Universidade de Aveiro
2023

Ricardo
Rodrigues
Azevedo

Uma plataforma computacional para análise e relato de emoções em sessões de videoconferência

A computational platform for emotion analysis and reporting for videoconference sessions

Universidade de Aveiro
2023

Ricardo Rodrigues Azevedo

Uma plataforma computacional para análise e relato de emoções em sessões de videoconferência

A computational platform for emotion analysis and reporting for videoconference sessions

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Doutor Ilídio Castro Oliveira, professor auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e da Doutora Susana Manuela Martinho dos Santos Baía Brás, Investigadora no Instituto de Engenharia Eletrónica e Informática de Aveiro da Universidade de Aveiro.

Dedico este trabalho à minha familia e amigos.

o júri / the jury

presidente / president	Professor Doutor Sérgio Guilherme Aleixo de Matos Professor Auxiliar, Universidade de Aveiro
vogais / examiners committee	Professor Doutor Rolando da Silva Martins Professor Auxiliar, Universidade do Porto - Faculdade de Ciências
	Professor Doutor Ilídio Fernando de Castro Oliveira Professor Auxiliar, Universidade de Aveiro

agradecimentos / acknowledgements

Agradeço toda a ajuda a todos os meus colegas, amigos e companheiros.

Palavras Chave

Reuniões Virtuais, Ferramentas de Videoconferência, Processamento de Imagem e Áudio, Reconhecimento de Emoções, Aprendizagem automática.

Resumo

Numa era marcada pela crescente dependência de reuniões virtuais em várias áreas, como trabalho, educação e saúde, a importância de compreender a dinâmica emocional dos participantes durante videoconferências não deve ser esquecida. As emoções influenciam inerentemente a eficácia da comunicação humana e, assim, moldam os resultados das reuniões virtuais.

O objetivo principal deste trabalho é desenvolver uma plataforma capaz de processar dados multimédia derivados de gravações de vídeo de videoconferências, gerando posteriormente indicadores que denotam a evolução emocional das reuniões. Para atingir tal objetivo, a plataforma precisa ser capaz de estabelecer um sistema de análise dos estados emocionais dos utilizadores durante videoconferências, integrar-se com serviços de videoconferência, implementar um ou mais modelos de classificação de emoções para dados de vídeo e áudio e gerar um relatório abrangente que represente visualmente a evolução emocional das reuniões virtuais. É importante mencionar que a intenção não é criar perfis individuais dos participantes, mas sim compilar uma análise emocional generalizada da reunião virtual.

A arquitetura desenvolvida nesta dissertação combina vários componentes, incluindo um Sistema de Conferência para criar e gerir videoconferências, um Sistema de Pipeline que extrai e analisa imagens e frames de áudio, finalizando com um Servidor de Relatórios que gera um relatório sobre a evolução do estado emocional na videoconferência.

Como prova de conceito, a plataforma cumpre o seu objetivo e consegue gerar um relatório com base no áudio e nas imagens dos utilizadores derivados de uma gravação de videoconferência.

Keywords

Virtual Meetings, Video Conferencing Tools, Image and Audio Processing, Emotion Recognition, Machine Learning.

Abstract

In an era marked by increasing reliance on virtual meetings across various domains such as work, education and healthcare, the significance of understanding the emotional dynamics of participants during video conferences cannot be overstated. Emotions inherently influence the effectiveness of human communication and thereby shape the outcomes of virtual meetings.

The primary objective of this work is to develop a platform capable of processing multimedia data derived from video recordings of video conferences, subsequently generating indicators that illuminate the emotional progression of the meetings. To achieve such an objective the platform needs to be able to establish a pipeline for the analysis of user emotion states during video conferences, integrate with video conferencing services, implement one or more emotion classification models for both video and audio data and generate a comprehensive report that visually represents the emotional evolution of the virtual meetings. It is important to mention that the intention is not to create individual profiles of participants but rather to compile a generalized emotional analysis of the virtual meeting.

The architecture developed in this dissertation combines multiple components, which include a Conference System to create and manage video conferences, a Pipeline System that extracts and analyses image and audio frames finalizing with a Report Server that generates a report of the emotional state evolution in the video conference.

As a proof of concept, the platform fulfills its objective and manages to generate a report based on the audio and images from the users derived from a video conference recording.

Contents

List of Figures

List of Tables

List of Acronyms

API	Application Programming Interfaces
CNN	Convolutional Neural Networks
ECaaS	Emotion Classification as a Service
FER	Facial Expressions Recognition
ICE	Interactive Connectivity Establishment
KMS	Kurento Media Server
RTC	Real-Time Communication
SER	Speech Emotion Recognition
SVM	Support vector machine
SDK	Software development kit
TCP	Transmission Control Protocol
WebRTC	Web Real-Time Communication

CHAPTER 1

Introduction

1.1 Context and Motivation

In today's interconnected world, video conferencing has become an integral part of communication in the work, education and health environment. With the increasing reliance on virtual meetings, understanding the emotional dynamics of users during video conferences has significant importance, as emotions are a crucial part of our everyday life. They influence decision-making and help us understand and be understood by others[1].

Emotion recognition and sentiment analysis using machine learning algorithms have reached newer milestone in recent times, and with the ability to classify emotions by audio and image we have the ground work to provide a good prediction of the evolution of the emotional state of participants in a video conference environment. The ability to analyze this evolution holds great potential for improving the quality and impact of these meetings. By gaining insights into the emotional experiences of participants, we can improve the conditions of the virtual meeting environment in order to improve the communication between the participants.

It is important to reinforce that the intention is not to profile each individual participant but adquire an general emotion analyse of the virtual meeting. Additionally, it's worth noting that the European Union has been actively pursuing initiatives to regulate emotion detection by computer systems, further emphasizing the need for ethical and responsible use of emotional analysis technologies in various contexts, including video conferencing. Some of this initiatives include:

- **Artificial Intelligence Act**: The EU unveiled a draft for the Artificial Intelligence (AI) Act in April 2021 [2]. This act aims to limit the use of biometric identification systems, including facial recognition, that could lead to ubiquitous surveillance.
- **Facial Emotion Recognition (FER)** The European Data Protection Supervisor published a TechDispatch on Facial Emotion Recognition in May 2021 **edps2021fer**. FER is a technology that analyses facial expressions from static images and videos to reveal information about one's emotional state.

- **Regulating Facial Recognition**: The European Parliament has published an in-depth analysis on regulating facial recognition technologies[2].This paper provides an overview of the technologies, economics, and different uses of facial recognition technologies, and examines the recent proposal for an EU artificial intelligence act.

This work is one of the many active researches being studied at the Institute of Electronics and Informatics Engineering of Aveiro (IEETA), from which the proposal for this dissertation emerged.

1.2 Objectives

The main purpose of this work is to create a computational platform to process multimedia data from video conferences and report indicators on the emotional evolution of the meeting. By creating a pipeline that analyzes the evolution of user emotion states during video conferences and reporting its results, this dissertation aims to provide a valuable method to improve human communication in the work, education and health environment in order to understand how the people feel in said environments. The resulting knowledge can be utilized to enhance the effectiveness of virtual meetings, improve communication, and ultimately contribute to a better environment to the people. By the end of this work we aim to have a platform that can:

- Establishing a pipeline for the systematic analysis of user emotion states during video conferences.
- Integrating the platform seamlessly with video conferencing services to facilitate the collection of video data directly from conferences.
- Implementing one or more emotion classification models for video and audio data, enabling the categorization of emotions exhibited by participants.
- Generating comprehensive reports that visually represent the emotional evolution of virtual meetings, providing valuable insights to enhance communication and improve the virtual meeting experience.

1.3 Outline

In Chapter 1 the motivations for the development of this dissertation are presented, as well as its objectives. Chapter 2 describes key concepts required for the understanding of the technologies used in this dissertation. In Chapter 3 we present the description of the system development as well as its requirements and use case. Chapter 4 is where the architecture of the system is presented describing each component and their interactions with one another. In Chapter 5 we describe how the system was implemented. Chapter 6 presents the system results and validates its performance. In Chapter 7 we finalize the dissertation discussing the results and proposition of future work.

