixels. The final images were of size 600×600 pixels, and the predicted value for each video was obtained by averaging the three outputs from the model. This configuration resulted in an accuracy of 81.07

The second experiment involved capturing fewer frames, specifically 81 frames, and reducing their size to 80×80 pixels. These frames were concatenated together to form a single image of size 720×720 pixels. This configuration achieved an accuracy of 84.78

Finally, the third experiment involved capturing the maximum number of frames possible, 100 frames, and using a frame size of 100×100 pixels. The resulting image was of size 1000×1000 pixels. This configuration achieved an accuracy of 85.13

It is worth noting that the size of the input image, and the number and size of the frames concatenated, can affect the performance of the model, and therefore it is important to adapt the proposed model to the size of the input image. Furthermore, the

model architecture was adapted to the size of the input image, by using a 6×6, 9×9, and 10×10 patches for the first, second and third experiment respectively.

Table 4.2: Results of the baseline methods, ViT from scratch and the proposed model on discrimination of Genuine from Posed Pain. The performance is evaluated on the BioVid Heat Pain Database [143] by measuring the accuracy.

Method	Accuracy (%)	Duration of training
RNN-GPP	71.96	30 Hours
LSTM-GPP	75.22	40 Hours
ViT-GPP	85.63	14 Hours
LinViT (proposed method)	85.71	5 Hours

As previously mentioned, our experiments indicate that representing a video by a single image, rather than multiple images, results in better performance for the LinViT model. This can be attributed to the preservation of the temporal relationship between frames when the video is not divided. When the video is divided, the sequence of frames is interrupted, resulting in the loss of some information that is crucial for the task of pain detection. The slight difference in accuracy between the experiments using 81 and 100 frames suggests that the majority of relevant information in the video is concentrated in the middle of the video. Therefore, in light of these findings, we have chosen to conduct further experiments using a 100×100 model with an input image size of 1000×1000. This allows us to capture the maximum number of frames while preserving the temporal relationship between frames, thus providing the best performance for our model.

In this study, we have compared the performance of the proposed LinViT model with two baseline methods, RNN-GPP and LSTM-GPP, for the task of discriminating between Genuine and Posed Pain. These methods were evaluated on the publicly available BioVid Heat Pain Database [143]. The results, as shown in Table 4.2, indicate that the proposed LinViT model outperforms the two baseline methods. The accuracy of the LinViT model reaches 85.71%, which is significantly better than the results obtained by RNN-GPP and LSTM-GPP, which are 71.96% and 75.22% respectively. These results suggest that RNNs, although taking the sequential aspect into consideration, may not be able to access information from a long time ago. On the other hand, the LSTM-GPP method showed an improvement in accuracy, which confirms the ability of LSTM to retain memory for longer sequences.

We have also trained the ViT for our classification. The ViT architecture was adapted to our case and been trained on the same dataset as the other model. The ViT for Gen-

uine versus Posed Pain (ViT-GPP) has achieved 85.63% which is near to the accuracy of our proposed model. This result emphasis the fact that the linearity used in our transformer doesn't impact performance. The superiority of LinViT over the two baseline methods highlights the validity of our proposed architecture for the differentiation between Genuine and Posed Pain. The LinViT architecture allows the model to learn temporal patterns in the video and make more accurate predictions. Furthermore, the ability to work on the entire video at once, and not just a sequence of frames.

Based on the information presented in Table 4.2, it appears that the LinViT model has the least amount of training time at 5 hours, while the LSTM-GPP model has the longest training time at 40 hours. The RNN-GPP model has a training time of 30 hours, and the ViT model has a training time of 14 hours. It is worth noting that the use of linear attention in the LinViT model affects the time consumption during training, which may account for its relatively short training time compared to the other models. It is worth noting that the training time and accuracy of a model are not always directly proportional, and other factors such as the complexity of the model and the availability of computational resources can also affect the training time. However, the LinViT model stands out for its balance between accuracy and time consumption, it reaches a high accuracy level while consuming less time than the other models.

4.3.3.b Impact of unordered database

In order to further understand the influence of the temporal relationship on Vision Transformers, and to demonstrate the ability of these models to capture long-range dependencies, we conducted experiments using disordered frames. Specifically, we configured a function that concatenated frames in a random order and trained the LinViT model on this dataset. The results showed that the model achieved an accuracy of 59.14% when using disordered frames. This finding highlights the importance of the temporal relationship for LinViT, as it demonstrates that the model is heavily dependent on the order of the frames in the video. As seen in Fig. 4.7, the difference in accuracy between the disordered and ordered datasets is considerable, emphasizing the importance of the temporal relationship in the LinViT model.

In contrast, when the RNN-GPP and LSTM-GPP models were trained using random images, the difference in performance between the ordered and disordered datasets was relatively small. This can be explained by the fact that RNNs and LSTMs also take into

account the spatial aspect of images and extract information from them. These results indicate that RNNs and LSTMs are less affected by the temporal relationship between frames than Vision Transformers, but also that Vision Transformers are particularly suited for sequential data.

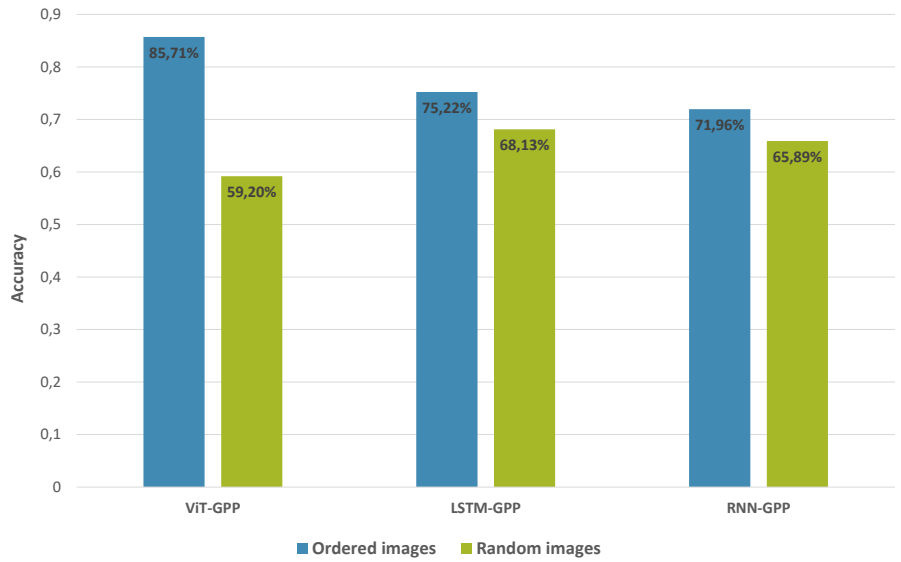

Figure 4.7: Comparison between performances of the proposed method LinViT while concatenating ordered images and random ones. The same comparison is also done for the state-of-the-art methods. The models take as input images in their sequential order and then randomly.

4.4 Pain Detection From Facial Expressions Based on Transformers and Distillation

Facial expressions are important in social interactions. They express spontaneously the emotions of certain people. Facial expressions, therefore, provide information that can be analyzed nowadays not only by humans but also by machines. We can highlight

the importance of introducing machines to emotion detection by the fact that, in some cases, humans are incapable of analyzing facial expressions (for instance, if a person is paralyzed or in the case of infants). One of the important applications of computer vision using facial expressions is pain assessment.

Pain presents a complex phenomenon that is not completely understood, starting with its definition as an unpleasant feeling that may be a consequence of numerous causes (for instance, medical causes, emotional or psychological ones[146]). Pain actually generates spontaneous facial expressions. Therefore, in most of the research in pain recognition, the researchers use images of facial expressions [141]. In addition, most of the publicly available databases of pain contain facial images of videos of patients [81] [143] [6].

