Prepoznavanje emocija na temelju izraza lica tenisača tijekom igre

Kopić, Palma

UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

MASTER THESIS No. 475

EMOTION RECOGNITION USING FACIAL EXPRESSIONS OF TENNIS PLAYERS DURING MATCHES

Palma Kopić

Zagreb, June 2024

UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

MASTER THESIS No. 475

EMOTION RECOGNITION USING FACIAL EXPRESSIONS OF TENNIS PLAYERS DURING MATCHES

Palma Kopić

Zagreb, June 2024

UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Zagreb, 04 March 2024

MASTER THESIS ASSIGNMENT No. 475

Student:	**Palma Kopić (0036518670)**
Study:	Computing
Profile:	Software Engineering and Information Systems
Mentor:	assoc. prof. Marina Bagić Babac

Title: **Emotion recognition using facial expressions of tennis players during matches**

Description:

This thesis investigates the use of machine learning techniques for emotion recognition based on facial expressions exhibited by tennis players during matches, aiming to understand their correlation with performance outcomes. The analysis involves the classification and characterization of emotions displayed by players throughout matches, examining how these emotions evolve over time and their impact on match outcomes. A publicly available dataset on Internet containing images of people in everyday environments classified according to their emotions (EMOTIC Dataset) will be used for machine learning of the model.

Submission date: 28 June 2024

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

DIPLOMSKI ZADATAK br. 475

Pristupnica:	**Palma Kopić (0036518670)**
Studij:	Računarstvo
Profil:	Programsko inženjerstvo i informacijski sustavi
Mentorica:	izv. prof. dr. sc. Marina Bagić Babac

Zadatak: **Prepoznavanje emocija na temelju izraza lica tenisača tijekom igre**

Opis zadatka:

Ovaj diplomski rad istražuje korištenje tehnika strojnog učenja za prepoznavanje emocija na temelju izraza lica koje tenisači pokazuju tijekom igre, s ciljem razumijevanja korelacije emocija s rezultatima izvedbe. Analiza uključuje klasifikaciju i karakterizaciju emocija koje igrači pokazuju tijekom igre, ispitivanje kako se te emocije razvijaju tijekom vremena i njihov utjecaj na ishod igre. Za strojno učenje modela koristit će se javno dostupni skup podataka s interneta (tzv. EMOTIC dataset) koji sadrži slike ljudi u svakodnevnim okruženjima klasificirane prema njihovim emocijama.

Rok za predaju rada: 28. lipnja 2024.

UNIVERSITY OF ZAGREB

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

MASTER THESIS ASSIGNMENT No. 475

Emotion recognition using facial expressions of tennis players during matches

Palma Kopić

Zagreb, September 2024.

UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Zagreb, 04 March 2024

MASTER THESIS ASSIGNMENT No. 475

Student:	**Palma Kopić (0036518670)**
Study:	Computing
Profile:	Software Engineering and Information Systems
Mentor:	assoc. prof. Marina Bagić Babac

Title: **Emotion recognition using facial expressions of tennis players during matches**

Description:

This thesis investigates the use of machine learning techniques for emotion recognition based on facial expressions exhibited by tennis players during matches, aiming to understand their correlation with performance outcomes. The analysis involves the classification and characterization of emotions displayed by players throughout matches, examining how these emotions evolve over time and their impact on match outcomes. A publicly available dataset on Internet containing images of people in everyday environments classified according to their emotions (EMOTIC Dataset) will be used for machine learning of the model.

Submission date: 28 June 2024

Prije svega, hvala mami i tati što su bili uz mene ovih 6 godina koliko je trajala moja fakultetska avantura i što ni u jednom trenutku nisu sumnjali u mene, nego su me pustili da sve riješim, podržavajući svaki moj korak. Beskrajno sam zahvalna.

Posebna hvala baki i dedi što su sa mnom proživljavali svaki moj ispit i bili tu da pruže potporu kada god je zatrebalo. Presretna sam što sam imala priliku učiniti ih ponosnima.

Hvala i mentorici koja me vodila kroz cijeli moj diplomski studij, strpljivo odgovarala na sva pitanja te pružala podršku, čak i u najizazovnijim trenutcima.

Content

Introduction

In recent years, the integration of technology has revolutionized the field of sport, from Video Assistant Referees (VARs) in football [1], Hawkeye in tennis [2], and similar tools that dissect every crucial moment, to wearable sensors that monitor physical performance in everyday life. Furthermore, the technological (r)evolution extends beyond just the physical side of sports, encompassing the psychological and emotional aspects as well. Facial emotion recognition (FER), a cutting-edge application of artificial intelligence, offers a new frontier in understanding and enhancing the mental game of athletes.

FER technology, which analyzes facial expressions to detect emotions, has significant potential in the realm of tennis, since tennis is a sport of mental fortitude, as much as it is of physical prowess, requiring the players to employ various psychological strategies to maintain focus, remain calm, and recover quickly from setbacks throughout the whole match lasting up to 10 hours long in extreme cases [3]. The ability to manage emotions during a match can either make or break a tennis player, making mental agility a pivotal factor in the performance of tennis athletes. Therefore, outward emotions displayed by players during matches provide a window into their mental state, and their recognition and analysis can offer valuable insights into the dynamics of the game and the mental challenges that the players face.

This thesis explores the application of FER in the context of tennis, aiming to bridge the gap between physical and mental performance analysis. First, a new dataset containing videos of players' emotions is created, capturing a wide range of emotional expressions during matches. Later, the videos are meticulously processed to ensure consistency and quality, preparing them for use in machine learning models with architectures that include 2D CNNs, 3D CNNs, and transformers, which are implemented to identify the most accurate model for emotion recognition. The results are then thoroughly analyzed to evaluate the effectiveness of each approach in understanding emotions during tennis performance.

By leveraging FER, the goal is to provide a tool that can assist players and coaches in managing these emotions more effectively. As the sport continues to evolve with technological advancements, integrating FER into tennis could mark a significant step

forward in optimizing the mental aspects of the game, ultimately leading to better performance and more consistent success on the court.

1. Tennis

Tennis is a widely spread sport with a history that can be traced back to the 11th and 12th century France. It rapidly gained popularity in modern society during the second half of the 20th century becoming one of the most popular sports today.

Played either by opposing individuals (singles) or by two teams of two players (doubles), the sport is played on a rectangular court divided by a net.

A single match is divided into sets, which are divided into games. To win a set, a player must win at least six games by a margin of two. In the event of a 6-6 tie in games, a tiebreaker is often used to decide the set, where players alternate serves, and the first to reach seven points with a two-point lead wins the set. Matches are usually played as best-of-three or best-of-five sets, depending on the tournament rules.

The scoring within games follows a unique system: points are awarded as 15, 30, 40, and game, with the requirement to win by at least two points to secure the game. If both players reach 40, known as "*deuce*," the game continues until one player wins two consecutive points.

In line with the popularity of tennis, there are thousands of tournaments, for both male and female players, played across the year globally, ranging from local and regional competitions to international events. The most prestigious, major tournaments, also known as Grand Slam tournaments, stand out as the pinnacle of tennis excellence, offering the highest levels of competition and recognition. These tournaments include Wimbledon, the US Open, the French Open, and the Australian Open. Distinguished by their unique court surfaces, Grand Slams add to the variety and challenge of the sport - Wimbledon is played on grass, the US Open on hard court, the French Open on clay, and the Australian Open on hard court, each offering a unique experience of playing the sport, as well as the watching.

1.1. Technology in Tennis

In recent years, technology has taken up a key transformative role in tennis, dramatically altering how the game is played, officiated, analyzed, and watched. Taking Hawk-Eye as an example, its cutting-edge technology has revolutionized line calling by providing accurate, real-time data on whether the ball is in or out. This technology not only enhanced the fairness

in the game but also reduced the psychological pressure on players and officials, knowing that decisions can be objectively verified [2]. In addition to Hawk-Eye, advancements in racket technology have allowed for more powerful and precise shots, while wearable sensors now monitor players' physical performance, tracking metrics like heart rate, speed, and even sweat levels to optimize training and in-match strategy.

Technology giants have had a big impact on tennis, with the greatest impact coming from IBM. IBM has found tennis to be the right ground for technological innovations, providing advanced data analytics and AI-driven insights for major tournaments, such as Wimbledon and the US Open [4] [5] [6], and using technologies like *Watson* to offer real-time match statistics player performance analysis and predictive insights [7]. Furthermore, IBM's *PointStream* technology delivers detailed point-by-point analysis, helping both fans and commentators understand the intricacies of the matches [8].

As technology continues to evolve, its influence on tennis will likely expand further, enhancing the experience for players and fans alike while shaping the future of the sport. So, with ongoing innovations on the horizon, the intersection of technology and tennis promises to redefine what is possible on and off the court.

2. Emotions

Emotions are fundamental to the human experience, greatly shaping thoughts, behaviors, and interactions with the world of every individual. While individual experiences of emotions are essentially unique, universal definitions are provided to make emotions less abstract.

Emotions are described as brief affective states elicited as responses to discrete stimuli whereas moods are defined as long-lasting affective states [9]. Moreover, discrete emotions are defined as specific, identifiable emotional states, such as anger, happiness, or sadness that can easily be distinguished from one another due to their distinct causes and unique physiological patterns [10].

Emotional states can further be described using terms like **valence** and **arousal**. Valence refers to the pleasantness or the unpleasantness of an emotion, helping to understand whether an emotion is intrinsically experienced as positive or negative, thus providing insight into its hedonic tone. On the other hand, arousal refers to the intensity of an emotional experience, indicating the energy or activation associated with the emotion, ranging from calming low-arousal states to stimulating high-arousal states [11], [12]. For instance, joy is usually characterized as a high-valence, high-arousal state, while sadness is low-valence and low-arousal, reflecting their impact on both the psychological and physiological state of an individual. Since emotions have such an influence on everyday life, classifying emotions has also been extensively researched in the field of psychology resulting in various proposed models, such as:

- **Plutchik's Wheel of Emotions**, proposed by Robert Plutchik in the 1980s, is a model that organizes emotions into eight primary bipolar categories joy vs. sadness, trust vs. disgust, fear vs. anger, and surprise vs. anticipation, which can further be combined to form more complex emotions (e.g. joy and trust combined form love) [13]. This model visually represents how primary emotions interact and blend to create more nuanced emotional experiences.
- **Ekman's Basic Emotions** model presents emotions that are universally recognized across different cultures. The initial work categorized emotions into 6 categories: happiness, sadness, fear, anger, surprise, and disgust. Each of these emotions is expressed in individuals with a set of specific facial expressions and gestures

acknowledged in the same manner globally. For example, happiness is characterized by smiling - the corners of the mouth turning up, often accompanied by crinkling around the eyes. Fear is, on the other hand, identified with wide eyes, raised upper eyelids, tensed low eyelids, and eyebrows drawn together [14].

- **The Circumplex Model of Affect** was proposed by James Russell, and it maps emotions on a two-dimensional circular space, defined by arousal (high vs. low) and valence (positive vs. negative). Emotions are then plotted around the circumference of the circle, demonstrating how they relate to one another and vary in intensity and positivity [11] (Figure 2.1). Similar to Plutchik's Wheel of Emotions [13] this model also helps in understanding the complex interplay between different emotions and their underlying dimensions.

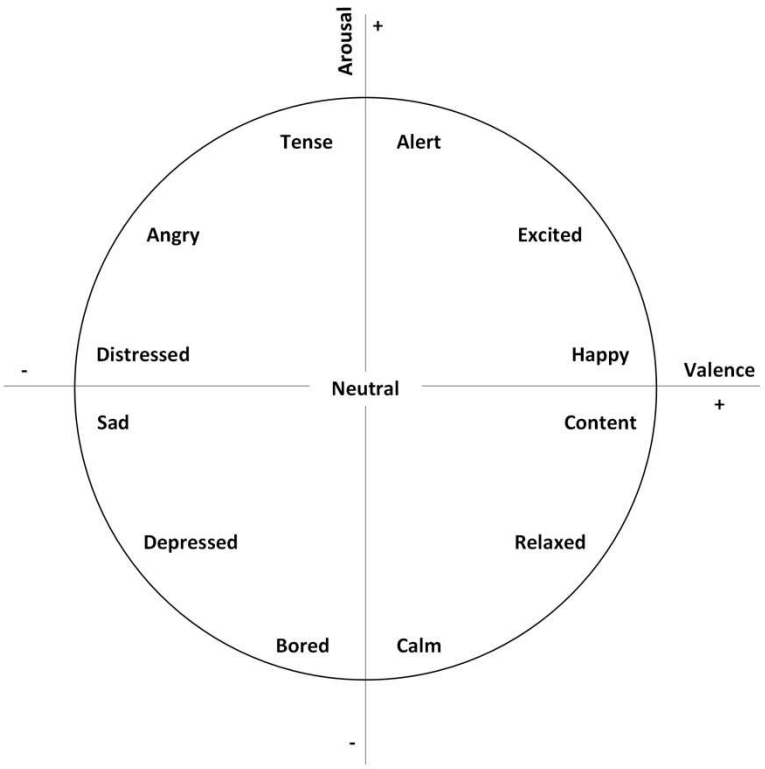

Figure 2.1 Circumplex model of affect

- James Russell opposes the idea of basic, biologically hardwired emotions in his later work with the **Core Affect and Psychological Construction of Emotion** model which suggests that emotions are constructed from more fundamental psychological ingredients such as core affect and conceptual knowledge to emphasize the role of individual experiences and cognitive processes in shaping emotional responses,

inherently suggesting the dynamic nature of emotions instead of the more static, fixed one.

2.1. Emotions in Tennis

Emotions play a critical role in all sports by influencing athletes' cognitive, affective, and behavioral responses [15]. The competitive environment, high pressure, and expectations represent situations not usually experienced in such combination by non-athletes which often evoke strong emotional reactions. Emotions in sports can arise from various sources such as competition stress, performance outcomes, and social interactions, and can impact the athletes' performance both positively and negatively.

Specifically in tennis, positive valence can manifest as joy after winning a crucial point or the feeling of relief after finishing a long rally on the winning side.