CHAPTER 2

Concepts and technologies

2.1 Emotions as a computational modality

According to Rosalind W.Picard computational modeling of emotion is an interdisciplinary endeavor between, in particular, psychology and computer science[3]. The goals of computational modeling of emotion largely correspond to the general goals of AI, when these are restricted to the domain of emotions [4]. Emotion is a complex and intense psycho-physiological experience of an individual's state of mind when reacting to biochemical and environmental influences [5] as so the emotional state of a person can be detected from a variety of behavioral signals, such as facial expressions, voice, text, and body gestures, and physiological signals, for instance, the heart rate, skin temperature, Electrocardiogram (ECG), Electromyogram (EMG) and Electroencephalogram (EEG) [6]. According to the studies, there are seven basic emotions that can be read by the current state of machine learning models namely - anger, sadness, happiness, disgust, fear, surprise and neutral [7]. In this dissertation the detection of emotion will be by facial expressions and voice with Facial Expressions Recognition (FER) and Speech Emotion Recognition (SER).

Facial Emotion Recognition (FER)

FER refers to identifying expression that convey basic emotions such as fear, happiness, and disgust, etc [8]. The method of recognizing face emotions starts with face detection to end in the emotion classification [7]. Face detection is the ability of the computer to recognize a face normally using a face detection model like the Haar feature-based cascade classifiers [9]. After the face detection the next step is feature extraction, key focal marks in the face that gives the shape and location of the important biometric parts of the face such as eyes, nose and mouth [7]. The best features are then chose, nowadays by a machine learning classifiers, and the emotion is deferred by the features chosen. In image classification, Convolutional

Neural Networks (CNN) have shown great potential due to their computational efficiency and feature extraction capability [10].

Speech Emotion Recognition (SER)

Speech is a natural and commonly used medium of interaction among human beings [11]. SER is the identification of the lower-level features of speech from its higher-level valid emotional states [12]. Speech is a continuous signal that convey information, express emotions, and share meaning [13] and SER algorithms, similar to FER required several steps [14], pre-processing of the signal, this involves noise reduction and silence removal, feature extraction, feature selection and classification. A large number of classification techniques are currently being used in the study of speech emotion recognition with some success[12], CNN and Support vector machine (SVM) are one of many in the list.

Emotions Classification as a Service

Emotion Classification as a Service (ECaaS) is a cloud-based offering that provides the capability to classify and analyze emotions in various types of data, including text, audio or video. ECaaS leverages machine learning and natural language processing models to automatically detect and categorize emotions, allowing businesses and developers to integrate emotion analysis into their applications and services.

This service finds applications in a range of industries, including customer service, market research, mental health, and entertainment. For instance, in customer service, ECaaS can be used to analyze customer feedback and sentiment to better understand customer emotions and improve service quality. In mental health, it can assist therapists in monitoring patients' emotional states and progress. In the entertainment industry, it can enhance user experiences in gaming and virtual reality by creating more emotionally responsive environments.

ECaaS providers typically offer Application Programming Interfaces (API)s and Software development kit (SDK)s, making it easy for developers to integrate emotion analysis into their software applications and systems without having to build complex emotion recognition models from scratch. Some examples of available APIs and SDKs are show in table 2.1. In this dissertation API Face - Microsoft, Skybiometric and Morphcast were used in early development of the system.

Service	SDK/API	Video/Audio	Free Subscription
Api Face - Microsoft	API	Video Only	YES
Amazon Rekognition	API	Video Only	NO
Skybiometric	API	Video Only	YES
Morphcast	SDK	Video Only	YES
Hume.ai	API and SDK	Video and Audio	YES

Table 2.1: Sdks and Apis Services

2.2 Video Conference Systems

Video conferencing is a communication technology that gives individuals the opportunity to communicate with each other in real time, using video and audio. It enables participants to see, hear, and communicate with each other as if they were in the same physical meeting room, regardless of geographical distances. Video conferencing has become an essential tool for businesses, educational institutions and healthcare providers. There are three key components in a video conference system:

- Audio: Participants use microphones and speakers or headphones to send and receive audio data.
- Video: Cameras capture real-time video feeds of participants.
- Networking: The internet or dedicated communication lines facilitate the exchange of audio and video data between participants.

The different modules that compose a video conference system are as follow:

- Endpoint Devices: These are the devices used by the participants to join and interact in the video conference. Desktop computers, laptops, tablets and smartphones are some examples of the devices. Each endpoint device requires a camera, microphone and speakers.
- Video Conferencing Applications: This is the interface participants use to interact in video conferences and what facilitates the connection between the different devices. Example of such applications include Zoom, Microsoft Teams and Discord.
- Servers: Serves handles the flow of media between the participants, the call setup, bandwidth management and security protocols.
- Network Infrastructure: A stable and reliable network connection is essential for a smooth video conference. The system relies on the internet or dedicated communication lines to transmit audio and video data between participants.

2.2.1 WebRTC and ICE Candidates

Web Real-Time Communication (WebRTC) [15], [16] is an open-source protocol that provides Real-Time Communication (RTC) [17] to web browsers, mobile devices, and other applications via API. It supports video and audio data to be sent between different devices, allowing developers to build video-communication solutions.

- Media Stream API: Grants access to the camera and microphone for video chats and audio calls.
- RTCDataChannel API: Facilitates direct data transfer between peers.
- RTCPeerConnection: Establishes direct links between browsers or applications.

An Interactive Connectivity Establishment (ICE) candidate is a critical component in WebRTC connections:

- Represents configuration details needed for communication.
- Includes IP addresses, port numbers, and transport protocols.
- Peers negotiate and agree upon the best candidate.
- Chosen candidate's details are used to set up the connection.

2.2.2 Kurento

The Kurento Media Server (KMS)[18]) is a WebRTC Media Server, it allows for media transmission, processing, recording, and playback, with a more flexible processing and control of the media compare to others. The control of the media functions as modules connecting each other to form a pipeline where the media enters, is processed and recorded, and then transmitted to the other users.
Kurento provides building blocks such as WebRTC senders and receivers, audio/video mixers, media recording, and more. These Media Elements are self-contained objects, called modules, that hold a specific media capability; they are extremely easy to compose by inserting, activating, or deactivating them at any point in time, even when the media is already flowing.
Kurento controls the flow of media, called in the system by Media Elements depending on the type of media, with a so-called Media Pipeline, effectively forming a customized architecture that can alter and transfer media as needed. Several built-in modules are provided for group communications, transcoding of media formats, and routing of audiovisual flows.
Alternatives to Kurento are OpenVidu and the Zoom SDK. Kurento was chosen as the core for the video-communication component of this dissertation since is the more versatile and customizable for the system requirements.

2.3 Video Processing

Video processing is a branch of signal processing that deals with the manipulation, analysis and enhancement of video data to improve its quality, extract useful information, and enable various applications. It involves a series of algorithms and techniques that work on individual video frames or sequences of frames to achieve specific objectives. Video processing plays a crucial role in various fields, including multimedia, computer vision, surveillance, entertainment, and video communication.
The most common library to use video processing is FFMPEG[19], a free open source software project that offers many tools for video and audio processing. It supports all media formats and is also highly portable. FFMPEG is designed to run on a command line interface so in this dissertation it is used a library in python called PyAv[20] that works as an interface to the FFMPEG.