Regarding the importance of automatic detection of pain from facial expressions, many researchers focus their studies on the detection of pain or no pain task. Others limited their research to the estimation of pain level or chronic versus non-chronic pain. Different methodologies have been used. Beginning with handcrafted methods and progressing through machine learning methods to deep learning approaches [141]. In our paper, we introduce a novel method for the automatic detection of pain. We propose a transfer learning method for pain detection from facial expressions that makes use of pre-trained data-efficient image transformers [124] (Deit). This method is based on the transformers [129] that were designed first for Natural Language Processing (NLP). The pioneering work in [33] demonstrated the effectiveness of these transformers for image recognition. We have chosen the Deit [124], as it incorporates distillation that exploits CNNs. To train our proposed architecture, we considered two databases, namely the UNBC McMaster Shoulder Pain [81] and the BioVid Heat Pain [143].

The contribution of this research is to provide an effective pain assessment method based on facial expressions. This report outlines the following contributions:

- Present a fine-tuned data-efficient image transformers (Deit) for pain and no pain detection.
- Highlight the importance of transformers in the image's recognition field in general, and in pain tasks more particularly.
- Prove the efficiency of transformers comparing to Convolutional Neural Networks (CNN) while studying the discrimination of pain from no pain task.

4.4.1 Databases and Proposed Method

In this section, we present the two databases used in our experiments. The first database is UNBC McMaster Shoulder Pain and the second is BioVid Heat Pain. Both databases have been widely used in previous research and have been proven to be challenging for the pain detection task. In order to prepare the databases for training, pre-processing steps were applied and are detailed below. Our proposed architecture is trained end-to-end, allowing the model to learn both from the rich feature representations of the ResNet50 encoder and the high efficiency of the transformer encoder. This architecture is elaborated in the second part of this section.

4.4.1.a Databases and Pre-Processing

The experiments of this study are done on the two Databases: UNBC McMaster Shoulder Pain Database [81] and BioVid Heat Pain Database [143]. Those two Databases are publicly available. In Fig. 4.8, we present some sequence examples from both databases.

UNBC McMaster Shoulder Pain Database: It consists of 25 adults with shoulder pain. This database includes four parts: first 200 video sequences containing spontaneous facial expressions; Second $48,398$ Facial Action Coding System (FACS) coded frames; Third, associated pain frame-by-frame scores and sequence-level self-report and observer measures; and finally 66-point Active Appearance Model(AAM) landmarks. In our study, we are interested in part two that consists of $48,398$ images. These images are capturing facial expressions, while pain intensity changes. In our case, we are working on a binary representation of pain. Therefore, our database is divided into two classes: pain and no pain.

BioVid Heat Pain Database: It is a multimodal database. It contains frontal videos, biomedical signals: Galvanic Skin Response (GSR), Electrocardiography (ECG), and Electromyography (EMG) at trapezius muscle. Pain in this database was stimulated by induced heat pain in four intensities. For each intensity, 20 experiments are done. In our research, we will be interested in frontal videos. In addition, this database is divided into four parts. We will be using part A during our experiments. This part contains 87 subjects with 5 classes (no pain and 4 pain intensities). We convert videos to frames. Thus, this database presents a total of 797343 images.

In order to focus on facial expressions, it is in our interest to crop the face of the subjects. First, we use the Multitask Cascaded Convolutional Networks [148] (MTCNN) as a face detector. Second, once the face is detected, we align it. Finally, we crop it to

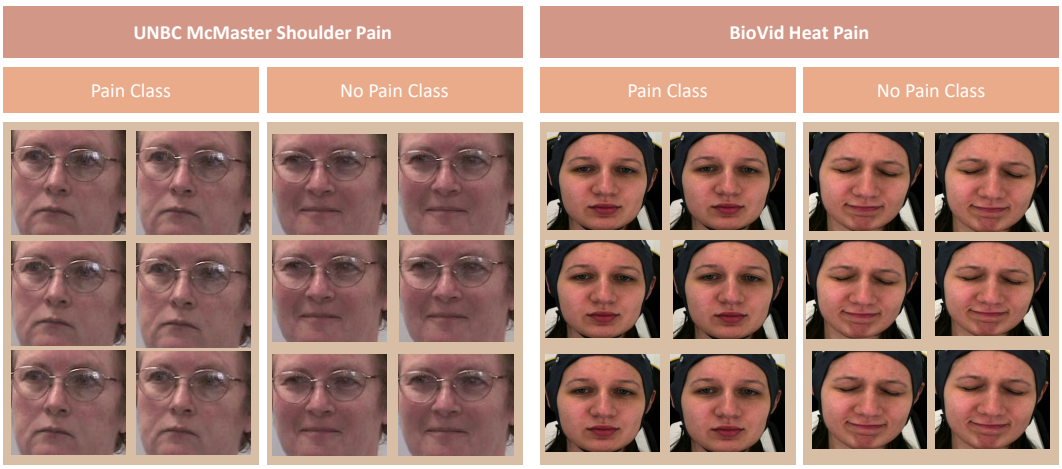

Figure 4.8: Examples of some sequences from the UNBC-McMaster shoulder pain [81] and from the BioVid Heat Pain Database [143] databases. These sequences show the difference of facial expressions for patients having pain and no pain.

Classes	**UNBC McMaster Shoulder Pain Database**			**BioVid Heat Pain Database**		
	Train	***Validation***	***Test***	***Train***	***Validation***	***Test***
Pain	5 574	1 393	1 402	311 040	77 760	168 480
No Pain	22 344	5 585	12 100	134 688	33 672	71 703
Total	27 918	6 978	13 502	445 728	111 432	240 183

Table 4.3: Amount of images in the used Databases : UNBC McMaster Shoulder Pain Database [81] and BioVid Heat Pain Database [143]. The amount of images for each class for train validation and test.

an image of size 256×256. We divide each database into two classes : one for images that represent no pain, and the other one gathers all pain intensities to constitute one class for pain. Table 4.3 gives more details about the amount of images in every class and every database.

The UNBC McMaster shoulder pain and BioVid heat pain datasets are unbalanced (Table 4.3). This means that there is a disproportionate number of observations in one class compared to the other. This can be a problem when training our model, as it can lead to bias towards the class with more observations. To overcome this problem, we balanced the databases using data augmentation. Data augmentation is a technique that involves creating new, synthetic observations from the existing ones. This can be done by applying various transformations, such as rotation, scaling, and flipping, to the images in the dataset. By doing this, we were able to create additional observations for

the class with fewer observations, which helped to balance the dataset.

4.4.1.b Proposed Method

In our approach, we propose a novel framework based on the transformer. This latter has been introduced in the paper [129]. The transformers are deep learning networks that were conceived first for the Natural Language Processing (NLP) tasks. It represents a sequence-to-sequence architecture. Moreover, the transformers are based on a self-attention mechanism which has the ability to learn the relationship between sequences' components. Self-attention is one of the key ideas of the novelty presented by transformers, in addition to the pre-training on large datasets. Therefore, self-attention is a mechanism that estimates the relevance of an item over another[24]. The attention mechanism can be defined by the equation 4.5.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{4.12}$$

Let's consider an image feature maps $\mathbf{X}$, where $\mathbf{X} \in \mathbf{R}^{n \times d}$. Q is the matrix of the query (vector of one word in NLP tasks and patch in image recognition), K represents the keys (vector of all patches or words in a sequence). V is a vector of values, containing also all the patches or words of a sequence. Therefore, attention mechanisms did bring novelty and efficiency in networks for computer vision in general, and for image recognition in particular. In our case, our proposed architecture is based on a transformer that uses distillation knowledge from a Convolutional Neural Network (CNN) as a teacher in addition to attention mechanism. (More details about Transformers are mentioned in Section 4.3.1)

In this work, we propose a novel approach for pain recognition using transfer learning with the Data-efficient image transformer (DeiT) model. The DeiT model was introduced in [124] as an efficient method for training transformers for image recognition tasks using mid-size databases. The DeiT model achieved promising results by only using the ImageNet dataset [32] for training. The DeiT architecture is based on the concept of distillation [54], which is the process of transferring knowledge from one network to another. In the original DeiT paper, the authors used a pre-trained CNN on the ImageNet dataset as the teacher model and a modified version of the Vision Transformer (ViT) as the student model. The output of the CNN is passed as input to the transformer, which is used to extract useful representations from the input images, thus improving the efficiency of the transformer.