Conversely, negative valence is evident when a player makes an unforced error and loses a point or doesn't hit the serve inside the borders on the opponent's side.

Evaluating the arousal of the tennis players during the matches can sometimes be straightforward, such as when the players smash their rackets indicating high stress and anger. On the other hand, in situations it may be more complex, involving analyzing subtle facial features. Since tennis is a specific sport, in which some moments in the match are more crucial than others, such as breakpoints, set points, or tie breaks, during those moments high arousal is more emphasized since the pressure on the players is higher. In such instances, it's highly important that the players keep their emotions under control and not succumb to the stress. This ability to manage emotions effectively under high-pressure situations can often distinguish successful athletes from their peers.

Monitoring and analyzing players' emotional states can help them identify situations that impact their emotions and highlight areas for focused mental training. Emotion recognition could also provide real-time feedback during the matches, aiding players in managing their emotions to maintain optimal performance.

In this paper, Ekman's Basic Emotions [14] model was utilized to facilitate the recognition and analysis of tennis players' emotions through their facial expressions during matches. This model was deemed the most appropriate since it identifies fundamental emotions, shared among all individuals, so the emotions are more recognizable across diverse players,

irrespective of the cultural backgrounds. Moreover, it offers a collection of ordinary tell-tale signs for each emotional state, simplifying the labeling process. The model was implemented with an extended set of emotions containing also a neutral state. All these features make Ekman's comprehensive approach particularly useful in the context of FER due to its simplicity and robustness, and by focusing on this set of basic emotions, machine learning systems can more efficiently and accurately classify facial expressions, either by training new models from scratch or by using already trained models in combination with transfer learning.

3. FER

Facial Emotion Recognition (FER), a branch of the computer vision field, is nowadays rapidly gaining attention as advancements in machine learning and computer vision make it increasingly accurate and versatile. To name a few areas of interest for FER and how its applications:

- **Gaming industry** - In recent years, facial emotion recognition has seen a significant interest in the gaming industry, increasingly being integrated into the gaming industry to enhance player experience and engagement. By analyzing players' facial expressions in real time, games can dynamically adjust difficulty levels, storylines, and interactions based on the player's emotional state, creating a more immersive and personalized gaming experience. FER can also be used in multiplayer settings to improve communication and collaboration, as avatars can mimic players' real emotions [15], [16].
- **Healthcare** - FER has been extensively studied for its usefulness in the healthcare industry, potentially used to assess and track the emotional states of patients with mental health conditions, such as depression, anxiety, or autism, providing healthcare professionals with valuable insights into their emotional well-being over time or for early recognizing the signs of mental disorders. Emotion recognition can also assist in pain assessment, particularly for patients who may be unable to communicate their discomfort verbally, allowing for more accurate and empathetic care [17].
- **Robotics** - Another important industry that utilizes FER is robotics [18], particularly for enhancing human-robot interaction. By integrating FER, robots can detect and respond to human emotions, making interactions more intuitive and empathetic. This capability is crucial for in-service robots, where understanding and reacting to user emotions can improve customer satisfaction and efficiency. In healthcare, social robots equipped with FER can provide emotional support and companionship to patients, particularly the elderly or those with cognitive impairments, fostering a more personalized and responsive caregiving environment [19], [20].

[21] also mentions the military sector, automotive industry, marketing, and education as grounds for facial emotion recognition implementation. There are many benefits to FER, and new uses for it in the real world are discovered every day.

3.1. Datasets

Datasets in Facial Emotion Recognition play a key role in training and evaluating models, and their evolution reflects the evolution of the field. These datasets can be categorized based on several criteria, including the type of data they contain - FER can be applied to both static datasets containing single images or datasets containing dynamic sequences of images, each category requiring different approaches to the problem. Image datasets are useful for training models to recognize emotions from still images, but video datasets offer a much larger potential for application in real life, being crucial for understanding the temporal aspects of emotions, such as the transition from a neutral expression to a smile or frown. As FER technology evolves, a third category is becoming more and more important - multimodal datasets, which combine images, videos, and other data types such as audio or physiological signals, offering a more comprehensive approach to emotion recognition. Another way to classify FER datasets is the environment in which the data was captured (controlled or *in-the-wild*), and the range of emotions they cover. Initially, FER datasets were predominantly composed of images captured in controlled environments, where participants were asked to display specific emotions. The Japanese Female Facial Expression (JAFFE) [22] dataset and the Extended Cohn-Kanade (CK+) [23] dataset are prime examples. These datasets typically feature primary emotions as described in the Emotions paragraph, such as happiness, sadness, anger, and surprise, captured under consistent lighting conditions with frontal facial poses. While these datasets were instrumental in early FER research, their limited variability made it difficult for models trained on them to generalize well to real-world scenarios. Consequently, as the demand for more robust and generalizable FER models grew, the focus shifted toward datasets collected in unconstrained environments, often referred to as “in-the-wild” datasets. These datasets, such as the AffectNet [24] and FER2013 [25], contain images and videos of people displaying a wide range of spontaneous emotions in natural settings, with varying lighting conditions, backgrounds, and head poses. The diversity and complexity of these datasets have significantly contributed to the development of more accurate and resilient FER models.

Since this paper focuses on "in-the-wild" application, the most important datasets from that category and their features are described below, and summarized in Table 3.1:

- **AffectNet** - one of the largest and most comprehensive databases used in FER, was created to facilitate research in the field of affective computing by [24]. It contains

more than 1 million facial images collected from the internet using multiple search engines, with each image annotated with one of eight basic emotions: neutral, happy, sad, surprise, fear, disgust, anger, and contempt. Additionally, AffectNet provides information on the intensity of the emotions (valence and arousal), so the database covers the categorical model as well as the continuous dimensional model. An interesting feature of the AffectNet database is that about half of the images have been manually annotated, and the other have been annotated using neural networks. The images vary in terms of pose, lighting, and occlusions, making AffectNet a valuable resource for training deep learning models that need to generalize well to real-world conditions. This dataset has become a benchmark for evaluating FER models, helping to advance the field by providing a challenging and diverse set of data.

- **RAF-DB** - The Real-world Affective Faces Database (RAF-DB) [26] is a widely used dataset designed to capture real-world variability in facial expressions. It contains approximately 30,000 facial images collected from the internet, with annotations for Ekmans six basic emotions [14], and a neutral facial expression. The dataset is particularly valuable for its diversity, including variations in age, gender, ethnicity, and lighting conditions. This dataset is frequently used to train and benchmark the FER model aimed at recognizing emotions in uncontrolled, real-world scenarios.
- **FER-2013** - FER-2013 is a widely recognized facial emotion recognition dataset that was introduced during the ICML (International Conference on Machine Learning) 2013 [25]. It contains roughly 35,000 grayscale images, each sized at 48x48 pixels, and labeled with one of seven basic emotions: anger, disgust, fear, happiness, sadness, surprise, and neutral. The dataset was collected from the internet, making it diverse and challenging, with various lighting conditions, poses, and occlusions, and labeled by crowdsourcing. FER-2013's simplicity in terms of image size and the balanced distribution of emotions make it a valuable resource for developing and testing algorithms that need to work in real-world, unconstrained environments.
- **FER+** - After some criticism of the FER-2013 database about mislabelled images [21], the FER+ database [27] has arisen as an enhanced version. In FER+, each image was relabeled using a more comprehensive set of eight emotions: neutral, happiness, surprise, sadness, anger, disgust, fear, and contempt. This relabeling process significantly improved the quality and accuracy of the annotations.

- **AFEW** - The AFEW (Acted Facial Expressions in the Wild) [28] database is a well-known dataset designed for facial emotion recognition in videos, particularly for emotions captured in real-world scenarios. It contains video clips extracted from 54 movies, making it one of the first datasets to capture naturalistic emotions in dynamic contexts. The dataset includes various expressions and emotional states, categorized into the same seven classes as FER-2013: anger, disgust, fear, happiness, sadness, surprise, and neutral. AFEW is widely used in the EmotiW (Emotion Recognition in the Wild) challenge, making it a critical benchmark for evaluating FER systems that need to handle real-world variability [29].
- **SFEW** - The Static Facial Expressions in the Wild (SFEW) [28] database is an extension of the AFEW dataset, designed specifically for static images. It consists of images extracted from the AFEW video sequences, offering a more focused set of facial expressions captured in challenging, real-world conditions. SFEW includes the same seven emotion categories as AFEW. Two versions of SFEW exist - SFEW 1.0 and 2.0, with the latter featuring improved image quality and more consistent labeling. SFEW serves as a valuable resource for training and testing facial emotion recognition models in static images, particularly in scenarios that involve complex backgrounds, varying lighting conditions, and naturalistic expressions.
- **Aff-Wild/Aff-Wild2b** - The Aff-Wild (Affect-in-the-Wild) [30] database is one of the most extensive and challenging datasets for facial emotion recognition in the wild. It consists of thousands of video clips extracted from a variety of real-world scenarios, such as YouTube videos, which capture a wide range of spontaneous facial expressions. The dataset includes annotations for both valence and arousal, making it particularly useful for studying continuous emotion prediction. A couple of years later, Aff-Wild2 was introduced in [31] as an advanced extension of the original Aff-Wild database representing one of the most comprehensive benchmarks for facial emotion recognition in the wild. It contains around 2.8 million frames from nearly 600 videos, capturing spontaneous facial expressions in various real-world conditions.

Table 3.1 Popular FER datasets

Dataset	Size	Emotions	Type of content

AffectNet	~1,000,000	Neutral, Happy, Sad, Surprise, Fear, Disgust, Anger, and Contempt + Valence and Arousal	Static (images)
RAF-DB	~30,000	Neutral, Happy, Sad, Surprise, Fear, Disgust, and Anger	Static (images)
FER-2013	~35,000	Neutral, Happy, Sad, Surprise, Fear, Disgust, and Anger	Static (images)
FER+	~35,000	Neutral, Happy, Sad, Surprise, Fear, Disgust, Anger, and Contempt	Static (images)
AFEW	~600	Neutral, Happy, Sad, Surprise, Fear, Disgust, and Anger	Dynamic (videos)
SFEW	~1,800	Neutral, Happy, Sad, Surprise, Fear, Disgust, and Anger	Static (images)
Aff-Wild	~1,180,000	Valence and Arousal + facial landmarks	Static (images)
Aff-Wild2	~2,500,000	Valence and Arousal + facial landmarks	Static (images)

In summary, FER datasets have evolved from small, controlled collections to large, diverse datasets that better represent the complexity of real-world emotions which has been crucial in advancing the accuracy and applicability of FER technologies across various domains.

3.2. Data preprocessing

The preprocessing of the video data is a crucial step in machine learning that involves preparing raw video footage for model analysis. It typically involves stages such as frame extraction, resizing, normalization, and data augmentation. In the next paragraphs, the steps will be explained in detail.

3.2.1. Frame Extraction

The first step in video preprocessing is frame extraction. Frame extraction implies breaking down the original video into a series of individual frames, which are treated as images. This step is usually implemented using libraries such as OpenCV [32] in Python, and it's essential

because most machine learning models for video analysis, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), are designed to process images. By treating each frame as an individual image, the temporal dynamics of the video can be captured and analyzed frame by frame, while it's also possible to isolate key moments within videos and focus on specific instances that are relevant to the task at hand.

3.2.2. Resizing

After the frames have been extracted, each should be resized to a consistent dimension. In this context, resizing refers to adjusting the resolution of the images to a uniform size, such as 224x224 pixels as is in this paper. The cropping is, the same as frame extraction, usually implemented using implementations from different libraries, for example, OpenCV [32] or TensorFlow [33] are commonly used for resizing operations in Python. Resizing is usually implemented using interpolation methods, which try to achieve the best approximation of the pixel values at the new size based on the surrounding pixels from the original image. The most common interpolation techniques include:

- **Nearest Neighbor Interpolation** - this is the simplest and fastest method of image scaling [34]. It assigns to each pixel in the scaled image the value of the nearest pixel from the original image, i.e. it involves rounding the coordinates of the target pixel to the nearest integer coordinates of the original image
- **Bilinear Interpolation** - Bilinear interpolation considers the closest 2x2 neighborhood of known pixels surrounding the target pixel in the original image. The value of the target pixel is then computed as a weighted average of these four surrounding pixels, considering vertical and horizontal directions (Figure 3.1). This is the technique used in the used in this paper by the OpenCV library [32].

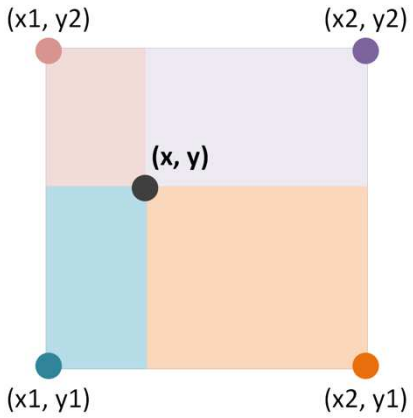

Figure 3.1 Visualization of Bilinear Interpolation

- **Bicubic interpolation** – it extends the bilinear method by considering the 4x4 (16) neighboring pixels of the target pixel. It uses cubic polynomials to calculate the interpolation, which gives a more accurate and smoother result than bilinear interpolation

The necessity of resizing lies in ensuring that all frames have the same spatial dimensions, which is critical for batch processing during model training. Without resizing, the model would have to handle images of varying sizes, leading to inefficiencies and potential errors.

3.2.3. Normalization

Normalization is a common term used in machine learning in the context of data preprocessing, and it implies scaling the data features to a standard range, typically between 0 and 1 or -1 and 1. This ensures that all features contribute equally to the model training process, preventing any single feature from disproportionately influencing the model due to larger numerical values [35]. In the context of video preprocessing for facial emotion recognition (FER), normalization is applied to the pixel values of video frames, meaning that each pixel's intensity value, originally ranging from 0 to 255, is scaled to a normalized range, such as $[0,1]$. This scaling is achieved by dividing the pixel values by 255:

$$Normalized\ value\ = \frac{Pixel\ value}{255}$$

Normalization is applied uniformly across all frames in the video and the normalized data is then fed into the model, where each pixel value is treated as an individual feature, contributing to the overall representation of the facial expressions.