CHAPTER 3

Use Case and Requirements

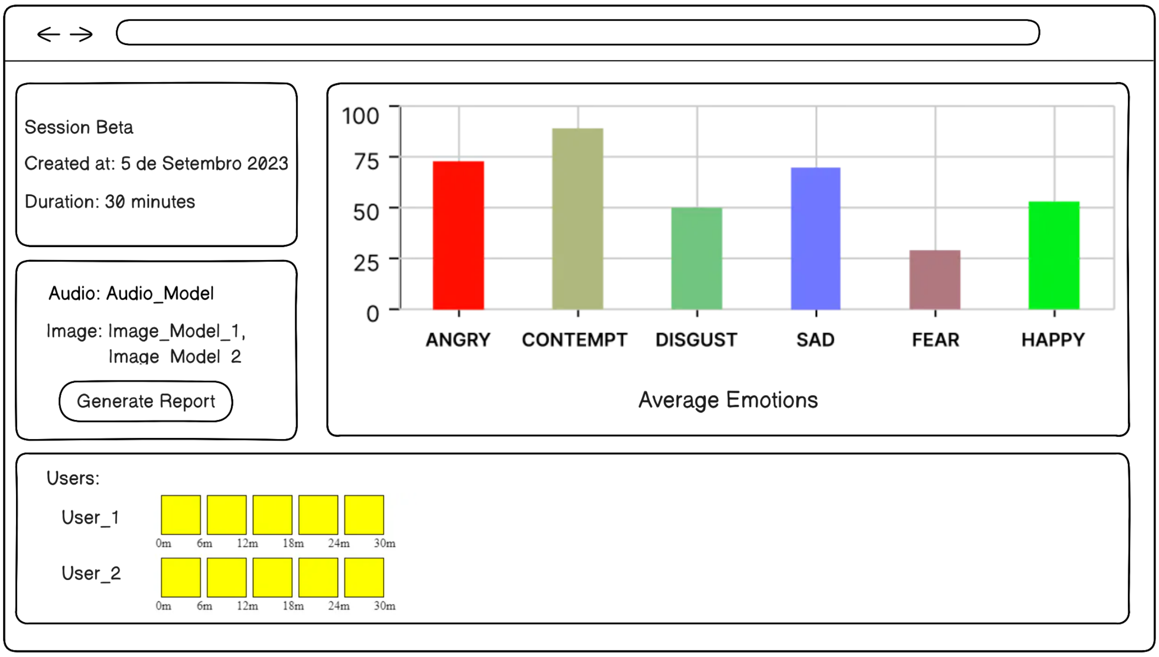
3.1 The Use Case

This section of the dissertation delves into a practical use case that exemplifies the real-world application of the platform in the context of workplace dynamics. In this scenario, a Team Leader assumes the central role in utilizing the system. The primary goal of this use case is for the Team Leader to start a video conference session, invite the members and proceeded with the session as normal, after the session analyze the report on the emotion analysis generated by the system. To accomplish the aforementioned goal, the Team Leader follows a structured series of procedural steps, as outlined below::

- Team leader logs into the system using their credentials and is presented with the dashboard of the web-based interface.
- Team leader navigates to the session creation section and creates a new video conference session for the team members.
- Team leader invites the team members to join the video conference session by sharing the session link
- The video conference session takes place, allowing the team members to communicate and collaborate remotely.
- During the video conference session, the system collects video and audio recordings to capture the emotional states of the team members.
- Once the video conference session concludes, the system analyses the data collect and generates an emotion analysis report, compiling the results of the analysis performed on the collected data.
- Team leader accesses the web-based interface and locates the session for which they want to analyze the emotion analysis report.

- Team leader opens the emotion analysis report associated with the session, which includes visualizations and insights about the emotional dynamics observed during the video conference, in a web format and a pdf format.
- Team leader examines the visualizations, such as timelines, charts, and graphs, that depict the emotional fluctuations and distribution of emotions.

In summary, the use case highlights how the system enables team leaders to create and conduct video conference sessions, collect data, and generate emotion analysis reports. By analyzing these reports, team leaders can gain valuable insights into the emotional dynamics within their teams and take appropriate actions to improve the workspace environment, fostering better communication, collaboration, and overall well-being. A draft example of the report in both formats is provided in the Figures 3.1 3.2



Session Beta

Created at: 5 de Setembro 2023

Duration: 30 minutes

Audio: Audio_Model

Image: Image_Model_1, Image_Model_2

Figure 3.1: Report Draft for Web View

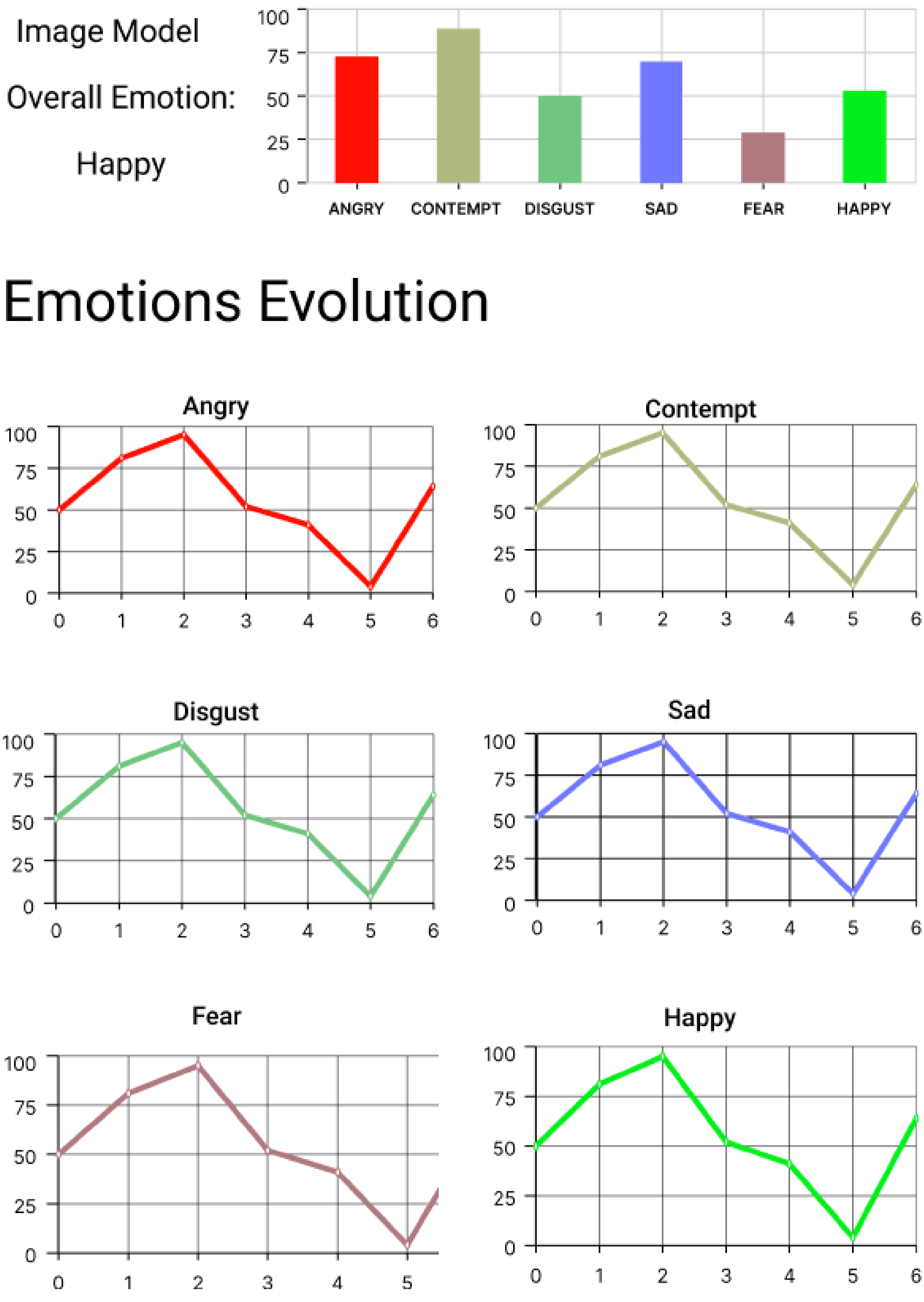


Image Model

Overall Emotion:

Happy

Emotions Evolution

Figure 3.2: Report Draft for PDF View

3.2 Requirements

Given the usage scenario is important to understating the requirements and needs of this product. In this section we will identify and document the functional and non-functional requirements for the pipeline.

3.2.1 Functional

Functional requirements define the capabilities and behaviors expected for the system. In our system there are a few key functionalities that need to be available for the user.

- Authentication System - the users require authentication and corresponding roles.
- Web interface - the users connect to the video conference system through a web site on the browser.
- Conference rooms - each video conference session is identified by a room. Users can only communicate with other users from the same room.
- Video and audio displays - users in the same room will ear and see other users.
- Results display - each video conference session has a report page available to be seen by the creator of the session

3.2.2 Non-Functional

Non-functional requirements describe how the system should behave, rather than the specific functions it should provide.