The DeiT architecture is composed of several layers of self-attention and Feed Forward Network (FFN), which are used to extract useful representations from the input images. The model also includes a distillation token, a class token, and patch tokens. The distillation token ensures that the student learns from the teacher through attention, the class token goes through all blocks for the original classification done by the transformer and the patch tokens are obtained from the input image. We will use a technique called hard distillation (as in the original model), where the temperature is set to one, which means that the label of the teacher model is taken as the true label. This distillation loss is then summed up with a cross-entropy loss of the transformer. This hard distillation method, along with the DeiT architecture's efficient use of patch tokens and class tokens, helps the model to accurately classify pain and no pain in images.

In our proposed method, we use the DeiT model as the base architecture and fine-tune it using the UNBC-McShoulder and BioVid databases for the task of pain recognition. The DeiT model, as previously discussed, is an efficient transformer-based architecture for image recognition tasks. By fine-tuning the DeiT model on these databases, we aim to have the model learn to differentiate between images depicting pain and no pain. As mentioned above, the main idea behind distillation is to train the student model to mimic the output of the teacher model on a given input. This can be achieved by minimizing the difference between the output of the student model and the output of the teacher model, using a distillation loss function. In our case, the teacher we will be using is ResNet50 which is pre-trained on the ImageNet dataset.

The distillation process in our study starts by training the teacher network (ResNet50) on the database, and the output of the fully connected layer (FC) is taken as the teacher output. Next, the student network, transformer-based model is trained on the same dataset and the output of the last layer of the transformer is taken as the student output. Then we calculate the distillation loss (the cross-entropy in our purpose), which is defined as the difference between the output of the student and the teacher networks (equation 4.13). The student network is trained by minimizing the distillation loss with respect to the student's parameters.

$$L = -(\frac{1}{N}) \times \sum (y_i \times \log(s_i) + (1 - y_i) \times \log(1 - s_i)) \tag{4.13}$$

where L is the loss, y_i is the teacher output, s_i is the student output, and N is the number of instances in the dataset.

In summary, our proposed method for pain recognition combines the efficiency of the DeiT model with the guidance of the ResNet50 [49] model, which leads to a more robust and accurate model. The fine-tuning process, combined with the use of the

ResNet50 [49] as a teacher, allows the model to learn to differentiate between pain and no pain images.

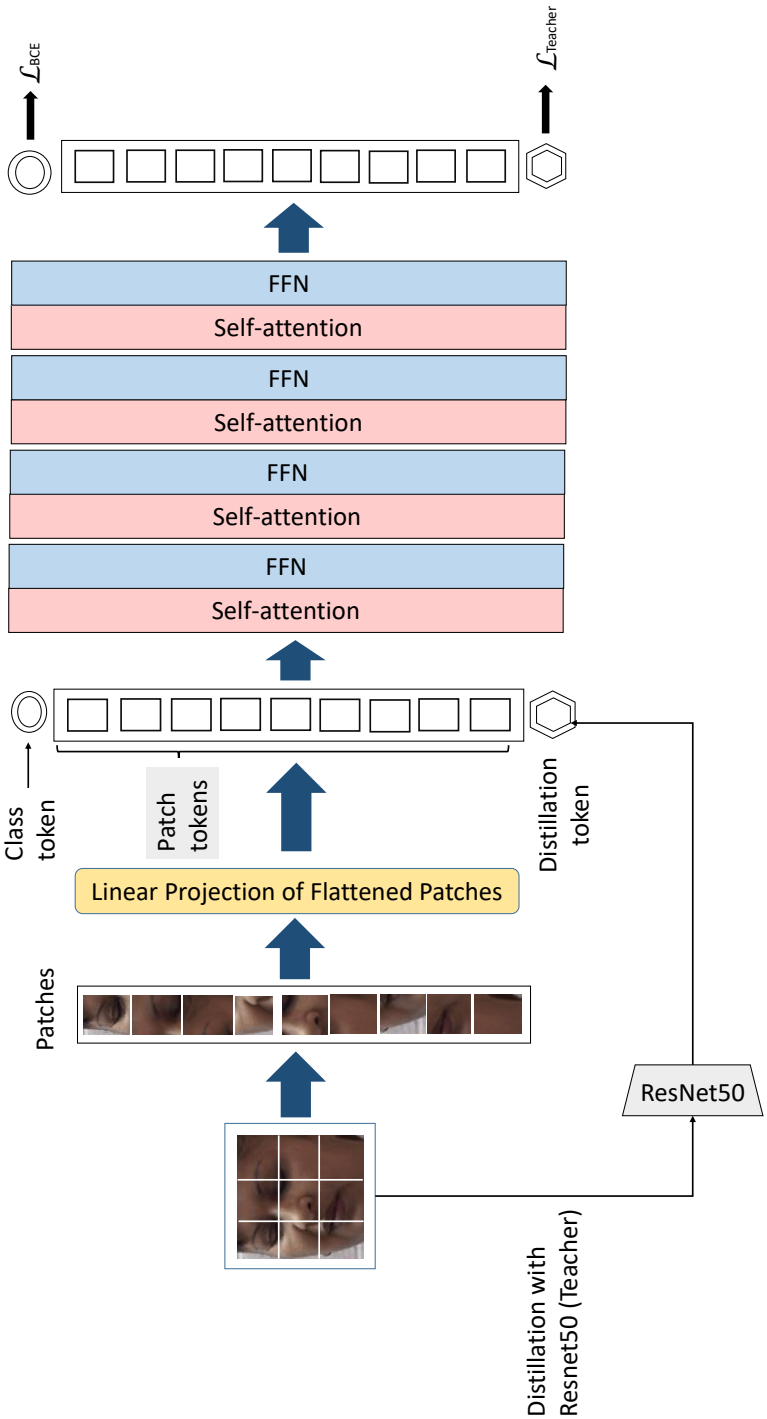

Figure 4.9: An overview of the proposed pipeline for Pain Recognition using Data-efficient image transformer (Deit) [124]. (FFN stands for Feed Forward Network. BCE is the Binary Cross Entropy.)

In order to compare state-of-the-art methods with our proposed model, we implemented various models such as GoogleNet [115], DenseNet-161 [56] and LSTM [55] for the detection of pain. To do that, we used the pre-trained version of GoogleNet on ImageNet, and adapted the last layer for two classification classes. GoogleNet is a model that gives interesting results while using it in pain detection and estimation [36]. Also, it is a model that does not take huge computational resources. Therefore, the fine-tuned GoogleNet will be trained on UNBC-McMaster shoulder pain [81] and the BioVid Heat Pain [143] datasets.

Similarly, DenseNet-161, which is a deep and dense convolutional neural network and LSTM, a Recurrent neural network, were trained on the same datasets. The DenseNet-161 model is known for its ability to handle large datasets, and LSTM model is known for its ability to handle sequential data, which makes them suitable for our task.

4.4.2 Experimental results

In this section, we present the experimental results of our proposed method for pain recognition. The section is divided into two main subsections: the first subsection, training details, provides information about the experimental setup. The second subsection, performance analysis, presents the results of our experimental evaluation. We will also compare our proposed method with other state-of-the-art methods and discuss the results in detail.

4.4.2.a Training details

As seen in the subsection 4.4.1, after the detection, alignment, and cropping of the images, we resize them to 256×256. We augment our training data using various data augmentation techniques such as random horizontal flipping, random rotation, and random cropping, to increase the diversity of our data. During the training, we fixed the patch size to 32 which is the size of the image used to extract the features. In addition, the learning rate is set to 0.00001 which is a small value that allows the model to converge gradually while avoiding overfitting. The model is trained for 30 epochs with a batch size of 64. To optimize the model during training we used the standard Adam Optimizer [66] which is a widely used optimizer for deep learning.