3.2.4. Data Augmentation

Data augmentation is essential in machine learning, particularly for tasks that include image and video data. It involves applying various transformations to the original dataset to create additional training examples, thereby improving the model's generalization capabilities. [36] broadly categorizes data augmentation into two types: data transformation methods and data synthesis methods. Simple geometric transformations include operations like flipping, rotating, scaling, and cropping and can be applied to the whole image or just a part. Transformations can also be photometric, meaning they change specific pixels' nature without changing their spatial features [36]. Common photometric augmentation methods include the **adjustment of brightness**, where the overall intensity of the image is modified to simulate different lighting conditions, **contrast adjustment**, which alters the difference between the light and dark areas of an image to make certain features more prominent or subdued, or **saturation adjustment** which modifies the intensity of colors in the image, among many others. One more interesting photometric data augmentation technique is **Gaussian noise**. It simulates imperfections like camera noise by adding a random value to the value of each pixel. The added values are drawn from a Gaussian distribution with a mean of zero and a specified standard deviation which means that most pixel values have small deviations from the original, but some can have larger variations. The Gaussian distribution is defined as:

$$\mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

where μ is the mean and σ^2 is the variance of the distribution. When adding Gaussian noise, the pixel intensity at position (x, y) is modified as follows:

$$I_{noisy}(x, y) = I(x, y) + \mathcal{N}(0, \sigma^2)$$

Here, $I(x, y)$ represents the original pixel intensity, and $\mathcal{N}(0, \sigma^2)$ is the Gaussian noise with zero mean and variance σ^2. This technique is particularly useful in tasks like facial emotion recognition, where it helps the model become more resilient to variations in facial features caused by different lighting or slight movements. The application of Gaussian noise can be seen in Figure 3.2.

Figure 3.2 Image before and after applying Gaussian noise

This paper mainly focused on data augmentation using the previously listed methods and techniques which all belong to the *Data transformation* augmentation techniques. The second data augmentation category is the *Data synthesis* one [36]. Data synthesis augmentation techniques focus on generating new data instances rather than just modifying existing ones. This comes in handy when a large amount of data is hard to come by.

One common technique is the use of **Generative Adversarial Networks (GANs)**, which consist of two neural networks—the generator network plus the discriminator network—competing against each other. The generator creates new data instances trying to *trick* the discriminator into believing it's the real data, while the discriminator evaluates their

authenticity, gradually improving the quality of the synthetic data until it is nearly indistinguishable from the real data [37]. GANs and other data synthesis techniques not only augment the dataset but also introduce new data distributions that the model may encounter in real-world applications. By incorporating synthetic data into the training process, the model becomes more resilient to variations and less prone to overfitting [37].

3.2.5. Temporal Normalization

Another important step in video preprocessing is temporal normalization. The purpose of temporal normalization is to uniform the number of frames across all video samples, ensuring consistency in temporal length despite variations in the original videos. If a video is too long and has too many frames, trimming is applied, conversely, if the video is too short and has fewer frames than desired, then it's padded. Trimming involves reducing longer videos by discarding excess frames, often selecting frames at regular intervals to maintain the video's temporal structure. Padding, on the other hand, extends shorter videos by repeating frames or adding blank frames until they match the target length. Temporal normalization can also involve techniques like interpolation, where additional frames are generated between existing ones to create a smooth sequence that preserves the video's temporal coherence while adjusting its length.

Trimming

If the original video has 15 frames: a1, a2, a3, ..., a15 and the target number of frames is 10 (a10), trimming involves selecting evenly spaced frames from the original sequence:

Original frames:

a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15

Trimmed to 10 frames:

a1	a2	a4	a5	a7	a8	a10	a11	a13	a14

In this sequence, every third frame is removed, and the resulting sequence retains only the selected frames to match the target length.

Padding

If the original video has 6 frames: a1, a2, a3, a4, a5, a6 and the target is 10 frames (a10), padding involves repeating some frames uniformly or adding extra frames to reach the target:

Original frames:

a1	a2	a3	a4	a5	a6

Padded to 10 frames:

a1	a2	a2	a3	a3	a4	a4	a5	a5	a6

Here, frames are repeated in sequence to fill the remaining slots, ensuring the total number of frames meets the target.

3.2.6. Face Detection and Cropping

A step that's closely related to video preprocessing in FER and other computer vision tasks involving human faces is **face detection**. The importance of face detection lies in its ability to isolate the region of interest (ROI) in an image or video, which is crucial for accurately analyzing facial features and expressions. Without proper face detection, an algorithm might be analyzing irrelevant parts of an image, leading to poor performance and unreliable results. Over the years, various face detection algorithms have been developed, evolving from simple methods to more sophisticated, deep-learning-based approaches. One of the most famous is the Viola-Jones detector, introduced in 2001 in [38], this algorithm uses Haar-like features (Figure 3.3), the AdaBoost algorithm [39], and a cascade of classifiers to quickly and accurately detect faces in images. Although it was revolutionary at the time and is still used today by many, it has limitations in terms of accuracy, especially under varying lighting conditions and facial orientations.

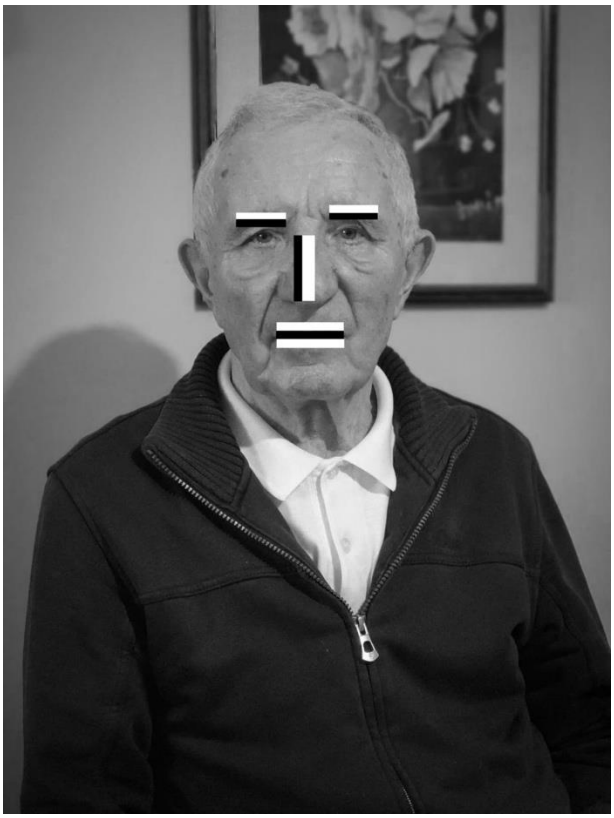

Figure 3.3 Haar-like features

As the field of computer vision progressed, more advanced methods were introduced. Histogram of Oriented Gradients (HOG) [40] combined with Support Vector Machines (SVMs) [41] became popular due to its robustness in detecting faces across different orientations and lighting conditions. HOG works by counting occurrences of gradient orientation in localized portions of an image, providing a feature set that can be used for detecting faces [42]. The HOG features are then fed into an SVM, a powerful classifier that finds the hyperplane that best separates the face from non-face examples in a high-dimensional space. The SVM is trained on labeled data to distinguish between positive (face) and negative (non-face) samples, thereby enabling accurate face detection.

The introduction of deep learning further revolutionized face detection with models like **Multi-task Cascaded Convolutional Networks (MTCNN)** [43], a model used in this paper. MTCNN is a deep learning-based approach that simultaneously detects faces and facial landmarks with high accuracy by utilizing three stages of cascaded convolutional networks that progressively refine the detection results, each with specific tasks.

The first stage, known as the *Proposal Network (P-Net)*, generates candidate windows that may contain faces. This stage is designed to quickly scan an image at multiple scales, producing a large number of potential face locations.

The second stage, the *Refine Network (R-Net)*, further processes these candidate windows, refining the bounding boxes and discarding false positives. The R-Net improves the accuracy of face detection by applying additional convolutional and fully connected layers, which provide more complex and refined features for classification and bounding box regression.

The third and final stage, the *Output Network (O-Net)*, is responsible for finalizing the face detections and predicting facial landmarks. This stage refines the bounding boxes further and outputs precise facial landmarks that are crucial for tasks like face alignment.

MTCNN's multi-task learning approach, where the networks simultaneously learn to detect faces and localize landmarks, leads to superior performance in both tasks. However, like any algorithm, it also has its downsides. One of the primary limitations is its computational cost, especially during the multi-scale detection phase, where the model processes the image at various scales, making it less suitable for resource-constrained environments.

Once a face is detected, the next step is cropping, where the detected face bounding box is extracted from the original image or video frame. Cropping is essential because it isolates the face from the rest of the image, allowing the model to focus solely on the facial features that are relevant for emotion recognition. In the context of video processing, cropping can also help reduce computational complexity by eliminating background noise and irrelevant parts of the frame.

3.3. Architectures

FER architectures employed to accurately detect and classify human emotions based on facial expressions evolved through time, parallel with other technological innovations.

Traditional approaches involve handcrafted features, such as the **Histogram of Oriented Gradients (HOG)** [40] and **Local Binary Patterns (LBP)** [44], which are then fed into classifiers like **Support Vector Machines (SVMs)** [41] or **k-Nearest Neighbors (k-NN)** [45]. While effective in certain scenarios, these methods had limitations in handling complex variations in facial expressions, lighting, and occlusions.

With the advancement of deep learning, more sophisticated architectures have been developed, with **Convolutional Neural Networks (CNNs)** [46], among the most popular, leveraging their ability to learn spatial hierarchies of features from the data automatically. From the CNNs, **3D Convolutional Neural Networks (3D CNNs)** [47] have been developed to handle dynamic sequences of facial expressions in videos. Additionally, **Recurrent Neural Networks (RNNs)** [48], particularly **Long Short-Term Memory (LSTM)** [49] networks, were employed to also capture temporal dependencies in video data, further improving the recognition of emotions in continuous sequences. Today, state-of-the-art methods include **Transformers** [50] with their self-attention mechanisms that capture global relationships between facial features and **Graph Neural Networks (GNNs)** [51] that model complex relationships between facial landmarks. In the next paragraph, the most used modern architectures will be explained in detail, and reputable examples will be listed for each.

3.3.1. Convolutional Neural Networks (CNNs)

Convolutional neural Networks (CNNs) have revolutionized the field of computer vision since their introduction in the 1980s in the paper [46] where the first CNN, known as LeNet, for handwritten digit recognition was introduced. This model demonstrated the potential of CNNs in using convolutional layers to learn spatial hierarchy in images. However, it wasn't until the 2012 breakthrough with the AlexNet model [52] which won the ImageNet competition that year by a large margin, that CNNs gained widespread attention. This moment marked the beginning of the deep learning revolution in computer vision, paving the way for more sophisticated CNN architectures [53], [54], [30].

The convolutional layer is a fundamental building block in every CNN, that processes the input data by applying a set of learnable filters, also known as kernels. Each filter is a small matrix of weights that slides over the input data, performing a mathematical operation called convolution. This operation involves element-wise multiplication between the filter and the corresponding section of the input, followed by summing the results to produce a single value which then forms a point in the output feature map. The goal of the convolution operation is to detect specific patterns in the data. For instance, in an image, early convolutional layers might detect simple features like edges or textures by using filters that highlight changes in pixel intensity. As the data passes through deeper layers, the filters

become more complex, capable of detecting higher-level features such as shapes, objects, or even facial expressions [55].

Mathematically, for an input image I and a filter K, the convolution operation to produce the output feature map O can be expressed as:

$$O(i,j) = \sum_m \sum_n I(i+m, j+n) \cdot K(m,n)$$

Here, i and j denote the position of the filter on the input image, and m and n are the dimensions of the filter. Once the feature map is generated, it is passed through a non-linear activation function, typically ReLU (Rectified Linear Unit). ReLU is defined as $\mathrm{f(x)} = \max(0, \mathrm{x})$, where any negative values in the feature map are set to zero, while positive values remain unchanged which introduces non-linearity into the model, allowing it to learn more complex representations.

The second fundamental concept in CNNs is the *Pooling layer*. Pooling layers are used to reduce the spatial dimensions of the feature maps. To be precise, the primary purpose of pooling is to down-sample the input representation, reducing its dimensionality and enabling the network to focus on the most critical features while discarding less important information. This process also helps to decrease the computational load and mitigate the risk of overfitting [35]. It is typically done using either max pooling or average pooling. In max pooling, a defined window matrix slides over the input feature map, and within each window, the maximum value is selected and retained in the pooled feature map. This operation effectively retains the most prominent features detected by the convolutional layers while discarding the rest. Average pooling works similarly but instead of taking the maximum value, it computes the average of the values within the window.

For example, if we have a 2x2 window and the values within that window are $\big[[1,3],[2,4]\big]$, max pooling will output 4, while average pooling will output 2.5. As the pooling operation is applied across the entire feature map, the output becomes a smaller, more condensed version of the input. Both pooling types are depicted in Figure 3.4.

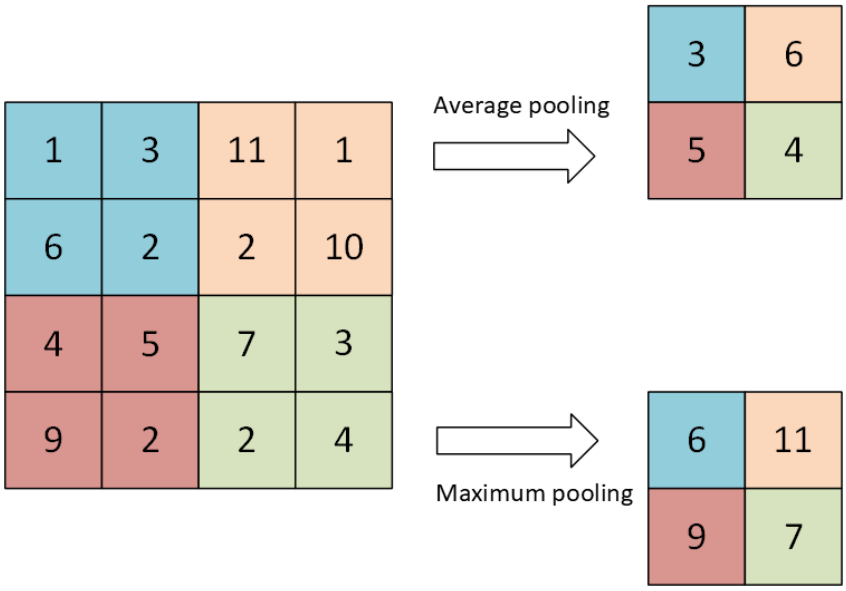

Figure 3.4 Pooling examples

After a series of convolutional and pooling layers, the feature maps are usually flattened, meaning that they are converted into a 1D vector which is then passed through one or more fully connected layers, where each neuron is connected to every neuron in the previous layer. These layers act as a high-level reasoning engine, combining the features extracted by the convolutional layers to make a final prediction.