- Simple Web interfaces - the system needs to provide simple and easy to use interfaces
- Simple and Rich graphical dashboards - the system should display its results in easy to read and detailed graphics
- Low Delay - the system needs to ensure the results are provided with minimal delay
- High Accuracy - the system needs to provided results according with the expectation
- Extensibility - different machine learning components need to be easy to integrate with the system

CHAPTER 4

System Architecture

4.1 Overview

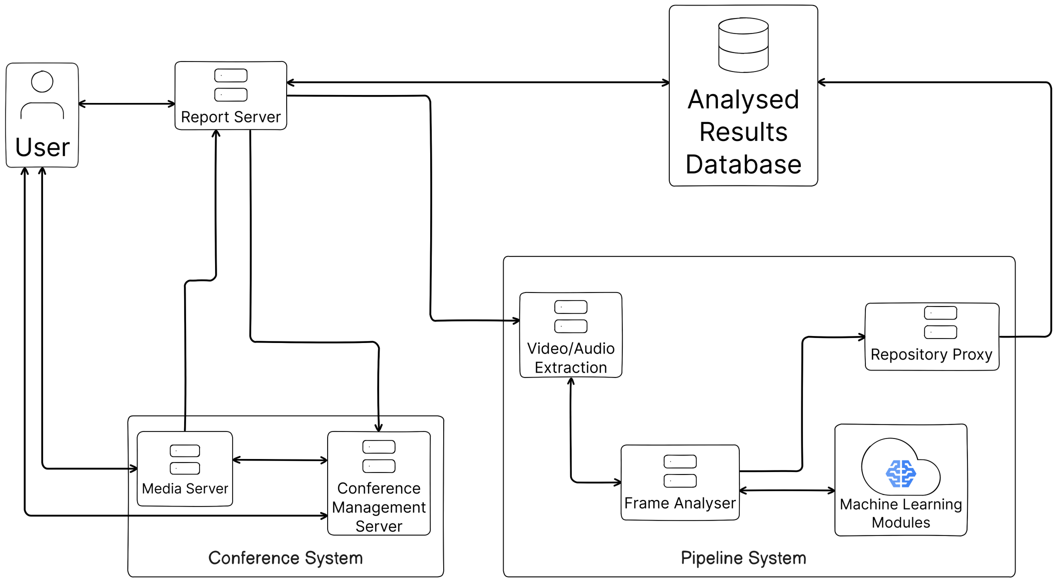


Figure 4.1: System Architecture

The system architecture consists of three main components: the Conference System, the Report Server, and the Pipeline System. The Conference System is responsible for providing video and audio calls to a group of users. It is composed of a media server and a Manager Conference Server. The Report Server connects users to the Conferences and saves the information of witch conferences the user participate. It also receives each video recording of each user in each conference, provided by the media server, and sends it to the Pipeline System to be analyzed. The Pipeline System extracts video and audio from the recordings of the users, analyzes the emotion state of each frame of video and audio using machine learning models, and finally saves the results in a database. Users can then access a report with the results from the server.

4.2 The Architecture

Report Server

This is the way user interacts with the system. The application is compose by 2 parts, the conference management and the report generation, both provided by a web server.

The conference management provides ways to create and join a video conference session. When a session is created, the user is redirected to the Conference System and when it leaves the server is notify of its exit. After the user leaves the session, their respective video record is send to the server to be redirected to the pipeline, alongside information of the user and the session. The server keeps track of the analyses progress in the pipeline.

After all the users had their records analysed, a report is generated and cached in the server to be available to the user when requested. This record has two forms, a web view and a pdf file, some information is only available in the web view.

Conference System

This part of the system functions by combining a media server and management server. The media server allows the connections and transmission of video and audio between the users of a group call or conference session and records the video for each user. The Management server manages the users inside the session, announcing to users on the entry or exit of other users and communicates with the media server.

Pipeline

This is the main part of the system, it receives a video as input and outputs to the Real-Time Database the results of emotion analyses of each audio and image of the input video. Each step of the pipeline can process on video coming from different session, the information of witch session and user the step is processing is provided on the messages that are distributed on the pipeline.

The pipeline is divide in 3 steps, extraction, analyses and repository respectively.

Starting on the extraction, it receives a video from the Report Server and the information of the begin of the session, the session id and the user id of the user present in the video and when the user joined the session. From the video it is extracted, in parallel, the video and audio frames. Video frames have a 33 milliseconds interval between each other. Audio frames are combine to form approximately 6 seconds chunks of audio.

In the analyses step, each frame is received and sent to all machine learning models for analyses. The results return by the models are organized and sent to the repository alongside the information of the user and session.

Reaching the repository the results are then save in the Analyse Results Database to be query in the future.

CHAPTER 5

Implementation

5.1 Implementation Overview

For the development of the system several technologies was used. In this chapter we will start by going over each component, explore the technology used and how each component interacts with one another, following after for a more detail view of the implantation in each component.
Starting in the Report Server, the technologies used are Python language and the Django Framework[21] including libraries to communicate with the Pipeline and to query from the Database, Thespian and Influx respectably. A MySql Database[22] is used to save information of the sessions created and users register. The python library Reportlab[23] is the library use to generate the reports since it allows to generate a PDF content by code. The communication with the Conference System is through a REST[24] API.
In the Conference System, Kurento is the media server implemented and the Management Server uses the Java SpringBoot framework[25] with a Web Controller proving a web page for the conference session and comunication with Kurento media server.
Concluding with the Pipeline, this is implemented with the Thespian python library and a Influx Time-Series Database to save the data. The Machine Learning models are serve by an FastApi[26] server to simulate a Emotion Classification Service. The pipeline developed uses the Actor Model[27] implementation.
The actor model in computer science treats an actor as the basic building block of concurrent computation. An actor, in this context, is a primitive unit of computation. When it receives a message, it can start local process, create more actors, send more messages and determine how to respond to the next message received. All interaction in the actor model is asynchronous and autonomous. This implementation was chose since it allows to asynchronous run process of extracting and analysing and in different machines, reducing the time needed to process and analyse a video. The Thespian is what allow us to use this implementation in the system FastApi fast and lightweight web framework for Python allowing the easy implementation of an API to call each of the Machine Learning Models,

InfluxDB[28] is an open-source time series database, it is designed for high-speed storage and retrieval of time series data. Flux is a lightweight scripting language for querying InfluxDB databases and working with data. Flux is designed to accommodate a wide array of data processing and analytical operations. It provides an SQL-like language with built-in time centric functions for querying a data structure composed of measurements, series, and points. A Docker[29] environment is used to run the system with containers for each of components and sub components of the system. The docker compose file is organized as follow:

- Databases:
  - InfluxDb
  - MySql
- Report Server
- Conference:
  - Kurento Media Server
  - Conference Management Server
- Pipeline:
  - Extraction
  - Analyses
  - Repository
- Models$_{Server}$

5.2 Report Server

This web application is divided in three main services: users, conference and report. As the name suggest the users provides simple authentication system with registration, login and logout features.
The conference service manages sessions, its creation, witch users participate in the session and state of the session. This services provides the following REST endpoints:

- 'session/new' - creates a conference session and redirects user to Conference Management Server
- 'session/join' - redirects user to Conference Management Server
- 'session/leave/<uuid:session_id>/' - removes user from the session and redirects to home page
- 'session/upload/<int:user_id>/<uuid:session_uuid>/' - uploads the record video of a user in the session and sends it to the Analyse pipeline

When a session is created it also creates a entry in the database with the follow structure:

- 'uuid' - unique id for the session
- 'title' - title of the session
- 'start_time' - date of session start
- 'end_time' - date of session end
- 'users' - list of users that were present

A Session is created by an User with permission to create Sessions. This permission is determined by a tag in the User named session_admin. A Session can only be created by a session_admin but does not exclude other admins to participate in a Session. The database entry that connects an User to a Session is the SessionEntry, it has information on the time the User joins the session and if that User is the session creator. In the case the User is not a session_admin, the system only presents them with a form to join a Session, asking the title of the Session as input. After a User leaves a Session or this one is finalized, their audio and video recording is send to the server to be redirected to the pipeline, as explain in the sections above.

The Report service provides a page with a dashboard of reports of all the session created by the User. Each item present on the list contains information of session, like title, duration and date of creation, the most relevant emotion and the emotions average distribution during the Session. Each item also contains a button to redirect the user to the report page and a button to retrieve the pdf report version.