For classification, we select the best parameters using back-propagation with the

binary cross-entropy (BCE) loss. The experiments were done on a machine with two NVIDIA Quadro RTX 5000 GPUs and 32GB of memory. The training parameters were used to train both DeiT [124] and Resnet50 [49] separately on the two datasets: UNBC-McMaster shoulder pain [81] and the BioVid Heat Pain [143]. We also ran multiple experiments with different parameters such as patch size, batch size, and learning rate to make sure that the final parameters we used for the model were the optimal for the task of pain recognition.

4.4.2.b Performance analysis

We conducted experiments on both UNBC-McMaster shoulder pain [81] and the BioVid Heat Pain [143] datasets. To evaluate the proposed fine-tuned Deit model for the recognition of pain, we used the accuracy as a metric. First, we train the proposed method separately on the two datasets. As shown in Table 4.4, we obtained an accuracy of 84.15% while the Deit-PNP is trained on the UNBC-McMaster shoulder pain [81]. Surprisingly, the accuracy achieved when we used the BioVid Heat Pain [65] dataset is 72.11%. Although the BioVid Heat Pain [143] dataset contains more data, the UNBC-McMaster shoulder pain [81] achieved better results.

We also trained three state-of-the-art models: GoogleNet, DenseNet-161 and LSTM. GoogleNet achieved an accuracy of 80.01% when trained on the UNBC-McMaster shoulder pain dataset, and 65.75% when trained on the BioVid Heat Pain dataset. DenseNet-161 achieved an accuracy of 79.6% on UNBC McMaster dataset, and 65.46% on BioVid dataset. LSTM achieved an accuracy of 80.4% on UNBC McMaster dataset and 71.09% on BioVid dataset. We can notice that the performance of these models decreases when using the BioVid dataset. Overall, the results of these experiments demonstrate the potential of the proposed fine-tuned DeiT model for the recognition of pain in facial expressions. The proposed method achieved better results than the state-of-the-art models, which confirms the importance of using transformers in this task.

In comparison to the results obtained in the state of the art, the proposed fine-tuned DeiT model shows competitive performance. For example, on the BioVid Heat Pain dataset, the proposed method achieved an accuracy of 79.11%, which is higher than the 72.4% accuracy achieved by Werner et al [139]. Similarly, on the UNBC-McMaster shoulder pain dataset, the proposed method achieved an accuracy of 84.15%, which is higher than the 75.2% accuracy achieved by Bargshady et al [11] and the 73.04% accuracy

achieved by Yang et al [151]. Additionally, it is worth noting that on the BioVid dataset, the method proposed by Yang et al [151] achieved 60.23% accuracy. This is lower than the accuracy obtained by the proposed method.

However, it should be noted that some state-of-the-art methods such as the method proposed by Karamitsos et al [63] achieved higher accuracy than the proposed method, with 92.5% on the UNBC-McMaster dataset. It is also worth mentioning that the method proposed by Laduona Dai et al [31] achieved 85% accuracy on the UNBC-McMaster dataset using a different approach, the Action Units based method. It is important to note that we have not used the Leave One Subject Out Cross Validation (LOOSCV) since we have big datasets and the training takes long time to be done. This cross-validation method is commonly used in the literature in order to evaluate the performance of a model while taking into account the inter-subject variability.

In conclusion, the proposed fine-tuned DeiT model has demonstrated promising results in the detection of pain from facial expressions. The model achieved competitive performance compared to the state-of-the-art methods, with similar or better results on the UNBC-McMaster shoulder pain and BioVid Heat Pain datasets. This is an important step in the use of transformers for pain detection, as it is one of the first studies to explore this approach. However, it is important to note that this is an emerging field of research, and there is still much to be done in order to improve the performance of the model. Future work could include the use of other transformer architectures, as well as the exploration of other datasets to further evaluate the generalization and robustness of the proposed method. Overall, the proposed method has the potential to be a useful tool in the detection of pain from facial expressions, but further research is needed to fully realize this potential.

It is worthwhile noting that the two databases are not balanced. The first one: UNBC-McMaster shoulder pain [81], the amount of images belonging to no pain class is much bigger that the one belonging to pain class. This can be noticed in Table 4.3. Concerning the second database: BioVid Heat Pain [143] is also unbalanced. Contrary to the first one, this database contains images of pain more than the ones with no pain. The fact that the databases are not balanced is a potential cause of the difference obtained in accuracy. Despite this problem, we can still state that our proposed architecture of a fine-tuned Deit for pain recognition exceeds the state-of-the-art methods in terms of accuracy.

Table 4.4: Results of the different experiments of the proposed method to detect pain from no pain, compared to the state-of-the-art models. The experiments are done using the publicly available datasets: UNBC-McMaster shoulder pain [81] and the BioVid Heat Pain [143]. We note the proposed architecture Deit-PNP to design the fine-tuned Deit for detection of Pain from No Pain. The same for the models LSTM, DenseNet-161 and GoogleNet.

Method	**Accuracy %**	
	UNBC McMaster Shoulder Pain dataset	**BioVid Heat Pain Database**
Werner et al [139]	-	72.4
Laduona Dai et al [31]	85	-
Bargshady et al [11]	75.2	-
Karamitsos et al [63]	92.5	-
Yang et al [151]	73.04	60.23
GoogleNet-PNP	80.01	65.75
DenseNet-161-PNP	79.6	65.46
LSTM-PNP	80.4	71.09
Deit-PNP (Proposed methos)	84.15	79.11

4.5 Conclusion

In this chapter, we have proved the efficiency of Transformers in two topics : classification of Pain and No Pain, and discrimination between Genuine and Posed Pain from facial expressions. This chapter was divided into two parts, each one details one topic. In the first part, presented a novel architecture for the binary recognition of pain from facial expressions. This architecture is based on the data-efficient image transformers [124] (Deit). We used fine-tuning of the pre-trained Deit model on the ImageNet [32] dataset. The Deit model achieved interesting results compared to the state of the art. We trained the proposed method using UNBC-McMaster shoulder pain [81] and BioVid Heat Pain [143] datasets. Moreover, to compare our proposed architecture with the state of the art, we implemented a pre-trained model to discriminate pain from no pain facial expressions. We chose the GoogleNet [115] model. At the end of the experiments, our proposed method showed promising results compared to the state-of-the-art method.

The Second part concerns the differentiation between Genuine and Posed Pain from facial expressions. We used a fine-tuning of a pre-trained Vision Transformer model. Our architecture was evaluated using the publicly available BioVid Heat Pain Database [143], and achieved promising results. In fact, our proposed method outperforms the state-of-the-art methods. This study proved the importance of the sequential aspect to detect Genuine from Posed Pain. We also demonstrate the fact that Vision Transformers require large databases to give better results. Therefore, Vision Transformers are a promising method to adopt in studies of pain in general.

These works are the first step towards future investigations of other image recognition Transformers. Considering that Transformers are capable of capturing long-range dependencies, and CNNs have the ability to detect important static information in an image, our further studies will focus on the combination of Transformers with CNNs, in order to improve our results.

Chapter 5

Conclusion

5.1 Conclusions and contributions

This thesis provided an overview of automated pain evaluation utilizing facial expressions and presented a thorough examination of automatic pain intensity assessment. Then, utilizing transformers, we introduce our novel method for detecting genuine versus posed pain. This chapter summarizes the findings and contributions of our proposed approaches. In addition, we explore obstacles to automatic pain evaluation and provide challenges for future research.

A reliable and objective assessment of pain is necessary for differential diagnosis, choosing the proper therapy, monitoring progress, and evaluating whether it is necessary to continue or modify a treatment. In addition to causing suffering and diminishing quality of life, uncontrolled pain impairs the neurological system, endocrine system, and immunological system [89]. Consequently, pain evaluation and treatment are crucial not only for providing relief but also for preventing both immediate and long-term effects that are detrimental to the individual's overall health [89]. Inappropriate pain management may result in chronic pain syndrome, which is often accompanied by reduced mobility, weakened immunity, lower concentration, anorexia, and trouble sleeping. Inappropriate treatment may also result in extra complications and worries for patients.