The output layer of a CNN usually involves applying an activation function, such as softmax or sigmoid, depending on the task. Figure 3.5 depicts an example of a CNN architecture.

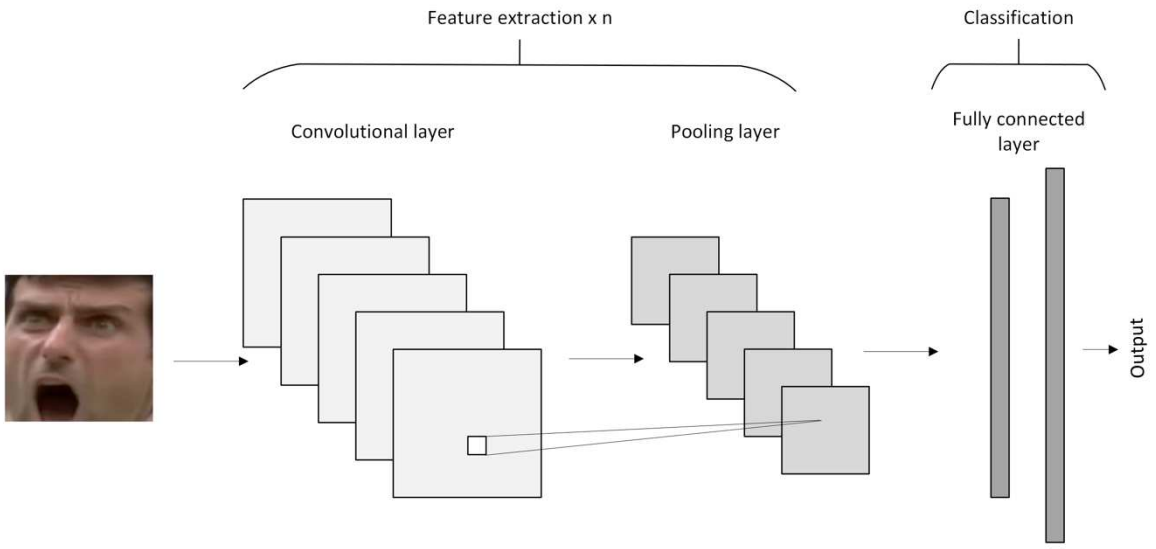

Figure 3.5 Convolutional Neural Network architecture example

There are many types of CNNs, but this paper mainly focuses on 2D and 3D CNNs.

2D CNNs are the most common type and are used primarily for analyzing two-dimensional images. In 2D CNNs, both the convolutional filters and the input data are two-dimensional. This makes 2D CNNs particularly well-suited for tasks like image recognition, object detection, and image segmentation. However, for tasks that involve temporal data, such as video analysis or FER from video sequences, 3D CNNs are more appropriate. 3D CNNs extend the 2D convolution operation by adding a third dimension, allowing them to process volumetric data or video frames. Consequently, this makes 3D CNNs particularly useful for video classification, action recognition, and 3D medical imaging. For FER, 3D CNNs can analyze how facial expressions change over time, providing a more comprehensive understanding of emotional states than 2D CNNs, which only capture spatial features.

In the context of Facial Emotion Recognition (FER), CNNs have been widely used due to their ability to automatically learn and extract relevant facial features from images or video frames. 2D CNNs are typically applied to static images or individual video frames to classify emotions based on facial expressions. Notable models for 2D CNNs in FER include VGG [55], ResNet [56], and MobileNet [57].

On the other hand, 3D CNNs are applied to video sequences, where they can capture the temporal evolution of facial expressions. Popular 3D CNN architectures for FER include MoViNets [58] and I3D (Inflated 3D) [59], which have successfully learned spatial and temporal features from video data. These models are particularly effective in capturing subtle emotional transitions, which is crucial for accurate emotion recognition in dynamic settings.

Next, some popular CNN models are described in more detail.

- **ResNet -** ResNet, short for Residual Network, is a groundbreaking deep learning architecture introduced in [56]. The key innovation in ResNet is the introduction of residual learning, which addresses the problem of vanishing gradients, which occurs in deep neural networks when gradients used for updating weights during training diminish as they are backpropagated through the layers, leading to slow or stalled training. This problem is particularly pronounced in networks with many layers, making it difficult to train very deep models. ResNet overcomes this by using *skip connections*, where the output of a layer is directly added to the output of a deeper layer. These kinds of connections between layers form *residual blocks*. Residual blocks allow the network to maintain strong gradient signals even in very deep architectures, enabling the successful training of models with more than a hundred layers, such as ResNet-101 and ResNet-152.

- **VGG -** VGG (Visual Geometry Group) is a convolutional neural network architecture introduced by the Visual Geometry Group at the University of Oxford and presented in [55] paper in 2015. It is known for its simple and uniform architecture, which is characterized by using small (3x3) convolutional filters stacked on top of each other, with increasing depth, followed by max-pooling layers to reduce spatial dimensions. The network's depth varies depending on the version, such as VGG16 and VGG19, which have 16 and 19 layers, respectively. This architecture tackled the problem of capturing complex features by using multiple small filters in sequence. For instance, using three 3x3 convolutional layers in a row has the same receptive field as a single 7x7 layer but with fewer parameters and more non-linearities, allowing for deeper and more effective feature extraction. The VGG models performed exceptionally well in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2014, significantly contributing to advancements in the field of computer vision. However, the main drawback of the VGG architecture is its computational cost. The deep network with many parameters requires significant memory and computational power, making it less efficient for deployment in resource-constrained environments.

- **I3D -** Inflated 3D ConvNet (I3D) is a neural network architecture specifically designed for video recognition tasks. Introduced by Joao Carreira and Andrew Zisserman in their paper [59], I3D extends the standard 2D convolutional networks into the spatiotemporal domain by *inflating* 2D convolutional filters into 3D. The core idea behind I3D is to take advantage of the successful architectures in image classification and adapt them for video analysis by adding a temporal component. This means that 2D filters used initially in image recognition models can be expanded to operate on spatial dimensions and the temporal dimension, making them suitable for processing video data. I3D has demonstrated strong performance in video classification and action recognition tasks, significantly improving accuracy on benchmarks like the *Kinetics* dataset. Its ability to leverage pre-trained 2D models and its effectiveness in capturing spatiotemporal features have made it a popular choice for video-related deep-learning tasks.

- **MoViNet** - MoViNets, or Mobile Video Networks [58] are a family of efficient and scalable neural networks specifically designed for video classification tasks. Introduced by Google Research, MoViNets are optimized for mobile and edge devices where computational resources are limited. This architecture integrates temporal information from video sequences, making it highly effective for understanding dynamic content, while leveraging efficient network designs, such as depthwise separable convolutions and temporal convolutions, to balance accuracy with computational efficiency.

3.3.2. Transformers

Transformers are a fairly new technology, introduced in a revolutionary paper *Attention Is All You Need* [50] in 2017. This architecture marked a significant departure from previous models like Recurrent Neural Networks (RNNs) [48] and Long Short-Term Memory (LSTM) networks [49], which relied heavily on sequential processing. Transformers, on the other hand, utilize self-attention mechanisms to process input data in parallel, enabling much more efficient training and improved handling of long-range dependencies in sequences.

The main idea behind transformers is the *attention* mechanism, which allows the model to weigh the importance of different parts of the input data when making predictions. In a

transformer architecture, the input is processed through an encoder-decoder structure, where both the encoder and decoder are composed of layers of self-attention and feedforward neural networks.

The encoder's role is to take the input sequence, apply self-attention to understand the relationships between all elements and produce a set of continuous representations known as the context. Each encoder layer consists of a self-attention mechanism followed by a feedforward neural network, both of which are equipped with layer normalization and residual connections to stabilize and optimize training. The self-attention mechanism works so that the model looks at an input sequence and computes a score for each element in the sequence relative to every other element which reflects how important each element is to understand the input sequence, e.g. some words in sentences are more crucial to the meaning than others. This process enables the model to focus on the most relevant parts of the data when making predictions.

Once the encoder has processed the input, the decoder takes over. The decoder's primary function is to generate the output sequence, one element at a time, using the information provided by the encoder. Like the encoder, each decoder layer also includes self-attention and feedforward neural networks. However, the decoder has an additional step: *cross-attention*. Cross-attention allows the decoder to focus on the specific parts of the encoder's output, ensuring that the decoder's predictions are informed by the entire input sequence.

Moreover, transformers introduced the concept of multi-head attention, which allows multiple self-attention mechanisms to be applied in parallel, capturing various aspects of the data simultaneously. The transformer architecture is illustrated in Figure 3.6.

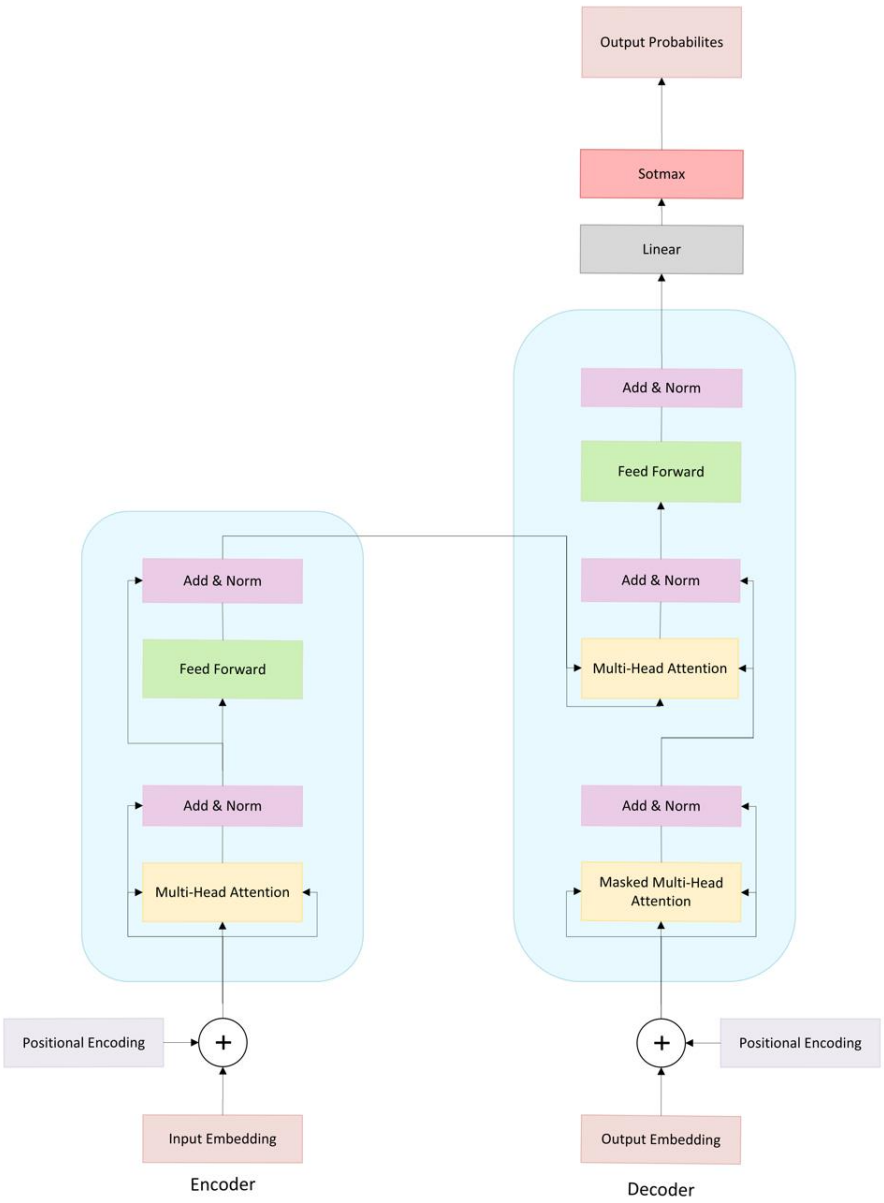

Figure 3.6 Transformer model architecture (adapted from [50])

Transformers have increasingly been adapted for use in computer vision tasks, including Facial Emotion Recognition (FER). The self-attention mechanism allows the model to weigh the importance of different image regions, capturing both global and local features effectively. This capability is particularly advantageous in FER, where subtle and nuanced facial expressions across various regions of the face need to be recognized and interpreted accurately.

In [60] the authors proposed a Vision Transformer (ViT) architecture for image classification tasks on common datasets such as ImageNet, CIFAR-100, VTAB, and others. ViT showed remarkable benchmarking results, accumulated with lower computational cost.

[61] put to test 13 different ViT models from the Tokens-to-Token ViT, and Mobile ViT to the Deep ViT, achieving the highest accuracy on the FER2013 dataset with the Tokens-to-Token ViT which was 61.28%.

Another paper that used ViT architecture based on the original model was [62], in which the authors achieved an accuracy of 87%.

These results show transformers are emerging as a powerful tool in advancing the accuracy and robustness of FER systems, especially in complex and dynamic settings.