When the server is inform that the analyses is complete proceeds to generate the report data, caching it in the server. This way when the client requests the report data from the server the report is already ready to be seen. To generate the report, data is query from the Influx database, queries are made to all the different ML models, it is then extracted the average emotion distribution, a average of the most predominant emotion of each user in small time intervals, example average of the emotion happy in each 5 minutes of the Session, each emotion distribution during the session and all the results of the analysing frames. Respectively, this data is display in a bar, table, line and radar chart, witch the radar chart is animated and shows the emotion state in all the frames analyse.

5.3 Conference System

The Management Server is composed by a websocket handler, a rest controller, an user registry and a room manager. The rest controller only offers a single GET endpoint that receives a session id and name and a user id and name. This endpoint only function is to provide a page for the user and get the context of the room the user is aiming to join or create, all the management between users and sessions rooms are made through websocket messages. In this page a websocket connection is created follow by a message to the server registering the user to the room and to.

The websocket handler process the following messages sended by the client:

- {registerUser} - adds user to a room
- {leaveRoom} - removes a user from a room

The registerUser message adds an User to a room and provides the necessary information to do so, the roomId and userId. The room instance is obtain through the room manager and the user is added to the room, a room instance is created if it did not already exist. This creates a UserSession instance that is saved in the UserRegistry. All other users in the room receive a message from the server informing of this user arrival and start trading the messages start the streaming media from and to the user.

When a room instance is created a Kurento pipeline is instantiated, but not initialize, this pipeline will manage all media from users present in the room. On the creation of the UserSession instance it is where the Kurento pipeline is initialize by adding two modules to the pipeline, the WebRTC and the Recorder. The pipeline begins with a WebRTC module witch the user connects to using the browser WebRTC apis. The WebRTC module is then connected to a Recorder module and to any WebRTC module of other users present in the room. The WebRTC module input/output endpoint provides media streaming for RTC has WebRTC technology to communicate with browsers, it streams the media it receives to other modules. The Recorder module receives a media stream as input and stores its content in a file.

A client that sends the leaveRoom message is removed from the room. His UserSession instance is deleted and the Kurento modules that it connects are deleted from the pipeline. All other users in the room receive a message from the server informing of this user exit and start trading the messages to stop the streaming media from and to the user.

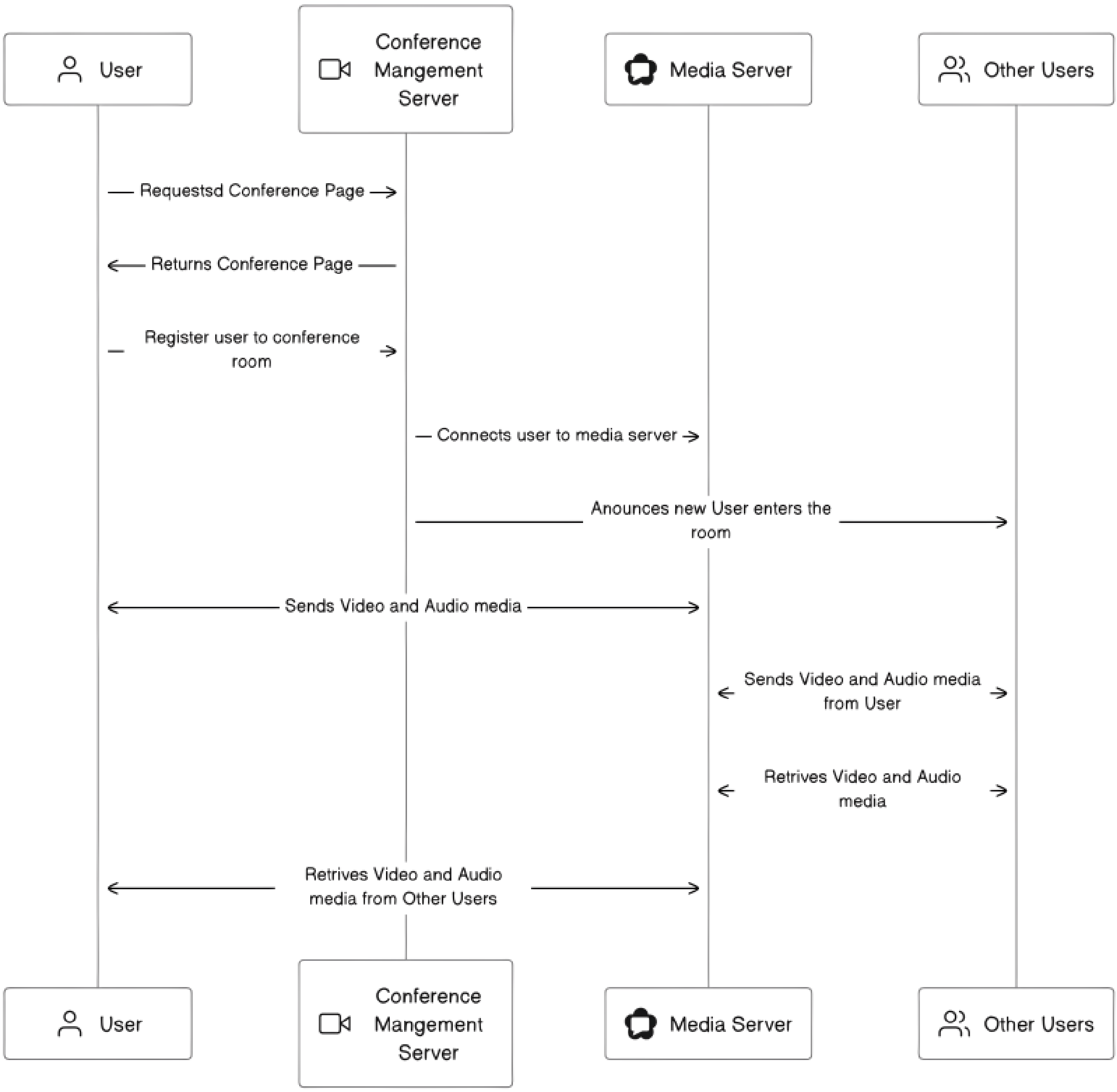

User
Conference Mangement Server
Media Server
Other Users
Requestsd Conference Page
Returns Conference Page
Register user to conference room
Connects user to media server
Anounces new User enters the room
Sends Video and Audio media
Sends Video and Audio media from User
Retrives Video and Audio media
Retrives Video and Audio media from Other Users
User
Conference Mangement Server
Media Server
Other Users

Figure 5.1: Sequence Diagram of the messages in a conference session

5.4 Pipeline System

5.4.1 Overview

Each one of the three steps in the pipeline is a Python thread called an Actor that are location independent and communicates with other Actors via messages. This allows the system to have each step run in different servers with the hardware and software necessary to run each step as efficient as possible.
Actors communicate with each other using Transmission Control Protocol (TCP)[30]. In order for the transmission to be possible, Actors need to know the IP Address (Internet Protocol address) of others Actors. This is provided by the Convention Leader. The Convention Leader works as a central node of information by responding with the IP Address of an Actor when it is requested. The Convention Leader then allows Actors to operate in different systems and networks for a more efficient way of processing and analysing the video. Each actor also

changes the number of processes in parallel depending on the flow of requests received in order to process faster.