In spite of advances in knowledge and technology, the management of pain in many cases is still inadequate [83]. Although this is a prevalent issue, patients whose communication skills are restricted, who are unable to describe their level of pain or whose reports have a poor level of authenticity, are the ones who are most adversely impacted by it. These vulnerable groups include newborns, toddlers, and children, as well as people with cognitive impairments (like severe dementia), intellectual disabilities, people who are very sick or unconscious, and people with terminal illnesses [52].

In the past decade, automated pain identification has progressed from an idea to a topic of intense research. Automatic pain identification systems based on pain behaviors (such as facial expressions, vocalizations, and physical movements) and physiological reactions are a supplement to the standard evaluation methods that are currently employed to improve pain management. These tools allow continuous monitoring of pain, as opposed to standard evaluation methods. This could lead to better clinical outcomes, like making it easier for people who can't get help on their own to take part in early intervention. Moreover, automated systems are more objective than human observers, whose assessments may be influenced by subjective factors such as the observer's emo-

tional connection to the patient or the patient's physical attractiveness [141].

This thesis presents our works to in the context of the pain assessment from facial expressions. These works aim to improve the automatic pain recognition research. In this thesis, we have focus on the facial expressions of pain.

Firstly, we present the background of the topic and our motivations for conducting the study. In addition, the contributions of the thesis are outlined, and the papers are briefly summarized.

The third chapter compares some popular and off-the-shelf CNN (Convolutional Neural Network) architectures for automatic pain recognition from facial expressions, including MobileNet, GoogleNet, ResNeXt-50, ResNet18, and DenseNet-161. We use these networks in two distinct modes: stand-alone mode or feature extractor mode. In stand-alone mode, the models (i.e., the networks) are used for directly estimating the pain. In feature extractor mode, the "values" of the middle layers are extracted and used as inputs to classifiers, such as SVR (Support Vector Regression) and RFR (Random Forest Regression). We performed extensive experiments on the benchmark and publicly available database called UNBC-McMaster Shoulder Pain.

The succeeding chapter consists of two Transformers-based methodologies used to address two issues: pain detection and the distinction between genuine and posed pain. Considering that pain typically induces spontaneous facial expressions, these facial expressions could be utilized to detect the existence of pain. In this section, we suggest the fine-tuning of data-efficient image transformers and distillation (Deit) for facial expression pain detection. The suggested architecture's effectiveness is assessed using two publicly available databases, UNBC McMaster Shoulder Pain and BioVid Heat Pain.

Concerning the use of facial expressions to distinguish between Genuine and Posed Pain, we describe a novel method based on Vision Transformer (ViT). To differentiate between Genuine and Posed Pain, the model must instead focus on the subtle changes in facial expressions that occur over time. The employed method takes the sequential aspect into account and detects the variations in facial expressions. Experiments utilizing the publicly accessible BioVid Heat Pain Database indicate the efficacy of our technique.

5.2 Perspectives

This inquiry has a long way to go, and several issues need to be resolved. Despite the progress made as a consequence of our contributions, there are still a number of challenges to overcome in order to develop effective and applicable facial expression analysis approaches for inferring pain. If automated pain evaluation is ever to be used, these concerns must be studied in greater depth. We suggest a new set of problems for using computer vision and machine learning to automate pain assessment. We discuss below challenges and promising directions, and our ongoing research.

- **Genuine versus Posed Pain**

  Our contribution to Genuine versus Posed Pain underlined the importance of Vision Transformers to detect the subtle changes in people's faces to discriminate between the two classes. Despite the importance of the temporal aspect in the treatment of pain, the spatial information are still of a high interest in automatic pain assessment. For this reason, we are currently investigating the use of both Vision Transformers and CNNs to detect Genuine versus Posed pain.

- **Data gathering**

  The availability of data, which might be difficult to gather, is a significant obstacle for the progression of pain identification. Sharing datasets is thus necessary to enhance the pace of advancement. An ideal dataset would be multimodal, have annotations of high quality, cover not just pain but also other important states to evaluate specificity and thereby reduce the number of false alarms, and be released with rigorous assessment criteria to increase the comparability of findings. It is necessary to have shared datasets of clinical pain in order to validate recognition algorithms in real use cases, such as with patients who are going through post-operative stages or who have dementia.

  The findings of future research should be validated on several datasets to demonstrate consistent performance over a broad variety of data and to evaluate how well a system generalizes to diverse contexts, medical populations, pain types, etc. Also, ready-to-use recognition systems need to be tested in independent clinical trials before they can be shown to have any therapeutic value.

- **Origin of pain**

An observer will have a very difficult time comprehending the nature of the painful experience if they do not engage in dialogue with the person who is experiencing it. If the training data for the computational models were expanded to incorporate more information regarding the cause of the patient's suffering, then it is possible that it would be feasible to identify the source of a patient's pain based on the facial expressions that the patient is displaying. The most significant obstacle that stands in the way of achieving this objective is the lack of a system that is able to establish connections between particular facial expressions, the duration of such expressions, and the response of the subject to the source of the pain. Having a dataset with many different types of pain causes is another problem to overcome.

Bibliography

[1] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. "Face description with local binary patterns: Application to face recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 28.12 (2006), pp. 2037–2041.

[2] MAH Akhand et al. "Facial Emotion Recognition Using Transfer Learning in the Deep CNN". In: *Electronics* 10.9 (2021), p. 1036.

[3] Raphael Angulu, Jules R Tapamo, and Aderemi O Adewumi. "Age estimation via face images: a survey". In: *EURASIP Journal on Image and Video Processing* 2018.1 (2018), pp. 1–35.

[4] Bradley M Appelhans and Linda J Luecken. "Heart rate variability and pain: associations of two interrelated homeostatic processes". In: *Biological psychology* 77.2 (2008), pp. 174–182.

[5] Ahmed Bilal Ashraf et al. "The painful face: Pain expression recognition using active appearance models". In: *Proceedings of the 9th international conference on Multimodal interfaces*. 2007, pp. 9–14.

[6] Min SH Aung et al. "The automatic detection of chronic pain-related expression: requirements, challenges and the multimodal EmoPain dataset". In: *IEEE transactions on affective computing* 7.4 (2015), pp. 435–451.

[7] Name Author and Name Author. "Article title". In: *Journal title* Volume number (Issuse number 1986), Pagestart–pageend.

[8] Murat Aydede. "Defending the IASP definition of pain". In: *The Monist* 100.4 (2017), pp. 439–464.

[9] Tadas Baltrusaitis et al. "Openface 2.0: Facial behavior analysis toolkit". In: *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*. IEEE. 2018, pp. 59–66.

[10] Andrea Bandini et al. "Automatic Detection of Orofacial Impairment in Stroke." In: *Interspeech*. 2018, pp. 1711–1715.

[11] Ghazal Bargshady et al. "A joint deep neural network model for pain recognition from face". In: *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*. IEEE. 2019, pp. 52–56.

[12] Ghazal Bargshady et al. "Enhanced deep learning algorithm development to detect pain intensity from facial expression images". In: *Expert Systems with Applications* 149 (2020), p. 113305.

[13] Marian Bartlett et al. “Data mining spontaneous facial behavior with automatic expression coding”. In: *Verbal and Nonverbal Features of Human-Human and Human-Machine Interaction*. Springer, 2008, pp. 1–20.

[14] Marco Bellantonio et al. “Spatio-temporal pain recognition in cnn-based super-resolved facial images”. In: *Video Analytics. Face and Facial Expression Recognition and Audience Measurement*. Springer, 2016, pp. 151–162.

[15] Eduardo E Benarroch. “Pain-autonomic interactions: a selective review”. In: *Clinical Autonomic Research* 11.6 (2001), pp. 343–349.

[16] Ridha Ilyas Bendjillali et al. “Improved facial expression recognition based on DWT feature for deep CNN”. In: *Electronics* 8.3 (2019), p. 324.

[17] Charles F Bond Jr and Bella M DePaulo. “Accuracy of deception judgments”. In: *Personality and social psychology Review* 10.3 (2006), pp. 214–234.

[18] Wolfram Boucsein. *Electrodermal activity*. Springer Science & Business Media, 2012.