3.3.3. Graph Neural Networks

Graph Neural Networks (GNNs) were introduced in 2009 in [51]. They proposed an architecture of neural networks that could work directly with graph-structured data consisting of nodes (representing entities) and edges (representing relationships between entities). This new kind of neural network has opened new possibilities for understanding complex relationships in data that can be naturally represented as graphs, such as social networks, and molecular structures [51]. However, GNNs have quickly expanded into various domains beyond just social networks and chemistry - they are utilized in natural language processing, where sentences can be represented as graphs of words; in computer vision, where the relationships between different parts of an image can be modeled as a graph; and in recommendation systems, where they help in understanding user-item interactions more deeply [63].

The core logic behind GNNs is their ability to model data in a non-Euclidean space, where data points are represented as nodes and the connections between them as edges. This structuring allows GNNs to capture the interdependencies and relationships between different nodes in a way that traditional neural networks cannot.

In a representative GNN, each node has a state which is a vector representing its features. The nodes' states are updated during an iterative process, based on the states of their

neighboring nodes and the nature of the edges connecting them. This process allows each node to gather information from its neighbors, effectively enabling the network to learn how the nodes influence each other. Over several iterations, this exchange of information continues until the states of the nodes stabilize, capturing a comprehensive representation of the entire graph structure. The training is similar to the traditional neural network architectures, where a loss function is minimized to adjust the parameters of the model.

Despite their effectiveness, GNNs are not without limitations. One of the main challenges is the difficulty in capturing long-range dependencies within large graphs, where the influence of distant nodes on each other may diminish over several iterations. Additionally, as the size of the graph increases, the computational demands grow, making it challenging to apply GNNs to extremely large datasets. Another downside to GNNs is that they are *"black boxes"* which means they are hard to interpret.

Graph Neural Networks (GNNs) have shown significant promise in computer vision by enabling the modeling of complex relational structures in visual data. While the traditional Convolutional Neural Networks (CNNs) are limited to grid-like data, such as images, where spatial locality is key, GNNs offer an understanding beyond simple pixel relationships, where entities and their interactions form complex structures better represented as graphs. To handle different real-world problems, different GNN architectures have been implemented such as Graph Convolutional Networks (GCNs) [64], Graph Attention Networks (GATs) [65], Spatial-Temporal Graph Convolutional Networks (ST-GCNs) [66], and Graph Recurrent Networks (GRNs) [67].

For the specific task of FER, Graph Convolutional Networks (GCNs) are usually used [68]. They adapt the convolutional operations used in traditional CNNs to work on graphs, enabling the extraction of features that capture the structural relationships between nodes. The process of training GCN begins with each node in the graph having an initial feature representation, known as an *embedding*. These node representations are processed through a feedforward network to generate messages that can be shared with neighboring nodes. The next step is aggregation, where each node gathers messages from its neighbors, taking into account the weights of the edges connecting them. This aggregation typically involves calculating a weighted sum or mean of the neighboring nodes' messages. Finally, the aggregated information is used to update the node's feature representation. This update is performed by processing the aggregated messages through another feedforward network, which combines the new information with the node's original state, resulting in a new,

refined feature representation. This sequence of steps allows GCNs to iteratively refine node representations across multiple layers, capturing both local and global patterns within the graph.

For tasks with sequences of images, Spatial-Temporal Graph Convolutional Networks (ST-GCNs) [66] are used to handle data that varies over both spatial and temporal dimensions. ST-GCNs are particularly useful for tasks like action recognition in videos, where it's crucial to capture not only the spatial relationships between different parts of a scene or object but also how these relationships evolve. ST-GCNs achieve this by integrating graph convolutions, which capture spatial dependencies, with temporal convolutions, which model the changes over time, allowing the network to learn complex spatiotemporal patterns within the data.

3.4. FER in Sports

FER (Facial Emotion Recognition) in sports is an emerging field that holds significant potential for understanding athletes' emotional and psychological states during performance. However, it remains underexplored, with relatively few studies addressing it directly. This aspect of FER falls under the broader category of "FER in-the-wild," which is gaining traction as researchers seek to develop systems that can accurately detect and interpret emotions in real-world, unconstrained settings. "FER in-the-wild" refers to the application of FER techniques in environments where factors such as lighting, background noise, and spontaneous, natural expressions introduce complexities that are not present in controlled settings. This approach is particularly relevant in sports, where athletes' emotional expressions are influenced by a variety of factors including competition pressure, physical exertion, and the unpredictability of game situations. Addressing these challenges requires robust models capable of generalizing across different scenarios and capturing subtle emotional cues.

The only notable study found as a subject of exploration for this paper that deals with the subject of FER in tennis specifically was [69]. This paper stands out as it moves away from traditional actor-based datasets and instead uses real-life tennis match footage, making the findings more relevant to actual sports scenarios. The researchers employed CNNs to analyze the expressive behavior of tennis players, achieving a notable accuracy rate of 68.9% in identifying affective states. The architecture of the CNNs was composed of two primary

networks - one for analyzing the player's body position, and the other that operated directly with the image data, with data sequences of 64 frames in the case of the highest accuracy. The networks were trained to recognize the valence of the players, i.e. if they have won or lost the point. The study's findings suggested that CNNs can match or even exceed human observers' capabilities - which had accuracy between 55.9% and 63.0% on the same dataset, in recognizing these emotional states, making this approach valuable for enhancing performance analytics in tennis.

Another significant study, [70], further expands on the application of FER in sports, but in ultra-distance runners. This study employed three different models Deepface, ResMaskNet, and SVM, all having consistent results recognizing fatigue as the most common emotion.

[71] utilized the SVM classifier combined with the KNN classification to achieve the highest recognition accuracy rate of 94.2% in recognition of emotional states in basketball players.

In conclusion, while the application of FER in sports, particularly tennis, is a relatively new area of research, it holds great promise. The studies that have been conducted so far demonstrate the potential of this technology to enhance our understanding of athletes' emotional states and to provide tools that can improve performance. As the field continues to evolve, FER will likely become an increasingly important component of sports analytics, offering new ways to optimize both mental and physical aspects of athletic performance.

4. Dataset

4.1. Data Scraping

The data have been accumulated from publicly available sources and official tennis organizations [72]. Per the match videos' availability, the facial emotion recognition task was limited to the male players playing top-level tournaments. To outset the bias, the matches were chosen to have players of multiple races and ages.

4.2. Video Processing

To extract the frames containing displays of the players' emotions, full matches have been thoroughly analyzed frame by frame using a video editing tool, and the corresponding frames have been cut out and saved as separate videos. The videos extracted displayed moments immediately after a point. To make data consistent and to have enough frames to recognize the emotion the videos were limited to be at least 2 seconds long [69]. An Excel sheet was created to contain data corresponding to the extracted videos. Every video is one input example, and its features (columns in the sheet) include

- **Index number** - video ordinal number
- **Match name** - the full name of the match the video was extracted from
- **Player** - player in the video
- **Emotion** - emotion in the video
- **Valence** - intrinsic attractiveness of an emotion, ranging from positive (pleasant) to negative (unpleasant)
- **Arousal** - the intensity of the emotion in the video, ranging from one to three, one being the calmest, and three being the most intensive
- **Won/Lost** - indicates if the player on the videos has just lost or won the point

The described video data is highly suitable for training facial emotion recognition (FER) neural networks and is easy to implement in the training process. Each video, representing a single input example, can be used as a labeled data point where the emotion displayed by the player is the target output for the network to learn. Features such as valence and arousal provide additional labels that can be utilized to train a model not only to recognize specific

emotions but also to predict the intensity and pleasantness of these emotions. By using the labeled data, a neural network can learn to identify patterns in the facial expressions that correlate with different emotional states. The context provided by the "Won/Lost" feature can also enable the network to learn how emotions are influenced by game outcomes, enhancing its ability to make accurate predictions in varied situations. With enough training data, the model can generalize well, allowing it to effectively recognize emotions in new, unseen videos.

It is important to acknowledge that the labeling of these videos was performed by an individual without formal training in psychology or professional expertise in tennis. Consequently, there is a risk of mislabeling, either due to misinterpretation of the players' emotional states or a lack of familiarity with the intricacies of tennis behavior. Additionally, the dataset suffers from an imbalance in the distribution of emotions due to the relatively small number of examples in the dataset. This imbalance can lead to challenges during the training of neural networks, as the models may become biased, achieving higher accuracy for the more frequently represented labels while performing poorly on those that are underrepresented.

The quantity of examples is a direct result of the time-intensive nature of both the data collection and preprocessing stages, coupled with the constraints of a project deadline.

4.3. Data analysis

The analysis of the extracted video data revealed a total of 198 videos, each capturing the emotional expressions of players immediately following a point in top-level tennis matches. The distribution of **emotions** observed in the videos included happiness, anger, sadness, and a neutral state, with the most frequently occurring emotion being anger occurring in 88 videos, accounting for 44 percent of the total videos.

In terms of **valence**, the data showed a predominance of 61% of videos with positive valence (pleasant emotions), while 39% exhibited negative valence (unpleasant emotions).

For **arousal**, the distribution indicated that 65% of the videos fell into the low arousal category (score of one), 19% were moderate (score of two), and 16% were high arousal (score of three). The average arousal score across the dataset was 1,52, suggesting that players tended to exhibit levels of intensity during critical points. This comprehensive emotional analysis offers valuable insights into the psychological dynamics of top-level

tennis players as they navigate the pressures of competition. All the data is summarized in the subsequent tables (**Error! Reference source not found.**, **Error! Reference source not found.**, **Error! Reference source not found.**).

Table 4.1 Emotions distribution

Value	**Quantity**	**Percentage**
Anger	88	44,4%
Sadness	53	26,4%
Neutral	48	24,2%
Happiness	10	5,0%

Table 4.2 Arousal distribution

Value	**Quantity**	**Percentage**
Low	88	44,4%
Mid	53	26,4%
High	48	24,2%

Table 4.3 Valence distribution

Value	**Quantity**	**Percentage**
Positive	77	39%
Negative	121	61%

Moreover, videos were systematically transformed into images to facilitate the training of 2D Convolutional Neural Networks (CNNs) models and a transformer model. This method capitalized on the inherent spatial features within frames, allowing the models to effectively learn and predict from static visual input. The size of the dataset varied significantly with

the number of frames per video, affecting the volume of training, validation, and testing data which is instrumental in accommodating different model capacities and training requirements. The concrete data on the image dataset is presented in **Error! Reference source not found.**.

Table 4.4 Image dataset statistics

Number of frames	**Dataset**	**Size**
32	Whole dataset	6304
	Training	4412
	Validation	946
	Test	946
64	Whole dataset	12608
	Training	8825
	Validation	1891
	Test	1892
96	Whole dataset	18912
	Training	13238
	Validation	2837
	Test	2837
128	Whole dataset	25216
	Training	17651
	Validation	3782
	Test	3783

5. Implementation

5.1. Video Preprocessing

5.1.1. Face Detection and Extraction

The first step of video preprocessing was to extract faces from frames and save them as new video clips. This approach, where the cropped face frames are stored as a new dataset reduces the time and effort of the video preprocessing phase before training the model, instead of implementing the face recognition step directly on the original videos, applying other video preprocessing methods and feeding the data into the neural network [73]. Apart from cropping the faces of the tennis players, the method also resizes new frames to a fixed size of 224x224, further reducing the need for later preprocessing. Moreover, this method offered the possibility to inspect each new video then and remove any faulty frames, making the dataset more accurate. Other preprocessing steps such as normalization, or adjusting the length are applied to the videos with the cropped faces before the dataset is split into training, validation, and test sets.

A structured approach involving face detection algorithms and frame extraction techniques was implemented for this task.

The face extracting logic was implemented in method `process_directory`. First, video files are systematically identified and loaded from the specified input directory using the `list_video_files` function, which traverses the directory and compiles a list of all video files with the extensions "*.mp4*".

Next, the `crop_faces_from_video` method is called which reads each video frame by frame, applying the `crop_face_frame` function to detect and crop faces. All the successfully cropped faces are then written to a new video file, with the output configured to match the original video's frame rate, thus preserving the temporal characteristics of the original footage.

The method uses the *cv2* library [32] for the video processing, It loads the video specified by the `input_video_path` using the `cv2.VideoCapture` method and then retrieves the frames per second (fps) of the video using the `cap.get(cv2.CAP_PROP_FPS)` method on the loaded video. Next, it defines the codec and creates a `VideoWriter` object to write the

output video with the frames per second that correspond to the original video and a new fixed frame size, in this case, 224x224 pixels. The method then enters a loop that processes the video frames until the video is fully read. Inside the loop, it reads a frame from the video using `cap.read()`. If the frame is read successfully, the `crop_face_frame` method is called with the newly read frame as a parameter. Once that method returns the new frame, if the frame contains a face, it's written to the output file. After all the frames are processed, the open resources are closed.

The `crop_face_frame` method takes a NumPy array of the original frame, and the desired height and width to resize the processed frame. Then it uses a face detector to detect the face in the frame. After trying multiple face detection libraries including Dlibs CNN Face Detector [74], and YOLO (You Only Look Once) [75], the MTCNN (Multi-task Cascaded Convolutional Networks) [43], [76] was regarded as the most accurate. If the face is found in the frame, the face bounding box is then cropped as a new frame and resized. The method returns an array of the cropped face with shape (new_height, new_width, channels) and a flag indicating if the face was detected.

In subsequent figures (Figure 5.1, Figure 5.2, **Error! Reference source not found.**, Figure 5.4), the extracted frames for every emotion after cropping the faces and resizing can be seen.

Figure 5.1 Neutral

Figure 5.2 Happiness

Figure 5.3 Sadness

Figure 5.4 Anger

5.1.2. Manual processing

After the faces had been cropped and saved as new videos, by analyzing the performance and the accuracy of the face detection models, it was noticed that many frames were faulty - containing either faces of people from the audience or some inanimate objects (Figure 5.5). To reduce the error in the dataset, all videos were carefully examined frame-by-frame using the Adobe Premiere Pro tool [77]. 2 seconds were taken as the lower limit of the video length

to preserve the temporal dimension, so, if the video was less than 2 seconds long after cropping the inadequate frames, it was removed from the dataset.