- {USER_ID}
- {SESSION_ID}
- {timestamp}
- {data}
- {last}

USER_ID and SESSION_ID identifies the user and the session where the video was recorded analyse. The timestamp indicates the point in time of the extract audio and image frames in the video. The data is the resulting data processed in a pipeline step and to be processed by the next step. Finally the last is a tag used to notify the System Web Application Server that this step is completed. The sequence of operations is demonstrated in the Figure 5.2

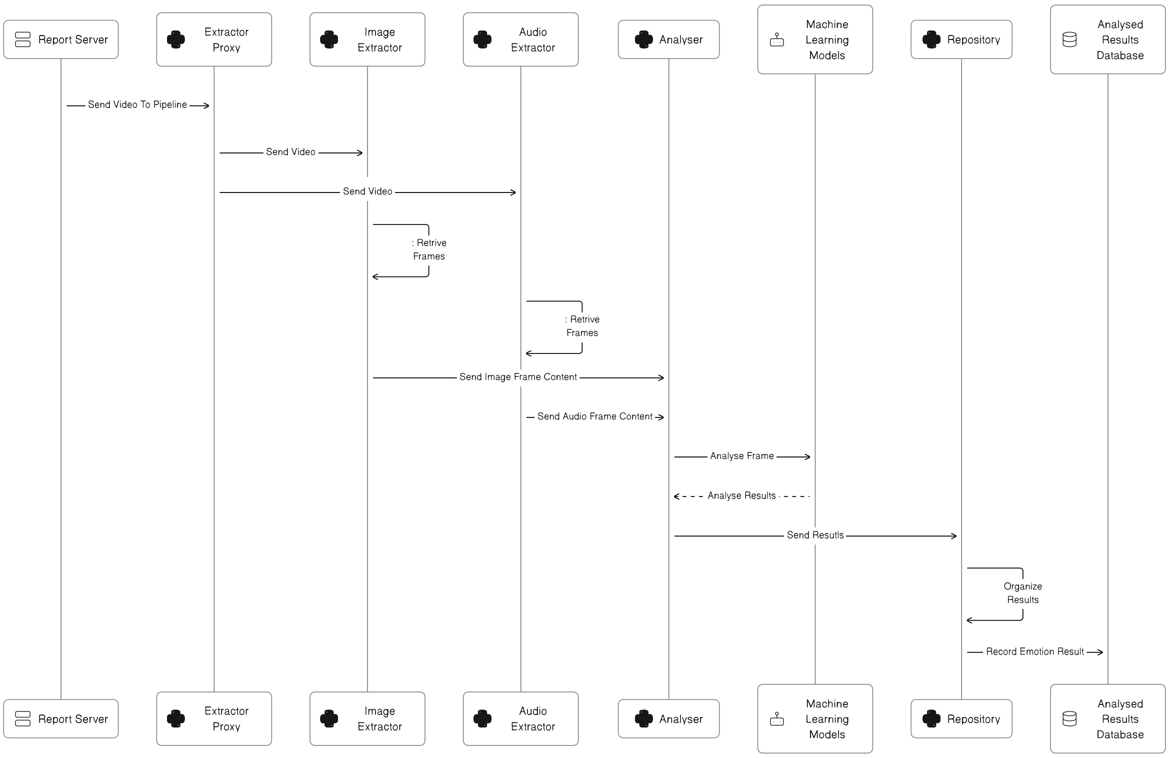


Figure 5.2: Sequence Diagram of the Processing Stages in the Pipeline

5.4.2 Video and Audio Extraction

This step is responsible for extracting images frames and audio samples from the video. It is composed of two actors, one for the images and the other for the audio, each one working in parallel.

The extraction starts by creating a media container where it's define which media stream we want to access and process. The method of extraction differences with the media chosen. Images frames, separated by thirty-three milliseconds of each other, are retrieved from the

video stream and their binary data are saved in a jpeg file-like object. In the audio extraction, it is retrieved small samples of around six seconds of sound and saved in WAV file-like object with 48.0 kHz of sample rate and 16 Bit depth, ressample occurs if needed. The start time position in the video of each sample is also saved. When a frame or audio sample is extracted and processed, it is then forwarded to the next step of the pipeline with the timestamp of the frame in the video. In the case of the audio sample the timestamp corresponds to the point where the six seconds interval occurs in the video.
The image 5.3 represents the extraction process.

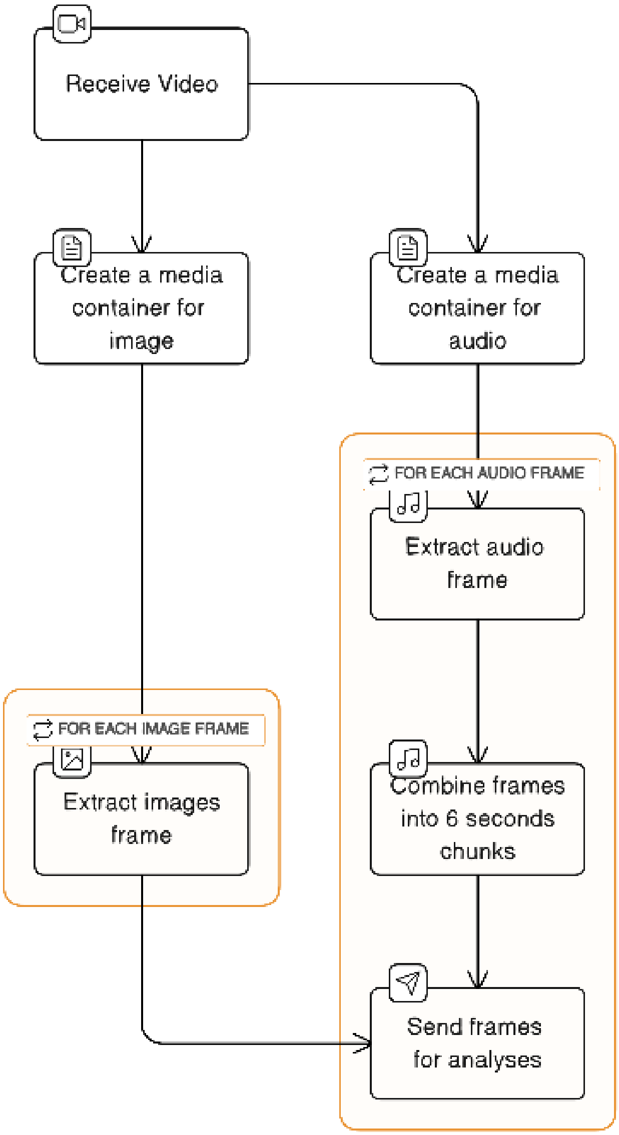


Figure 5.3: Sequence Diagram of the Processing Stages for Video and Audio Extraction

5.4.3 Frame Analyzer

The role of this step is to receive the image and audio frame and make a classification of this segment in terms of emotion data. To do this the Emotion Analyses REST API is called by sending the respective segment and await the analyse of all the models for that particular type of media.

Emotion Analyses API

This is a small REST containing the different machine learning models used by the system. For the audio analysis a Speech Emotion Model and for the image analysis two different iterations of a DEEPFACE[31]. The result given by the model return to the pipeline with the given structure below, where each emotion is represent by value between 0 and 1, signifying the probability of being the correct emotion represented in the frame segment.

- {MODEL}
  - Surprise
  - Happy
  - Neutral
  - Contempt
  - Sad
  - Disgust
  - Fear
  - Angry
- {SUCCESS}

5.4.4 Repository Proxy

This is the last step where the results of the analyses of the frame are save in the time-series database. For each emotion result provided a record is created in the database representing a point in time. The combinations of this records gives an evolution of the emotion along the video allowing the visualization of the emotion at the point in time and how much each emotion appears in the video. The record saved contains the following fields:

- {EMOTION}
- {PERCENTAGE} - probability of the emotion being represented in the frame
- {ALGORITHM} - algorithm used in the analyses
- {TIME} - time where the frame is position in the video
- {SESSION_ID} - session where the video was recorded
- {USER_ID} - user represented in the video
- {SUCCESS} - if the analyses was successful
- {FRAME_TYPE} - type of media of the frame (either a Image or a Audio chunk)