[19] Sheryl Brahnam, Loris Nanni, and Randall S Sexton. “Neonatal Facial Pain Detection Using NNSOA and LSVM.” In: *IPCV*. 2008, pp. 352–357.

[20] Sheryl Brahnam et al. “SVM classification of neonatal facial images of pain”. In: *International Workshop on Fuzzy Logic and Applications*. Springer. 2005, pp. 121–128.

[21] Siqi Cao et al. “How Can AI Recognize Pain and Express Empathy”. In: *arXiv preprint arXiv:2110.04249* (2021).

[22] Pilar Carrera-Levillain and Jose-Miguel Fernandez-Dols. “Neutral faces in context: Their emotional meaning and their function”. In: *Journal of Nonverbal Behavior* 18.4 (1994), pp. 281–299.

[23] C Richard Chapman et al. “Phasic pupil dilation response to noxious stimulation in normal volunteers: relationship to brain evoked potentials and pain report”. In: *Psychophysiology* 36.1 (1999), pp. 44–52.

[24] Sneha Chaudhari et al. “An attentive survey of attention models”. In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 12.5 (2021), pp. 1–32.

[25] Jixu Chen et al. “Person-specific expression recognition with transfer learning”. In: *2012 19th IEEE International Conference on Image Processing*. IEEE. 2012, pp. 2621–2624.

[26] Junkai Chen, Zheru Chi, and Hong Fu. “A new approach for pain event detection in video”. In: *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE. 2015, pp. 250–254.

[27] Junkai Chen, Zheru Chi, and Hong Fu. "A new framework with multiple tasks for detecting and locating pain events in video". In: *Computer Vision and Image Understanding* 155 (2017), pp. 113–123.

[28] Yaqi Chu et al. "Physiological signal-based method for measurement of pain intensity". In: *Frontiers in neuroscience* 11 (2017), p. 279.

[29] William H Cordell et al. "The high prevalence of pain in emergency medical care". In: *The American journal of emergency medicine* 20.3 (2002), pp. 165–169.

[30] Kenneth D Craig, Kenneth M Prkachin, and Ruth E Grunau. "The facial expression of pain." In: (2011).

[31] Laduona Dai, Joost Broekens, and Khiet P Truong. "Real-time pain detection in facial expressions for health robotics". In: *2019 8th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*. IEEE. 2019, pp. 277–283.

[32] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[33] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[34] Joy Egede, Michel Valstar, and Brais Martinez. "Fusing deep learned and hand-crafted features of appearance, shape, and dynamics for automatic pain estimation". In: *2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017)*. IEEE. 2017, pp. 689–696.

[35] Paul Ekman, Wallace V Friesen, and Phoebe Ellsworth. *Emotion in the human face: Guidelines for research and an integration of findings*. Vol. 11. Elsevier, 2013.

[36] Safaa El Morabit et al. "Automatic pain estimation from facial expressions: a comparative analysis using off-the-shelf CNN architectures". In: *Electronics* 10.16 (2021), p. 1926.

[37] Yun Fu, Guodong Guo, and Thomas S Huang. "Age synthesis and estimation via faces: A survey". In: *IEEE transactions on pattern analysis and machine intelligence* 32.11 (2010), pp. 1955–1976.

[38] Xiaohong Gao, Yu Qian, and Alice Gao. "Covid-vit: Classification of covid-19 from ct chest images based on vision transformer models". In: *arXiv preprint arXiv:2107.01682* (2021).

[39] Anjith George and Sébastien Marcel. "On the effectiveness of vision transformers for zero-shot face anti-spoofing". In: *2021 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE. 2021, pp. 1–8.

[40] Afsane Ghasemi et al. "Social signal processing for pain monitoring using a hidden conditional random field". In: *2014 IEEE Workshop on Statistical Signal Processing (SSP)*. IEEE. 2014, pp. 61–64.

[41] Behnood Gholami, Wassim M Haddad, and Allen R Tannenbaum. "Relevance vector machine learning for neonate pain intensity assessment using digital imaging". In: *IEEE Transactions on biomedical engineering* 57.6 (2010), pp. 1457–1466.

[42] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[43] Julie Gregory and Linda McGowan. "An examination of the prevalence of acute pain for hospitalised adult patients: a systematic review". In: *Journal of clinical nursing* 25.5-6 (2016), pp. 583–598.

[44] Diego L Guarin et al. "Toward an automatic system for computer-aided assessment in facial palsy". In: *Facial Plastic Surgery & Aesthetic Medicine* 22.1 (2020), pp. 42–49.

[45] Zakia Hammal and Jeffrey F Cohn. "Automatic detection of pain intensity". In: *Proceedings of the 14th ACM international conference on Multimodal interaction*. 2012, pp. 47–52.

[46] Zakia Hammal and Miriam Kunz. "Pain monitoring: A dynamic and context-sensitive system". In: *Pattern Recognition* 45.4 (2012), pp. 1265–1280.

[47] Zakia Hammal et al. "Spontaneous pain expression recognition in video sequences". In: *Visions of Computer Science-BCS International Academic Conference*. 2008, pp. 191–210.

[48] Mohammad A Haque et al. "Deep multimodal pain recognition: a database and comparison of spatio-temporal visual modalities". In: *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*. IEEE. 2018, pp. 250–257.

[49] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[50] Shu He, JohnJ Soraghan, and BrianF O'Reilly. "Biomedical image sequence analysis with application to automatic quantitative assessment of facial paralysis". In: *EURASIP journal on image and video processing* 2007 (2007), pp. 1–11.

[51] Young-Jin Heo et al. “Deepfake detection scheme based on vision transformer and distillation”. In: *arXiv preprint arXiv:2104.01353* (2021).

[52] Keela Herr et al. “Pain assessment in the patient unable to self-report: position statement with clinical practice recommendations”. In: *Pain management nursing* 12.4 (2011), pp. 230–250.

[53] Marilyn L Hill and Kenneth D Craig. “Detecting deception in pain expressions: the structure of genuine and deceptive facial displays”. In: *Pain* 98.1-2 (2002), pp. 135–144.

[54] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. “Distilling the knowledge in a neural network”. In: *arXiv preprint arXiv:1503.02531* 2.7 (2015).

[55] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[56] Gao Huang et al. “Densely connected convolutional networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[57] Yibo Huang et al. “HybNet: a hybrid network structure for pain intensity estimation”. In: *The Visual Computer* 38.3 (2022), pp. 871–882.

[58] Shashank Jaiswal, Joy Egede, and Michel Valstar. “Deep learned cumulative attribute regression”. In: *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE. 2018, pp. 715–722.

[59] Shan Jia et al. “Detection of genuine and posed facial expressions of emotion: databases and methods”. In: *Frontiers in Psychology* 11 (2021), p. 580287.

[60] Markus Kächele et al. “Adaptive confidence learning for the personalization of pain intensity estimation systems”. In: *Evolving Systems* 8.1 (2017), pp. 71–83.

[61] Markus Kächele et al. “Multimodal data fusion for person-independent, continuous estimation of pain intensity”. In: *International Conference on Engineering Applications of Neural Networks*. Springer. 2015, pp. 275–285.

[62] Sebastian Kaltwang, Sinisa Todorovic, and Maja Pantic. “Doubly sparse relevance vector machine for continuous facial behavior estimation”. In: *IEEE transactions on pattern analysis and machine intelligence* 38.9 (2015), pp. 1748–1761.

[63] Ioannis Karamitsos, Ilham Seladji, and Sanjay Modak. “A Modified CNN Network for Automatic Pain Identification Using Facial Expressions”. In: *Journal of Software Engineering and Applications* 14.8 (2021), pp. 400–417.

[64] Shehzad Khalid and R Shane Tubbs. “Neuroanatomy and neuropsychology of pain”. In: *Cureus* 9.10 (2017).

[65] Salman Khan et al. “Transformers in vision: A survey”. In: *ACM Computing Surveys (CSUR)* (2021).

[66] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[67] Byoung Chul Ko. “A brief review of facial emotion recognition based on visual information”. In: *sensors* 18.2 (2018), p. 401.

[68] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012).