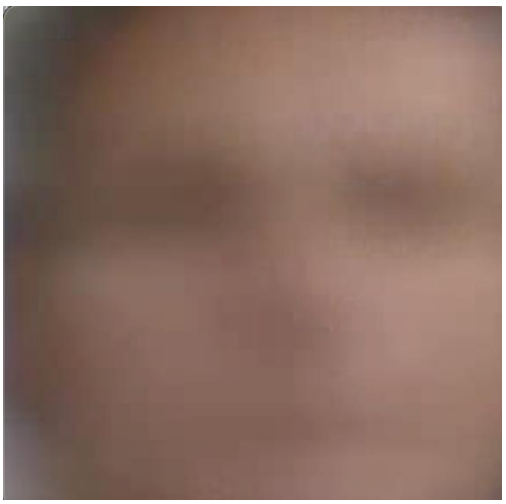

Figure 5.5 Faulty frame

5.1.3. Preprocessing

After the videos are manually edited, they are ready for the preprocessing pipeline. The preprocessing pipeline contains several video and frame processing steps.

The first step is the adjustment of the video length. To establish video length uniformity across all the videos, the `adjust_video_length` method is utilized. This method takes the desired number of frames and the original array of frames, and either repeats frames in videos with less than the desired number of frames, or samples frames from the longer videos until the wanted frame number is not reached.

After adjusting the length of the videos, the next step is normalization in the `normalize_frames` function which takes the frames with the adjusted lengths and normalizes the pixel values in every frame to the range [0, 1].

To further increase the robustness of the models, data augmentation techniques are applied to normalized frames through the `augment_frames` function. The implemented augmentations include:

- **Horizontal flipping** - each frame has a 50% chance of being horizontally flipped. This helps the model's invariance to the direction of facial expressions.
- **Gaussian blurring** - each frame also has a 50% chance of undergoing Gaussian blur which simulates variations in image sharpness and helps the model become more robust to slight blurriness or out-of-focus circumstances

After the frames go through the preprocessing pipeline, they are ready to be split into training, validation, and test sets.

5.1.4. Training

For this paper, several cutting-edge technologies and libraries were used to design, train, and evaluate the neural networks more efficiently. The TensorFlow [33] and Keras [78] libraries were the foundational tools for building and training the models. TensorFlow provides a flexible, powerful platform that supports large-scale machine learning and deep learning tasks, making it ideal for both research and production environments. Its support for GPU acceleration was particularly beneficial for computationally expensive tasks that were necessary for this project. Keras, which is integrated with TensorFlow, simplifies the process of creating neural network architectures with its user-friendly, high-level API, allowing rapid experimentation with different models.

So, the paper focused on TensorFlow's pre-trained 2D Convolutional Neural Network (CNN) models, specifically VGG16, VGG19 [55] and ResNet [56]. These models, known for their robust feature extraction capabilities, are well-suited for tasks such as image classification, or more specifically, facial emotion recognition. By employing VGG models and ResNet , the training process was accelerated, and high performance was maintained by leveraging the strengths of these proven architectures.

Hugging Face Transformers [79] were also integrated to leverage pre-trained transformer models for specific tasks. Hugging Face provides state-of-the-art natural language processing (NLP) and vision transformer (ViT) models that have been trained on large datasets.

The code for the implementations of the neural network training is contained in three training files, one for training 2D CNNs, the second for training 3D CNNs, and the third for training transformers. First, each file contains a `train_model` method, which takes in a model

(`tf.keras.Model`) to train as a parameter, as well as training and validation sets, epoch size, and batch size. The method then creates a directory for that specific model to save training data such as frame number, category (emotions, arousal, valence), number of labels, and all labels in a new file. It then defines checkpoints for saving the best results, trains the model, evaluates the model on the validation set, and returns the results.

Models are compiled in a separate `models.py` file leveraging transfer learning with pre-trained CNN architectures. The main method in this file is `build_model` with `base_model` and `num_classes` parameters, the former representing an already trained base model such as ResNet or VGG16, and the latter representing the number of output classes. The models are loaded without their top layers, which are typically used for classification tasks, allowing for the customization necessary to adapt to the specific dataset used in the project.

In this transfer learning approach, the layers of the pre-trained base models are frozen, preventing the weights from being updated during training. This step is crucial as it preserves the knowledge these models have learned from large-scale datasets like ImageNet [52], allowing the FER model to benefit from this pre-existing feature extraction capability. On top of the base model, custom layers are added to tailor the model to the specific task of emotion recognition. A GlobalAveragePooling2D layer for 2D CNNs, or a GlobalAveragePooling3D layer for 3D CNNs is used to reduce the spatial dimensions of the feature maps, converting them into a 1D feature vector. This is followed by a fully connected Dense layer with 1024 units and ReLU activation, which introduces non-linearity and enhances the model's ability to learn complex patterns. Finally, an output Dense layer with a softmax activation function is added, corresponding to the number of emotion classes in the dataset. This layer outputs a probability distribution over the classes, allowing the model to make predictions. After the model is assembled, it's compiled using the Adam optimizer and categorical cross-entropy loss function.

The training process then starts by calling the `preprocess_videos` method from the preprocessing step, which returns processed videos with augmented frames and their one-hot encoded labels.

If the base model is a 3D CNN the videos are directly fed to the neural network, while for 2D neural networks and transformer neural networks, the videos are transformed into a set of images using the `transform_videos_to_images` method. The dataset is then split into training, validation, and testing sets. After that, the model is built by calling builder methods

from the `models.py` file, and the model and datasets are forwarded to the `train_model` method.

After the model is done with the training, the next stage is to save and visualize results which is done by calling methods from the `result_processing.py`.

For this paper, multiple types of neural networks were trained, including both convolutional neural networks (CNNs) and Vision Transformers, to evaluate their performance on emotion recognition in tennis players.

The first set of models were 2D CNNs, specifically the **VGG16**, **VGG19**, and **ResNet50** architectures, which are well-known for their effectiveness in image classification tasks. These models were trained using various combinations of input frame counts, ranging from 32 to 128 frames ([32, 64, 96, 128]), and different numbers of training epochs ([10, 15, 20]). The goal was to understand how the amount of temporal data (i.e., the number of frames) and training time (i.e., the number of epochs) impacted the model's ability to learn and generalize to unseen data.

In addition to the 2D CNNs, the paper also experimented with 3D CNN models. These were based on pre-trained 3D implementations of VGG16 and ResNet [80], which extend the standard 2D CNN architectures to handle video input by incorporating an additional dimension for time. For these 3D CNNs, different frame counts ([32, 64, 96, 128]) were similarly tested, but the number of epochs was limited to 15 and 20. This allowed the paper to assess how well the models could extract spatiotemporal features from the video data, which is crucial for understanding dynamic information across multiple frames.

Finally, a **Vision Transformer (ViT)** model was trained [60]. Like the CNN models, the Vision Transformer was trained using four combinations of frame counts and epochs including [(64, 10), (96, 10), (96, 15), (128, 15)] where the first number is the number of frames and the second number the number of epochs.

The inclusion of Vision Transformers made it possible to compare the performance of these transformer-based models with the more traditional CNN-based approaches, particularly in how well they handle temporal information across sequences of video frames.

5.1.5. Result processing

The last part of the training is the processing of the results. This process is handled through a series of methods that save the training history, plot performance metrics, and display the

confusion matrix, which is crucial for assessing the model's classification performance. The `save_results` function saves both the training history and the evaluation metrics to files. It ensures that a directory named after the model is created if it doesn't already exist. The training history, which includes data such as loss and accuracy per epoch, is stored in a JSON file, making it easy to reload and analyze later. Similarly, the evaluation metrics, such as validation loss and accuracy, are saved in another JSON file. This structured storage approach allows for easy comparison between different model configurations and training runs.

To visualize the model's learning progress, the `plot_loss` function is utilized. This function generates a plot of the training and validation loss over epochs, providing a clear visual indication of how well the model is learning and whether it is overfitting. By examining these plots, trends in loss reduction can be identified, potential overfitting detected (when validation loss begins to increase while training loss decreases), and consequently, parameters can be tweaked for better performance.

The `plot_confusion_matrix` function plays a key role in evaluating the model's performance on the test set. It generates and displays a confusion matrix, which provides detailed insights into the classification accuracy across different classes. The confusion matrix is particularly useful for identifying specific classes where the model might be underperforming.

Metrics such as precision, recall, F1 score, and overall accuracy are computed using the `calculate_metrics` function, which aggregates these crucial performance indicators into a comprehensive dictionary. This information is further saved using the `save_metrics` function, allowing for structured reporting and review.

Together, these methods form a comprehensive results-processing framework that not only saves and organizes important data but also provides visual tools to assess the model's performance effectively. This approach ensures that the results of the machine learning experiments are well-documented, easily interpretable, and ready for further analysis or reporting.

For each model, validation and testing metrics were calculated and saved to clearly show the models' capabilities.

6. Results

During the training, different computational models showcased varying degrees of effectiveness, with 2D CNNs emerging as particularly proficient, notably VGG19 and ResNet50. VGG19 excelled, achieving a validation accuracy of 99.31% with a loss of 0.0396 for datasets processed with 128 frames. This contrasts with VGG16, which posted a validation accuracy of 89.04% and a loss of 0.2956 under the same conditions. VGG19's additional convolutional layers provide enhanced capability for extracting more complex features from static frames, which is essential for accurately discerning emotional expressions from visual data.

On the other hand, the paper also hints at potential over-training in models processed with 96 or 128 frames. The duplication of frames in these scenarios is significantly higher compared to those with 32 or 64 frames, suggesting that the latter might offer more reliable and generalizable results. This over-training could manifest as models learning to recognize repeated patterns specific to the training dataset rather than generalizing from genuine emotional expressions, thereby diminishing their effectiveness when exposed to new data. Specifically, for datasets with 32 frames, the highest validation accuracy reached was 94.44% with a loss of 0.19 using ResNet50, as well as for 64 frames, where ResNet50 achieved 98.52% accuracy with a loss of 0.0552.

Analyzing the impact of varying the number of epochs and frames on the performance of 2D CNN architectures like ResNet50, VGG16, and VGG19 revealed interesting patterns in the training results. Generally, increasing the number of epochs tended to improve model performance, as evidenced by lower validation losses and higher accuracy, suggesting more extensive training allows the models to better learn and generalize from the dataset. For instance, when training ResNet50 with 32 frames across 10, 15, and 20 epochs, there's a clear trend of increasing accuracy and decreasing loss with more epochs, highlighting the benefits of extended training periods. The influence of the number of frames already mentioned in the previous paragraph on the models' performance is also noteworthy. As said, models trained with a higher number of frames, such as 96 or 128, consistently show improved accuracy compared to those trained with fewer frames, like 32 or 64. Comparatively, changes in the number of epochs seem to have a more pronounced effect on the outcomes than variations in the number of frames. For example, increasing epochs from

10 to 15 or 20 generally results in more significant improvements in performance metrics than increasing the number of frames from 64 to 96. This suggests that, within the tested ranges, deeper training - more epochs is more beneficial than simply providing more input data, i.e. in this case, more frames.

Detailed results for 2D models are presented in the next tables (**Error! Reference source not found.**, **Error! Reference source not found.**,) and figures (Figure 6.1, Figure 6.2, Figure 6.3, Figure 6.4):

Table 6.1 2D VGG16 results

Frames	32			64			96			128		
Epochs	10	15	20	10	15	20	10	15	20	10	15	20
Validation loss	0,719	0,611	0,563	0,559	0,435	0,337	0,423	0,296	0,249	0,37	0,269	0,19
Validation accuracy	0,7	0,75	0,769	0,779	0,828	0,872	0,836	0,89	0,905	0,857	0,897	0,932

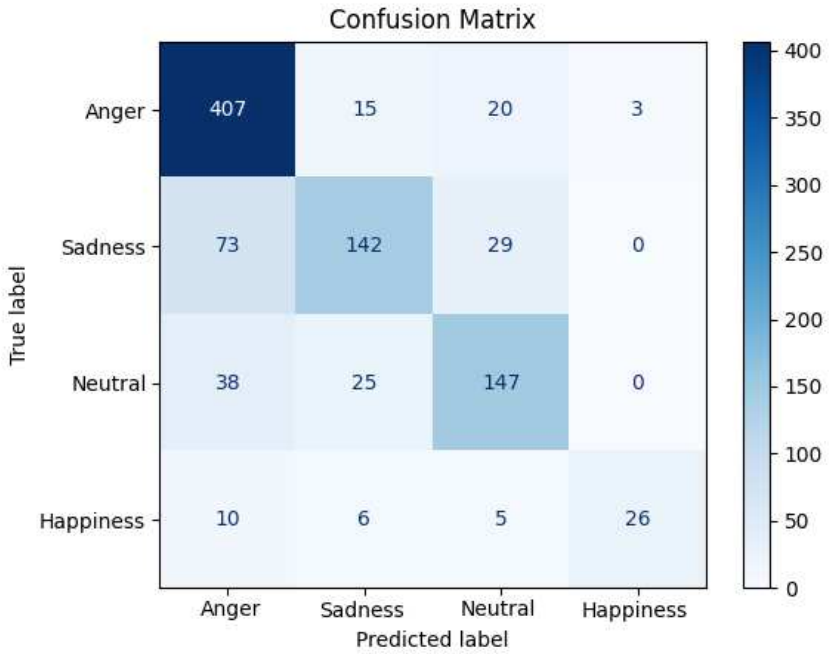

Figure 6.1 Confusion matrix of test set results for 2D VGG16 (32 frames, 20 epochs)

Table 6.2 2D ResNet 50 results

Frames	32			64			96			128		
Epochs	10	15	20	10	15	20	10	15	20	10	15	20
Validation loss	0,213	0,21	0,19	0,091	0,053	0,055	0,075	0,032	0,026	0,053	0,032	0,022
Validation accuracy	0,936	0,946	0,944	0,978	0,984	0,985	0,978	0,993	0,995	0,983	0,994	0,995