CHAPTER 6

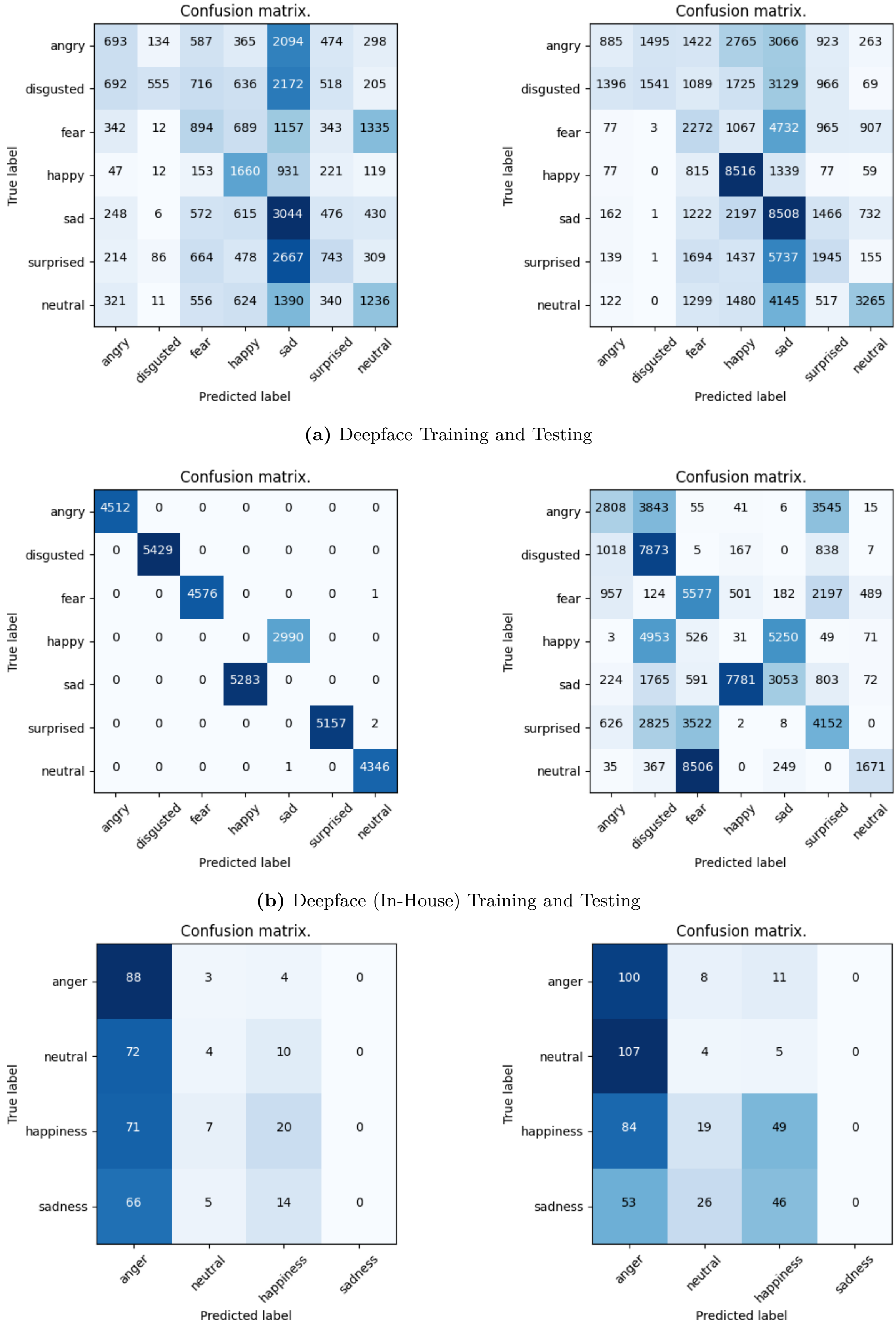
Validation and Results

6.1 Emotion Validation

In order to validate the system develop it was used several videos, already classified, to simulated different sessions on the system. These videos were part of a dataset called MEAD [32], also use in the training and validation of the machine learning models used by the system, each video has approximately 6 seconds framing one person in front of the camera.
For the validation of the machine learning models it was calculated the percentage of fail classification attempts, wrong and right classifications, it is also provided the confusion matrix of each models. This results are were retrieved for all the models and calculate for the training and validation part of the dataset. The results are present in Table 6.1 and Figure 6.1

Model	Dataset	Face/Audio not Detected	Incorrect Classification	Correct Classification
Deepface	Traning	3.77%	70.56%	25.67%
Deepface	Testing	26.51%	48.07%	25.42%
Deepface (In-House)	Traning	6.06%	24.07%	69.86%
Deepface (In-House	Testing	26.97%	49.28%	23.75%
Audio (In-House)	Traning	0.27%	30.68%	69.04%
Audio (In-House)	Testing	0.58%	29.71%	69.70%

Table 6.1: Percentage Table



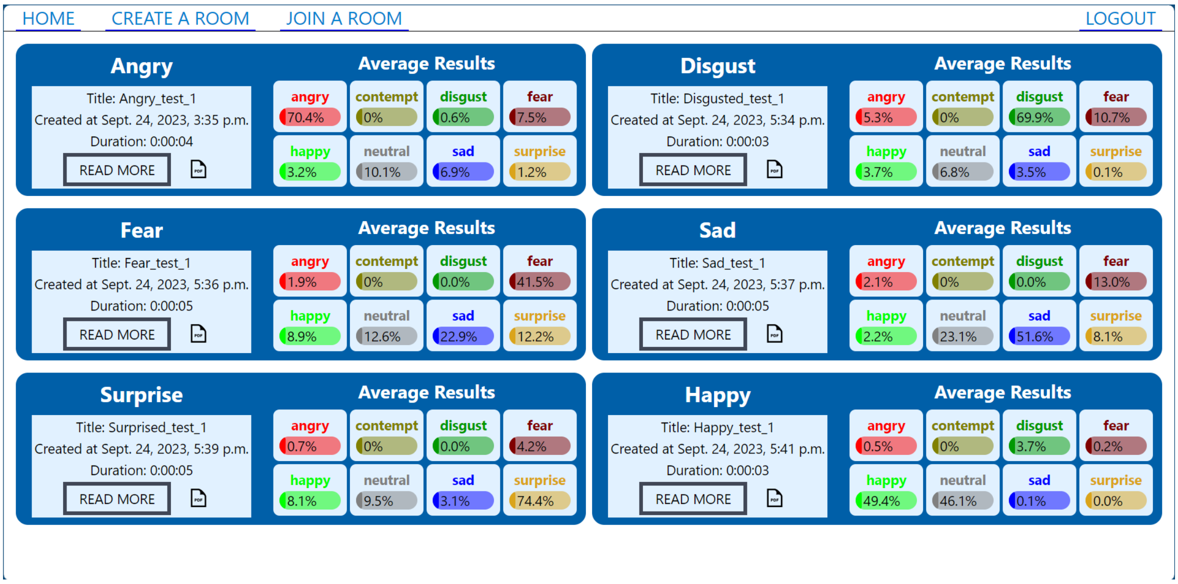
(a) Deepface Training and Testing

(b) Deepface (In-House) Training and Testing

(c) Audio (In-House) Training and Testing

Figure 6.1: Confusion Matrix for the Machine Learning Models

For validation of the system we select videos from different users representing an emotion in order to simulated a small session. Each session is of length three to five seconds and contains two different users. The results can be see in 6.2.
Each session has classify correctly, however we can detect a lot of variation in the points forming the emotion evolution charts. This is because of the high amount of frames being analise and represent in the chart, the emotion classify in a frame not always means that the next or previous frame will be or was classify with the same emotion. So we get charts that do not have a smooth form. This could be improve if we analise less frames or represent in the charts less results, for example it could we could use the average or median result instead.



Angry

Title: Angry_test_1
Created at Sept. 24, 2023, 3:35 p.m.
Duration: 0:00:04

Average Results

angry	contempt	disgust	fear
70.4%	0%	0.6%	7.5%
happy	**neutral**	**sad**	**surprise**
3.2%	10.1%	6.9%	1.2%

Disgust

Title: Disgusted_test_1
Created at Sept. 24, 2023, 5:34 p.m.
Duration: 0:00:03

Average Results

angry	contempt	disgust	fear
5.3%	0%	69.9%	10.7%
happy	**neutral**	**sad**	**surprise**
3.7%	6.8%	3.5%	0.1%

Fear

Title: Fear_test_1
Created at Sept. 24, 2023, 5:36 p.m.
Duration: 0:00:05

Average Results

angry	contempt	disgust	fear
1.9%	0%	0.0%	41.5%
happy	**neutral**	**sad**	**surprise**
8.9%	12.6%	22.9%	12.2%

Sad

Title: Sad_test_1
Created at Sept. 24, 2023, 5:37 p.m.
Duration: 0:00:05

Average Results

angry	contempt	disgust	fear
2.1%	0%	0.0%	13.0%
happy	**neutral**	**sad**	**surprise**
2.2%	23.1%	51.6%	8.1%

Surprise

Title: Surprised_test_1
Created at Sept. 24, 2023, 5:39 p.m.
Duration: 0:00:05

Average Results

angry	contempt	disgust	fear
0.7%	0%	0.0%	4.2%
happy	**neutral**	**sad**	**surprise**
8.1%	9.5%	3.1%	74.4%