[69] Eva G Krumhuber and Antony SR Manstead. “Can Duchenne smiles be feigned? New evidence on felt and false smiles.” In: *Emotion* 9.6 (2009), p. 807.

[70] Kaustubh Kulkarni et al. “Automatic recognition of facial displays of unfelt emotions”. In: *IEEE transactions on affective computing* 12.2 (2018), pp. 377–390.

[71] Gwen C Littlewort, Marian Stewart Bartlett, and Kang Lee. “Automatic coding of facial expressions displayed during posed and genuine pain”. In: *Image and Vision Computing* 27.12 (2009), pp. 1797–1803.

[72] Gwen C Littlewort, Marian Stewart Bartlett, and Kang Lee. “Faces of pain: automated measurement of spontaneousallfacial expressions of genuine and posed pain”. In: *Proceedings of the 9th international conference on Multimodal interfaces*. 2007, pp. 15–21.

[73] Dianbo Liu, Peng Fengjiao, Rosalind Picard, et al. “DeepFaceLIFT: interpretable personalized models for automatic estimation of self-reported pain”. In: *IJCAI 2017 Workshop on Artificial Intelligence in Affective Computing*. PMLR. 2017, pp. 1–16.

[74] Liliana Lo Presti and Marco La Cascia. “Using Hankel matrices for dynamics-based facial emotion recognition and pain detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2015, pp. 26–33.

[75] Beth Logan. “Mel frequency cepstral coefficients for music modeling”. In: *In International Symposium on Music Information Retrieval*. Citeseer. 2000.

[76] Daniel Lopez-Martinez and Rosalind Picard. “Continuous pain intensity estimation from autonomic signals with recurrent neural networks”. In: *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2018, pp. 5624–5627.

[77] Daniel Lopez Martinez, Rosalind Picard, et al. “Personalized automatic estimation of self-reported pain intensity from facial expressions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017, pp. 70–79.

[78] Daniel Lopez-Martinez, Ognjen Rudovic, and Rosalind Picard. “Physiological and behavioral profiling for nociceptive pain estimation using personalized multitask learning”. In: *arXiv preprint arXiv:1711.04036* (2017).

[79] David G Lowe. “Distinctive image features from scale-invariant keypoints”. In: *International journal of computer vision* 60.2 (2004), pp. 91–110.

[80] Patrick Lucey et al. “Automatically detecting pain in video through facial action units”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 41.3 (2010), pp. 664–674.

[81] Patrick Lucey et al. “Painful data: The UNBC-McMaster shoulder pain expression archive database”. In: *2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG)*. IEEE. 2011, pp. 57–64.

[82] Patrick Lucey et al. “Painful monitoring: Automatic pain monitoring using the UNBC-McMaster shoulder pain expression archive database”. In: *Image and Vision Computing* 30.3 (2012), pp. 197–205.

[83] Mary E Lynch. *The need for a Canadian pain strategy*. 2011.

[84] Warren S McCulloch and Walter Pitts. “A logical calculus of the ideas immanent in nervous activity”. In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133.

[85] Anita Mehta and Lisa S Chan. “Understanding of the concept of" total pain": a prerequisite for pain control”. In: *Journal of Hospice & Palliative Nursing* 10.1 (2008), pp. 26–32.

[86] Ronald Melzack and Kenneth L Casey. “Sensory, motivational, and central control determinants of pain: a new conceptual model”. In: *The skin senses* 1 (1968), pp. 423–43.

[87] Hongying Meng and Nadia Bianchi-Berthouze. “Affective state level recognition in naturalistic facial and vocal expressions”. In: *IEEE Transactions on Cybernetics* 44.3 (2013), pp. 315–328.

[88] Zuheng Ming et al. “A survey on anti-spoofing methods for facial recognition with rgb cameras of generic consumer devices”. In: *Journal of Imaging* 6.12 (2020), p. 139.

[89] Anita Mitchell and Barbara J Boss. "Adverse effects of pain on the nervous system of newborns and young children: a review of the literature". In: *Journal of Neuroscience Nursing* 34.5 (2002), p. 228.

[90] Loris Nanni, Sheryl Brahnam, and Alessandra Lumini. "A local approach based on a local binary patterns variant texture descriptor for classifying pain states". In: *Expert Systems with Applications* 37.12 (2010), pp. 7888–7894.

[91] Robert Niese et al. "Towards pain recognition in post-operative phases using 3d-based features from video and support vector machines". In: *International Journal of Digital Content Technology and its Applications* 3.4 (2009), pp. 21–31.

[92] Temitayo A Olugbade et al. "Pain level recognition using kinematics and muscle activity for physical rehabilitation in chronic pain". In: *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE. 2015, pp. 243–249.

[93] Keiron O'Shea and Ryan Nash. "An introduction to convolutional neural networks". In: *arXiv preprint arXiv:1511.08458* (2015).

[94] Seho Park et al. "Differences in facial expressions between spontaneous and posed smiles: Automated method by action units and three-dimensional facial landmarks". In: *Sensors* 20.4 (2020), p. 1199.

[95] Adam Paszke et al. "Automatic differentiation in pytorch". In: (2017).

[96] Eulália Silva Dos Santos Pinheiro et al. "Electroencephalographic patterns in chronic pain: a systematic review of the literature". In: *PloS one* 11.2 (2016), e0149085.

[97] Liliana Lo Presti and Marco La Cascia. "Boosting Hankel matrices for face emotion recognition and pain detection". In: *Computer Vision and Image Understanding* 156 (2017), pp. 19–33.

[98] Kenneth M Prkachin and Patricia E Solomon. "The structure, reliability and validity of pain expression: Evidence from patients with shoulder pain". In: *Pain* 139.2 (2008), pp. 267–274.

[99] Xiaoqian Qin, Dakun Liu, and Dong Wang. "A literature survey on kinship verification through facial images". In: *Neurocomputing* 377 (2020), pp. 213–224.

[100] Andrei Racoviţeanu et al. "Spontaneous emotion detection by combined learned and fixed descriptors". In: *2019 International Symposium on Signals, Circuits and Systems (ISSCS)*. IEEE. 2019, pp. 1–4.

[101] Joseph P Robinson et al. “Families in the wild (fiw) large-scale kinship image database and benchmarks”. In: *Proceedings of the 24th ACM international conference on Multimedia*. 2016, pp. 242–246.

[102] Pau Rodriguez et al. “Deep pain: Exploiting long short-term memory networks for facial expression classification”. In: *IEEE transactions on cybernetics* (2017).

[103] Bernardino Romera-Paredes et al. “Transfer learning to account for idiosyncrasy in face and body expressions”. In: *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*. IEEE. 2013, pp. 1–6.

[104] Sourav Dey Roy et al. “An approach for automatic pain detection through facial expression”. In: *Procedia Computer Science* 84 (2016), pp. 99–106.

[105] Sylvain Roy et al. “STOIC: A database of dynamic and static faces expressing highly recognizable emotions”. In: *Montréal, Canada: Université De Montréal* (2007).

[106] Ognjen Rudovic, Vladimir Pavlovic, and Maja Pantic. “Automatic pain intensity estimation with heteroscedastic conditional ordinal random fields”. In: *International Symposium on Visual Computing*. Springer. 2013, pp. 234–243.

[107] Marcella Sacco et al. “The relationship between blood pressure and pain”. In: *The journal of clinical hypertension* 15.8 (2013), pp. 600–605.

[108] Chisa Saito, Katsutoshi Masai, and Maki Sugimoto. “Classification of spontaneous and posed smiles by photo-reflective sensors embedded with smart eyewear”. In: *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction*. 2020, pp. 45–52.

[109] Mark Sandler et al. “Mobilenetv2: Inverted residuals and linear bottlenecks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4510–4520.

[110] Anvita Saxena, Ashish Khanna, and Deepak Gupta. “Emotion recognition and detection methods: A comprehensive survey”. In: *Journal of Artificial Intelligence and Systems* 2.1 (2020), pp. 53–79.

[111] A Schnitzler and M Ploner. “Neurophysiology and functional neuroanatomy of pain perception”. In: *Journal of clinical neurophysiology* 17.6 (2000), pp. 592–603.