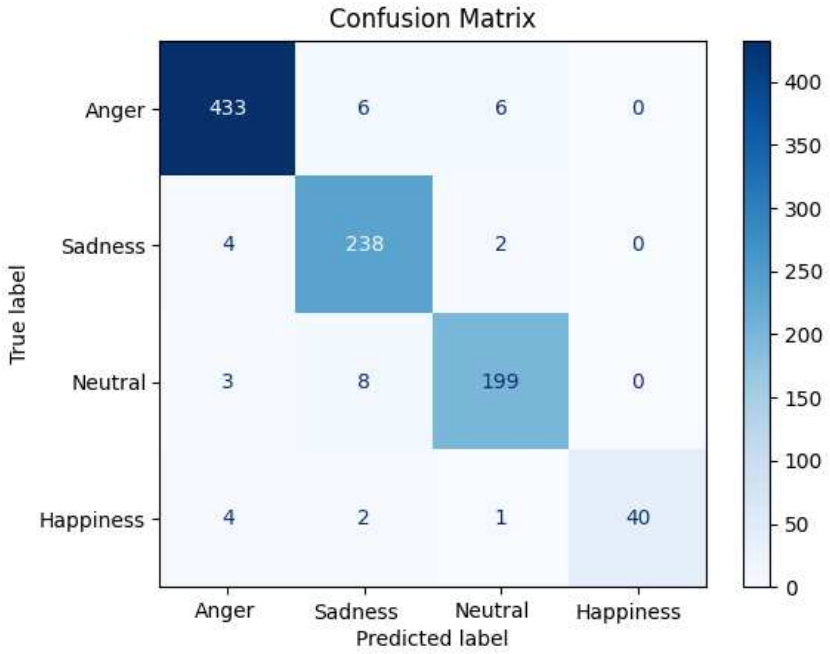

Figure 6.2 Confusion matrix of test set results for 2D ResNet50 (32 frames, 20 epochs)

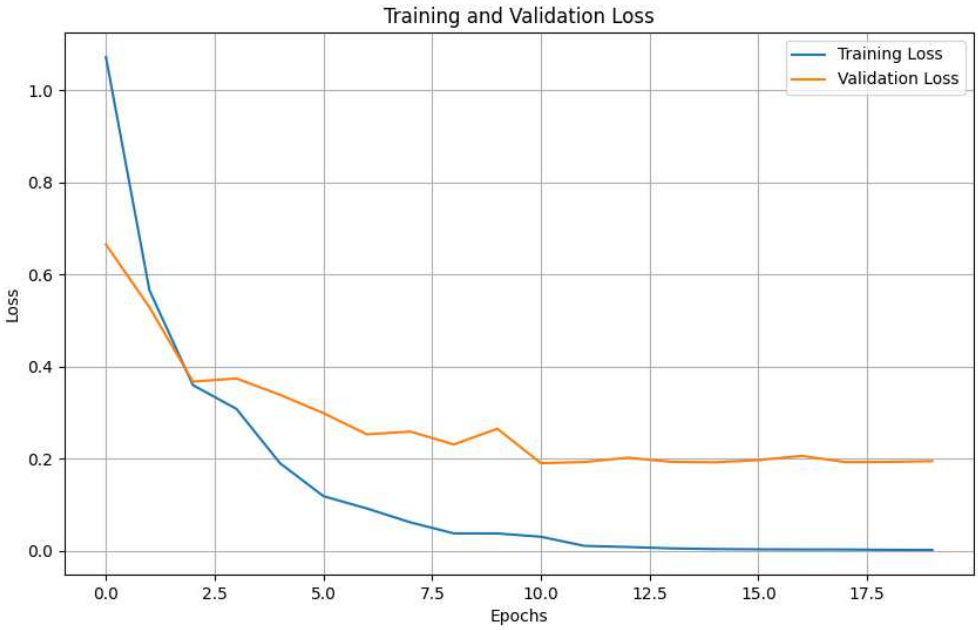

Figure 6.3 Training and validation loss for 2D ResNet50 (32 frames, 20 epochs)

Table 6.3 2D VGG19 results

Frames	32	64	96		128	
Epochs	10	10	10	15	10	15
Validation loss	0,193	0,116	0,07	0,046	0,07	0,041
Validation accuracy	0,938	0,962	0,981	0,992	0,981	0,962

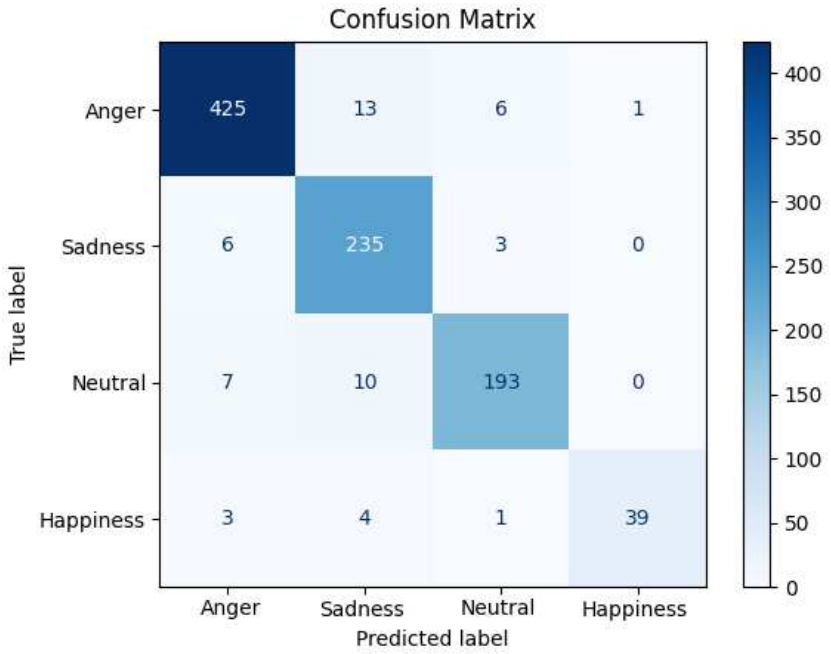

Figure 6.4 Confusion matrix of test set results for VGG19 (32 frames, 10 epochs)

Conversely, both 3D CNNs demonstrated lower efficacy, consistently achieving around 53.33% accuracy through all combinations (Figure 6.5). These models were trained directly on whole video sequences, aiming to capitalize on both spatial and temporal data. However, the training on a relatively small video dataset likely hindered the 3D CNNs' ability to learn. This limitation, coupled with the high computational demands of 3D CNNs, may have negatively impacted their performance.

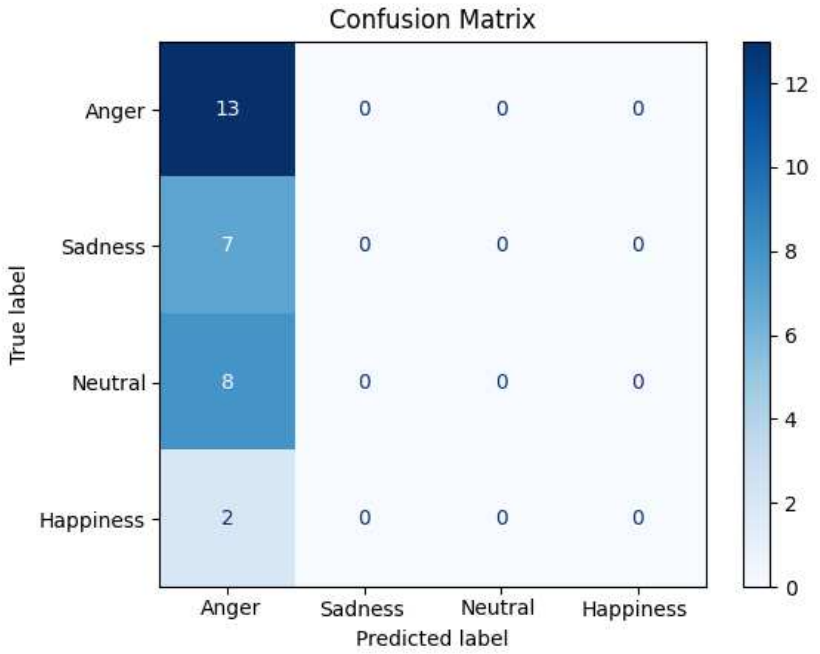

Figure 6.5 Confusion matrix on test set for 3D ResNet model

The transformer model, in this case, the Vision Transformer (ViT) model, also exhibited low performance in emotion recognition tasks from tennis match videos, with the highest validation accuracy being 44.7% when trained on datasets with 96 frames in 10 epochs (Table 6.4, Figure 6.6). This suboptimal performance may have stemmed from several key factors including the nature of their training and deployment environment. Transformers usually require extensive pre-training on large datasets to develop a robust feature understanding. Another significant factor for under-performing could be a class imbalance within the training data, which can lead the model to predict the over-represented class more frequently to minimize loss. In this case, the '*Anger'* emotion was the most represented, which may have biased the model towards predicting this class more frequently. Potential overfitting due to training on a small dataset with repetitive patterns (such as 96 or 128 frames leading to many duplicate frames) might have also degraded the model's predictive accuracy. Furthermore, there's also a possibility that a bug or an error in the implementation might be influencing the results, where the model might not be utilized or evaluated correctly, leading to seemingly poor performance. This highlights the need for rigorous testing and validation of the model's implementation to ensure that all components function as intended. Addressing these challenges could lead to more refined models that perform

better not only in specific tasks like emotion recognition from tennis videos but also in broader image and video-based machine learning applications.

Table 6.4 ViT results

Frames	32	64	96	128
Epochs	10	10	10	10
Validation loss	1,236	0,207	0,22	0,21
Validation accuracy	0,424	0,447	0,437	0,441

Figure 6.6 Training and validation loss for ViT (64 frames, 10 epochs)

In conclusion, based on the training results presented for emotion recognition in tennis using various models, the VGG19 model under the 2D CNN architecture emerged as the most effective in achieving high validation accuracy. Specifically, VGG19, when trained on 128 frames for 10 epochs, achieved a notable validation accuracy of 99.31%, which stands out as the highest across all the tested configurations. The best metrics on the test set acquired by the VGG19 models are shown in Table 6.5.

Table 6.5 VGG19 best test set metrics

Metric	**Score**
Precision	0,993
Recall	0,991
F1 score	0,992
Accuracy	0,992

This high performance can be attributed to VGG19's deep architecture, which is well-suited for capturing complex patterns in visual data, making it particularly effective for detailed image-based tasks like emotion recognition. The architecture's multiple layers of convolution and pooling operations help in extracting robust features from the frames extracted from the tennis video footage, which likely contributed to its superior performance.

Conclusion

This master's thesis aimed to advance the understanding and implementation of Facial Emotion Recognition (FER) within the unique context of tennis, seeking to bridge the analytical gap between physical prowess and mental resilience in sports. The research began with the creation of a new dataset comprised of video footage capturing emotional expressions of players during matches. The dataset was then carefully curated to ensure quality and variability, subsequently facilitating robust training and evaluation of various machine learning models. The primary architectures explored were 2D CNNs, 3D CNNs, and transformers, each offering distinct advantages in processing and recognizing facial expressions from static images and video sequences.

The results from the model training were revealing. 2D CNNs demonstrated high efficacy, with improved results observed in scenarios with an increased number of frames per video, suggesting that more temporal data provides a richer context for accurate emotion recognition. In contrast, 3D CNNs, which analyzed the full temporal sequence of videos, showed lower accuracy. This underperformance could be attributed relatively smaller size of the video dataset, which limited the training effectiveness of these inherently data-hungry models. Moreover, the transformer models frequently misclassified subtle or less pronounced emotional states, often defaulting to an 'Anger' category, which might indicate either an issue with the model's training on the specific dataset characteristics or a need for further fine-tuning and adjustment of training parameters.

Also, further investigation into the emotional dimensions of valence and arousal could enrich the analysis, providing a more precise definition of players' emotional states during matches, and offering deeper insights into how players react under pressure.

Looking forward, the exploration could extend into alternative architectures such as Graph Neural Networks (GNNs), which might provide novel ways to model the complex, non-Euclidean data structures present in facial recognition tasks. Additionally, addressing issues like class imbalance within the training data and optimizing model parameters more effectively could enhance the accuracy and reliability of FER systems in sports analytics.

Literature

[1] *Video Assistant Referee (VAR) protocol | IFAB*. (n.d.). Retrieved 4 August 2024, from https://www.theifab.com/laws/latest/video-assistant-referee-var-protocol/

[2] *Hawkeye Tennis Line-Calling System*. (n.d.). Retrieved August 4, 2024, from https://www.topendsports.com/sport/tennis/hawkeye.htm

[3] *What are the longest tennis matches in Grand Slam history?* (2024, January 24). ESPN.Com. Retrieved August 4, 2024, from https://www.espn.com/tennis/story/_/id/39379014/what-longest-major-matches-tennis-history

[4] admin. (2019, October 21). How Advanced Technology Has Helped Tennis. *Dragon Courts*. Retrieved August 4, 2024, from https://www.allabouttennis.co.uk/blog/how-advanced-technology-has-helped-tennis/

[5] *Wimbledon | IBM*. (n.d.). Retrieved 4 August 2024, from https://www.ibm.com/sports/wimbledon

[6] *IBM and the US Open*. (n.d.). Retrieved 4 August 2024, from https://www.ibm.com/sports/usopen

[7] *IBM watsonx — An AI and data platform built for business*. (2024, April 4). Retrieved August 4, 2024, from https://www.ibm.com/watsonx

[8] Norris, D. (2011, September 12). *IBM PointStream at the US Open tennis tournament, Big Data in the real world - Bloor Research*. Retrieved August 4, 2024, from https://www.bloorresearch.com/2011/09/ibm-pointstream-open-tennis-tournament-big-data-real/

[9] Janelle, C. M., Fawver, B. J., & Beatty, G. F. (2020). Emotion and Sport Performance. In G. Tenenbaum & R. C. Eklund (Eds.), *Handbook of Sport Psychology* (1st ed., pp. 254–298). Wiley.

[10] Gu, S., Wang, F., Patel, N. P., Bourgeois, J. A., & Huang, J. H. (2019). A Model for Basic Emotions Using Observations of Behavior in Drosophila. *Frontiers in Psychology*, *10*, 781

[11] Russell, J. (1980). A Circumplex Model of Affect. *Journal of Personality and Social Psychology*, *39*, 1161–1178

[12] Barrett, L. F. (1998). Discrete Emotions or Dimensions? The Role of Valence Focus and Arousal Focus. *Cognition & Emotion*, *12*(4), 579–599

[13] Plutchik, R. (1982). A psychoevolutionary theory of emotions. *Social Science Information*, *21*(4–5), 529–553

[14] Ekman, P. (1999). Basic emotions. In *Handbook of cognition and emotion* (pp. 45–60). John Wiley & Sons Ltd

[15] Dehghani, F., & Zaman, L. (2023). *Facial Emotion Recognition in VR Games* (No. arXiv:2312.06925). arXiv

[16] Akbar, M. T., Ilmi, M. N., Rumayar, I. V., Moniaga, J., Chen, T.-K., & Chowanda, A. (2019). Enhancing Game Experience with Facial Expression Recognition as Dynamic Balancing. *Procedia Computer Science*, *157*, 388–395

[17] Guo, R., Guo, H., Wang, L., Chen, M., Yang, D., & Li, B. (2024). Development and application of emotion recognition technology — a systematic literature review. *BMC Psychology*, *12*, 95

[18] Canal, F. Z., Müller, T. R., Matias, J. C., Scotton, G. G., de Sa Junior, A. R., Pozzebon, E., & Sobieranski, A. C. (2022). A survey on facial emotion recognition techniques: A state-of-the-art literature review. *Information Sciences*, *582*, 593–617

[19] Spezialetti, M., Placidi, G., & Rossi, S. (2020). Emotion Recognition for Human-Robot Interaction: Recent Advances and Future Perspectives. *Frontiers in Robotics and AI*, *7*

[20] McColl, D., Hong, A., Hatakeyama, N., Nejat, G., & Benhabib, B. (2016). A Survey of Autonomous Human Affect Detection Methods for Social Robots Engaged in Natural HRI. *Journal of Intelligent & Robotic Systems*, *82*(1), 101–133

[21] Boughanem, H., Ghazouani, H., & Barhoumi, W. (2023). Facial Emotion Recognition in-the-Wild Using Deep Neural Networks: A Comprehensive Review. *SN Computer Science*, *5*, 96

[22] Kamachi, M., Lyons, M., & Gyoba, J. (1997). The japanese female facial expression (jaffe) database. *Availble: Http://Www. Kasrl. Org/Jaffe. Html*.

[23] Lucey, P., Cohn, J., Kanade, T., Saragih, J., Ambadar, Z., & Matthews, I. (2010). The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, CVPRW 2010* (p. 101)

[24] Mollahosseini, A., Hasani, B., & Mahoor, M. H. (2019). AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild. *IEEE Transactions on Affective Computing*, *10*(1), 18–31

[25] Goodfellow, I. J., Erhan, D., Carrier, P. L., Courville, A., Mirza, M., Hamner, B., Cukierski, W., Tang, Y., Thaler, D., Lee, D.-H., Zhou, Y., Ramaiah, C., Feng, F., Li, R., Wang, X., Athanasakis, D., Shawe-Taylor, J., Milakov, M., Park, J., … Bengio, Y. (2013). *Challenges in Representation Learning: A report on three machine learning contests* (No. arXiv:1307.0414). arXiv

[26] Li, S., Deng, W., & Du, J. (2017). Reliable Crowdsourcing and Deep Locality-Preserving Learning for Expression Recognition in the Wild. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2584–2593

[27] Barsoum, E., Zhang, C., Ferrer, C. C., & Zhang, Z. (2016). *Training Deep Networks for Facial Expression Recognition with Crowd-Sourced Label Distribution* (No. arXiv:1608.01041). arXiv

[28] Dhall, A., Goecke, R., Lucey, S., & Gedeon, T. (2012). Collecting Large, Richly Annotated Facial-Expression Databases from Movies. *IEEE Multimedia*, *19*, 34–31.

[29] Dhall, A., Goecke, R., Joshi, J., Sikka, K., & Gedeon, T. (2014). Emotion Recognition In The Wild Challenge 2014: Baseline, Data and Protocol. *Proceedings of the 16th International Conference on Multimodal Interaction*, 461–466

[30] Zafeiriou, S., Papaioannou, A., Kotsia, I., Nicolaou, M., & Zhao, G. (2016). Facial Affect "In-the-Wild": A Survey and a New Database. *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1487–1498

[31] Kollias, D., & Zafeiriou, S. (2019). *Aff-Wild2: Extending the Aff-Wild Database for Affect Recognition* (No. arXiv:1811.07770). arXiv

[32] *opencv-python: Wrapper package for OpenCV python bindings.* (4.10.0.84). (n.d.). [C++, Python; MacOS, Microsoft :: Windows, POSIX, Unix]. Retrieved 31 July 2024, from https://github.com/opencv/opencv-python

[33] *TensorFlow*. (n.d.). Retrieved 6 August 2024, from https://www.tensorflow.org/

[34] Parsania, P., & Virparia, P. (2015). A Review: Image Interpolation Techniques for Image Scaling. *International Journal of Innovative Research in Computer and Communication Engineering*, *02*, 7409–7414

[35] Kopalidis, T., Solachidis, V., Vretos, N., & Daras, P. (2024). Advances in Facial Expression Recognition: A Survey of Methods, Benchmarks, Models, and Datasets. *Information*, *15*(3), Article 3

[36] Mumuni, A., & Mumuni, F. (2022). Data augmentation: A comprehensive survey of modern approaches. *Array*, *16*, 100258

[37] Biswas, A., Nasim, M. A. A., Imran, A., Sejuty, A. T., Fairooz, F., Puppala, S., & Talukder, S. (2023). *Generative Adversarial Networks for Data Augmentation* (No. arXiv:2306.02019). arXiv.

[38] Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, *1*, I–I.

[39] Wang, Y., Ai, H., Wu, B., & Huang, C. (2004). Real time facial expression recognition with AdaBoost. *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, *3*, 926-929 Vol.3.

[40] Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, *1*, 886–893 vol. 1.

[41] Michel, P., & El Kaliouby, R. (2003). Real time facial expression recognition in video using support vector machines. *Proceedings of the 5th International Conference on Multimodal Interfaces*, 258–264.

[42] Face Recognition With Histograms Of Oriented Gradients: (2010). *Proceedings of the International Conference on Computer Vision Theory and Applications*, 339–344.

[43] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. *IEEE Signal Processing Letters*, *23*(10), 1499–1503.

[44] Zhang, Z., Lyons, M., Schuster, M., & Akamatsu, S. (1998). Comparison between Geometry-Based and Gabor Wavelets-Based Facial Expression Recognition Using Multi-Layer Perceptron. In *textordmasculine Proc. Int'l Conf. Automatic Face and Gesture Recognition* (p. 459).

[45] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27. IEEE Transactions on Information Theory.

[46] Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. Proceedings of the IEEE.

[47] *Video-based emotion recognition using CNN-RNN and C3D hybrid networks | Proceedings of the 18th ACM International Conference on Multimodal Interaction*. (n.d.). Retrieved 11 August 2024, from https://dl.acm.org/doi/10.1145/2993148.2997632

[48] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations* (pp. 318–362). MIT Press.

[49] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780.

[50] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). *Attention Is All You Need* (No. arXiv:1706.03762). arXiv.

[51] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2009). The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, *20*(1), 61–80. IEEE Transactions on Neural Networks.

[52] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, *25*.

[53] Bharadiya, J. P. (2023). *Convolutional Neural Networks for Image Classification*.

[54] Ballesteros, J. A., Ramírez V., G. M., Moreira, F., Solano, A., & Pelaez, C. A. (2024). Facial emotion recognition through artificial intelligence. *Frontiers in Computer Science*, *6*, 1359471.

[55] Simonyan, K., & Zisserman, A. (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition* (No. arXiv:1409.1556). arXiv.

[56] He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition* (No. arXiv:1512.03385). arXiv.

[57] Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*.

[58] Kondratyuk, D., Yuan, L., Li, Y., Zhang, L., Tan, M., Brown, M., & Gong, B. (2021). *MoViNets: Mobile Video Networks for Efficient Video Recognition* (No. arXiv:2103.11511). arXiv.

[59] Carreira, J., & Zisserman, A. (2018). *Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset* (No. arXiv:1705.07750). arXiv.

[60] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale* (No. arXiv:2010.11929). arXiv.

[61] Bobojanov, S., Kim, B. M., Arabboev, M., & Begmatov, S. (2023). Comparative Analysis of Vision Transformer Models for Facial Emotion Recognition Using Augmented Balanced Datasets. *Applied Sciences*, *13*(22), Article 22.

[62] Rodríguez-Azar, P. I., Mejía-Muñoz, J. M., & Ochoa-Zezzatti, C. A. (2022). Recognition of Facial Expressions Using Vision Transformer. *Científica*, *26*(2), 1–9.

[63] Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, *1*, 57–81.

[64] Henaff, M., Bruna, J., & LeCun, Y. (2015). *Deep Convolutional Networks on Graph-Structured Data* (No. arXiv:1506.05163). arXiv.

[65] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). *Graph Attention Networks* (No. arXiv:1710.10903). arXiv.

[66] Yan, S., Xiong, Y., & Lin, D. (2018). *Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition* (No. arXiv:1801.07455). arXiv.

[67] Ruiz, L., Gama, F., & Ribeiro, A. (2020). Gated Graph Recurrent Neural Networks. *IEEE Transactions on Signal Processing*, *68*, 6303–6318.

[68] Xu, X., Ruan, Z., & Yang, L. (2020). Facial Expression Recognition Based on Graph Neural Network. *2020 IEEE 5th International Conference on Image, Vision and Computing (ICIVC)*, 211–214.

[69] Jekauc, D., Burkart, D., Fritsch, J., Hesenius, M., Meyer, O., Sarfraz, S., & Stiefelhagen, R. (2024). Recognizing affective states from the expressive behavior of tennis players using convolutional neural networks. *Knowledge-Based Systems*, *295*, 111856.

[70] Santana, O. J., Freire-Obregón, D., Hernández-Sosa, D., Lorenzo-Navarro, J., Sánchez-Nielsen, E., & Castrillón-Santana, M. (2023). Facial expression analysis in a wild sporting environment. *Multimedia Tools and Applications*, *82*(8), 11395–11415.

[71] Zhou, L., Zhang, C., & Wang, M. (2023). Emotion recognition algorithm of basketball players based on deep learning. *International Journal of Information and Communication Technology*, *22*(4), 377–390.

[72] *TennisCraft*. (2022, August 9). TennisCraft. https://www.tenniscraft.nyc

[73] Kartali, A., Roglic, M., Barjaktarovic, M., Duric-Jovicic, M., & Jankovic, M. M. (2018). Real-time Algorithms for Facial Emotion Recognition: A Comparison of Different Approaches. *2018 14th Symposium on Neural Networks and Applications (NEUREL)*, 1–4.

[74] King, D. (n.d.). *dlib: A toolkit for making real world machine learning and data analysis applications* (19.24.5) [C++, Python; MacOS :: MacOS X, Microsoft, Microsoft :: Windows, POSIX, POSIX :: Linux]. Retrieved 31 July 2024, from https://github.com/davisking/dlib

[75] Qi, D., Tan, W., Yao, Q., & Liu, J. (2022). *YOLO5Face: Why Reinventing a Face Detector* (No. arXiv:2105.12931). arXiv.

[76] Centeno, I. de P. (2024). *ipazc/mtcnn* [Jupyter Notebook]. https://github.com/ipazc/mtcnn (Original work published 2018)

[77] *Professional video editing software | Adobe Premiere Pro*. (n.d.). Retrieved 4 August 2024, from https://www.adobe.com/uk/products/premiere.html

[78] *Keras: Deep Learning for humans*. (n.d.). Retrieved 6 August 2024, from https://keras.io/

[79] *Transformers*. (n.d.). Retrieved 21 August 2024, from https://huggingface.co/docs/transformers/en/index

[80] Solovyev, R., Kalinin, A. A., & Gabruseva, T. (2022). 3D Convolutional Neural Networks for Stalled Brain Capillary Detection. *Computers in Biology and Medicine*, *141*, 105089.

Sažetak

U ovom radu bavimo se primjenom i vrednovanjem različitih metoda prepoznavanja emocija s lica tenisača tijekom igre. U tu svrhu implementiramo 2D i 3D konvolucijskih neuronskih mreža (CNN) te Vision Transformer (ViT) model prijenosnim učenjem. Korištenjem programskog okvira TensorFlow, primjenjujemo VGG19, ResNet50 i ViT za prepoznavanje emocija iz video zapisa teniskih mečeva. Svi su modeli trenirani i evaluirani na novom skupu podataka sastavljenom od video zapisa teniskih igrača sakupljenih s interneta. Eksperimenti pokazuju da 2D CNN modeli, posebno VGG19 i ResNet50, postižu visoke rezultate u točnosti prepoznavanja emocija.

Ključne riječi: tenis, prepoznavanje emocija lica, konvolucionalne neuronske mreže, transformeri

Summary

In this paper, we explore the application and evaluation of various methods for facial emotion recognition of tennis players during gameplay. For this purpose, we implement 2D and 3D convolutional neural networks (CNN) and the Vision Transformer (ViT) model using transfer learning. Using the TensorFlow framework, we apply VGG19, ResNet50, and ViT to an emotion recognition task on video footage of tennis matches. All models were trained and evaluated on a new dataset consisting of video recordings of tennis players collected from the internet. Experiments show that 2D CNN models, particularly VGG19 and ResNet50, achieve high accuracy in emotion recognition.

Keywords: tennis, facial emotion recognition (FER), convolutional neural networks (CNNs), transformers

Abbreviations

FER	*Facial Emotion Recognition*
NN	*Neural Network*
CNN	*Convolutional Neural Network*
MTCNN	*Multi-task Cascaded Convolutional Networks*
HOG	*Histogram of Oriented Gradients*
SVM	*Support Vector Machine*
RNN	*Recurrent Neural Network*
LSTM	*Long Short-Term Memory*
GNN	*Graph Neural Network*
k-NN	*k-Nearest Neighbors*
ViT	*Vision Transformer*
VGG	Visual Geometry Group
ST-GCN	Spatial-Temporal Graph Convolutional Networks