[112] Karan Sikka et al. “Automated assessment of children’s postoperative pain using computer vision”. In: *Pediatrics* 136.1 (2015), e124–e131.

[113] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[114] Cheryl L Stucky, Michael S Gold, and Xu Zhang. “Mechanisms of pain”. In: *Proceedings of the National Academy of Sciences* 98.21 (2001), pp. 11845–11846.

[115] Christian Szegedy et al. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[116] Chuanqi Tan et al. "A survey on deep transfer learning". In: *International conference on artificial neural networks*. Springer. 2018, pp. 270–279.

[117] Mohammad Tavakolian, Carlos Guillermo Bermudez Cruces, and Abdenour Hadid. "Learning to detect genuine versus posed pain from facial expressions using residual generative adversarial networks". In: *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*. IEEE. 2019, pp. 1–8.

[118] Mohammad Tavakolian and Abdenour Hadid. "A spatiotemporal convolutional neural network for automatic pain intensity estimation from facial dynamics". In: *International Journal of Computer Vision* 127.10 (2019), pp. 1413–1425.

[119] Mohammad Tavakolian and Abdenour Hadid. "Deep binary representation of facial expressions: a novel framework for automatic pain intensity recognition". In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE. 2018, pp. 1952–1956.

[120] Mohammad Tavakolian, Miguel Bordallo Lopez, and Li Liu. "Self-supervised pain intensity estimation from facial videos via statistical spatiotemporal distillation". In: *Pattern Recognition Letters* 140 (2020), pp. 26–33.

[121] Astrid Juhl Terkelsen et al. "Acute pain increases heart rate: differential mechanisms during rest and mental stress". In: *Autonomic Neuroscience* 121.1-2 (2005), pp. 101–109.

[122] Patrick Thiam, Viktor Kessler, and Friedhelm Schwenker. "Hierarchical Combination of Video Features for Personalised Pain Level Recognition." In: *ESANN*. 2017, pp. 465–470.

[123] Patrick Thiam et al. "Exploring deep physiological models for nociceptive pain recognition". In: *Sensors* 19.20 (2019), p. 4503.

[124] Hugo Touvron et al. "Training data-efficient image transformers & distillation through attention". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 10347–10357.

[125] RD Treede. "Transduction and transmission properties of primary nociceptive afferents." In: *Rossiiskii fiziologicheskii zhurnal imeni IM Sechenova* 85.1 (1999), pp. 205–211.

[126] Fu-Sheng Tsai et al. "Toward Development and Evaluation of Pain Level-Rating Scale for Emergency Triage based on Vocal Characteristics and Facial Expressions." In: *Interspeech*. 2016, pp. 92–96.

[127] Dennis C Turk and Ronald Melzack. "The measurement of pain and the assessment of people experiencing pain." In: (2011).

[128] Pieter Van Der Geld et al. "Tooth display and lip position during spontaneous and posed smiling in adults". In: *Acta Odontologica Scandinavica* 66.4 (2008), pp. 207–213.

[129] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[130] Maria Velana et al. "The senseemotion database: A multimodal database for the development and systematic validation of an automatic pain-and emotion-recognition system". In: *Multimodal Pattern Recognition of Social Signals in Human-Computer-Interaction: 4th IAPR TC 9 Workshop, MPRSS 2016, Cancun, Mexico, December 4, 2016, Revised Selected Papers 4*. Springer. 2017, pp. 127–139.

[131] Molly T Vogt et al. "Analgesic usage for low back pain: impact on health care costs and service use". In: *Spine* 30.9 (2005), pp. 1075–1081.

[132] Robert Walecki et al. "A framework for joint estimation and guided annotation of facial action unit intensity". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2016, pp. 9–17.

[133] Joseph Walsh, Christopher Eccleston, and Edmund Keogh. "Pain communication through body posture: The development and validation of a stimulus set". In: *PAIN®* 155.11 (2014), pp. 2282–2290.

[134] Steffen Walter et al. ""What About Automated Pain Recognition for Routine Clinical Use?" A Survey of Physicians and Nursing Staff on Expectations, Requirements, and Acceptance". In: *Frontiers in medicine* 7 (2020), p. 566278.

[135] Chongyang Wang et al. "Recurrent network based automatic detection of chronic pain protective behavior using mocap and semg data". In: *Proceedings of the 23rd international symposium on wearable computers*. 2019, pp. 225–230.

[136] Feng Wang et al. "Regularizing face verification nets for pain intensity regression". In: *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2017, pp. 1087–1091.

[137] Sinong Wang et al. "Linformer: Self-attention with linear complexity". In: *arXiv preprint arXiv:2006.04768* (2020).

[138] Xiaoyu Wang, Tony X Han, and Shuicheng Yan. "An HOG-LBP human detector with partial occlusion handling". In: *2009 IEEE 12th international conference on computer vision*. IEEE. 2009, pp. 32–39.

[139] Philipp Werner et al. “Automatic pain assessment with facial activity descriptors”. In: *IEEE Transactions on Affective Computing* 8.3 (2016), pp. 286–299.

[140] Philipp Werner et al. “Automatic pain recognition from video and biomedical signals”. In: *2014 22nd international conference on pattern recognition*. IEEE. 2014, pp. 4582–4587.

[141] Philipp Werner et al. “Automatic recognition methods supporting pain assessment: A survey”. In: *IEEE Transactions on Affective Computing* (2019).

[142] Philipp Werner et al. “Head movements and postures as pain behavior”. In: *PloS one* 13.2 (2018), e0192767.

[143] Philipp Werner et al. “Towards pain monitoring: Facial expression, head pose, a new database, an automatic system and remaining challenges”. In: *Proceedings of the British Machine Vision Conference*. 2013, pp. 1–13.

[144] Philipp Werner et al. “Twofold-multimodal pain recognition with the X-ITE pain database”. In: *2019 8th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*. IEEE. 2019, pp. 290–296.

[145] Amanda C de C Williams. “Facial expression of pain: an evolutionary account”. In: *Behavioral and brain sciences* 25.4 (2002), pp. 439–455.

[146] Amanda C de C Williams and Kenneth D Craig. “Updating the definition of pain”. In: *Pain* 157.11 (2016), pp. 2420–2423.

[147] WD Willis and KN Westlund. “Neuroanatomy of the pain system and of the pathways that modulate pain”. In: *Journal of clinical neurophysiology: official publication of the American Electroencephalographic Society* 14.1 (1997), p. 2.

[148] Jia Xiang and Gengming Zhu. “Joint face detection and facial expression recognition with MTCNN”. In: *2017 4th international conference on information science and control engineering (ICISCE)*. IEEE. 2017, pp. 424–427.

[149] Saining Xie et al. “Aggregated residual transformations for deep neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1492–1500.

[150] Philip Yancey and Paul W Brand. *The Gift of Pain: Why We Hurt & what We Can Do about it*. Zondervan, 1997.

[151] Ruijing Yang et al. “On pain assessment from facial videos using spatio-temporal local descriptors”. In: *2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE. 2016, pp. 1–6.

[152] Zuhair Zafar and Nadeem Ahmad Khan. "Pain intensity evaluation through facial action units". In: *2014 22nd International Conference on Pattern Recognition*. IEEE. 2014, pp. 4696–4701.

[153] Meigui Zhang, Kehui Zeng, and Jinwei Wang. "A Survey on Face Anti-Spoofing Algorithms". In: *Journal of Information Hiding and Privacy Protection* 2.1 (2020), p. 21.

[154] Zheng Zhang et al. "Multimodal spontaneous emotion corpus for human behavior analysis". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3438–3446.

[155] Rui Zhao et al. "Facial expression intensity estimation using ordinal information". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3466–3474.

[156] Jing Zhou et al. "Recurrent convolutional neural network regression for continuous pain intensity estimation in video". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2016, pp. 84–92.

[157] Sigridur Zoëga et al. "Quality pain management in the hospital setting from the patient's perspective". In: *Pain Practice* 15.3 (2015), pp. 236–246.