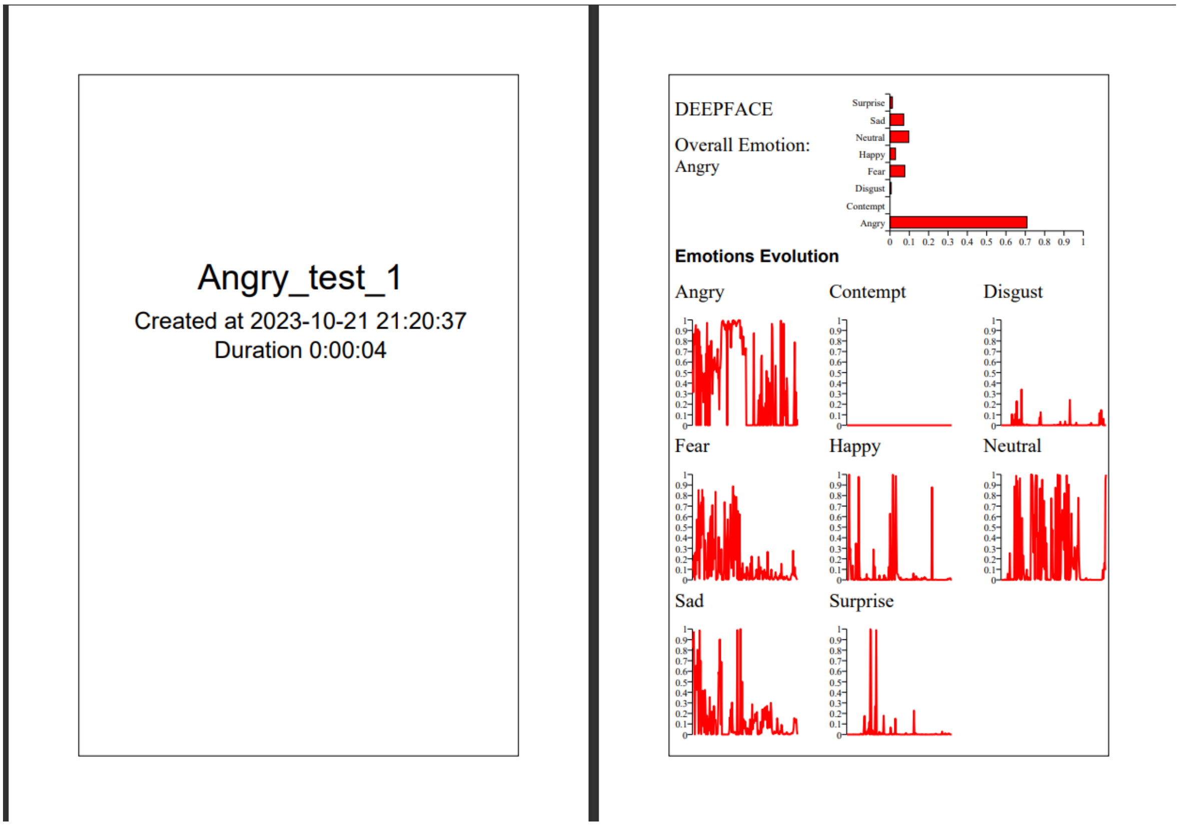
Happy

Title: Happy_test_1
Created at Sept. 24, 2023, 5:41 p.m.
Duration: 0:00:03

Average Results

angry	contempt	disgust	fear
0.5%	0%	3.7%	0.2%
happy	**neutral**	**sad**	**surprise**
49.4%	46.1%	0.1%	0.0%

Figure 6.2: Emotion Validation in System



Angry_test_1

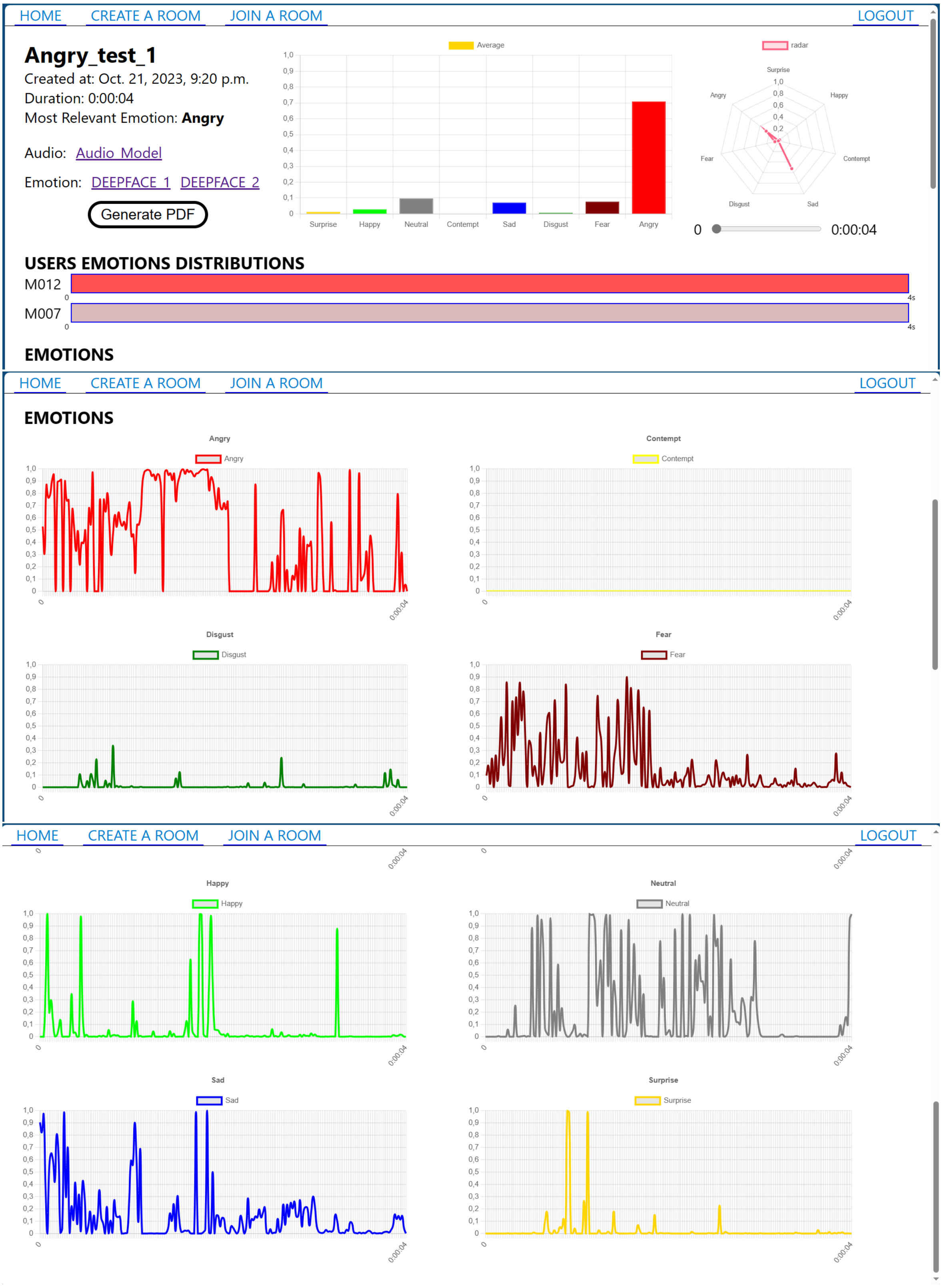
Created at 2023-10-21 21:20:37
Duration 0:00:04

DEEPFACE

Overall Emotion:
Angry

Emotions Evolution

Figure 6.3: Report PDF View



Angry_test_1

Created at: Oct. 21, 2023, 9:20 p.m.

Duration: 0:00:04

Most Relevant Emotion: **Angry**

Audio: Audio Model

Emotion: DEEPFACE 1 DEEPFACE 2

USERS EMOTIONS DISTRIBUTIONS

M012

M007

EMOTIONS

(a) Result Web Page View

6.2 Performance Validation

To validate the performance of the system we calculate the time it takes to analyse, generate and present the results of different conference sessions, all the test are from the same user and emotion and were test in a Windows Computer with 16Gigas of RAM and a 12th Intel Core i7 CPU. The results of this tests can be seen in the Table 6.2.

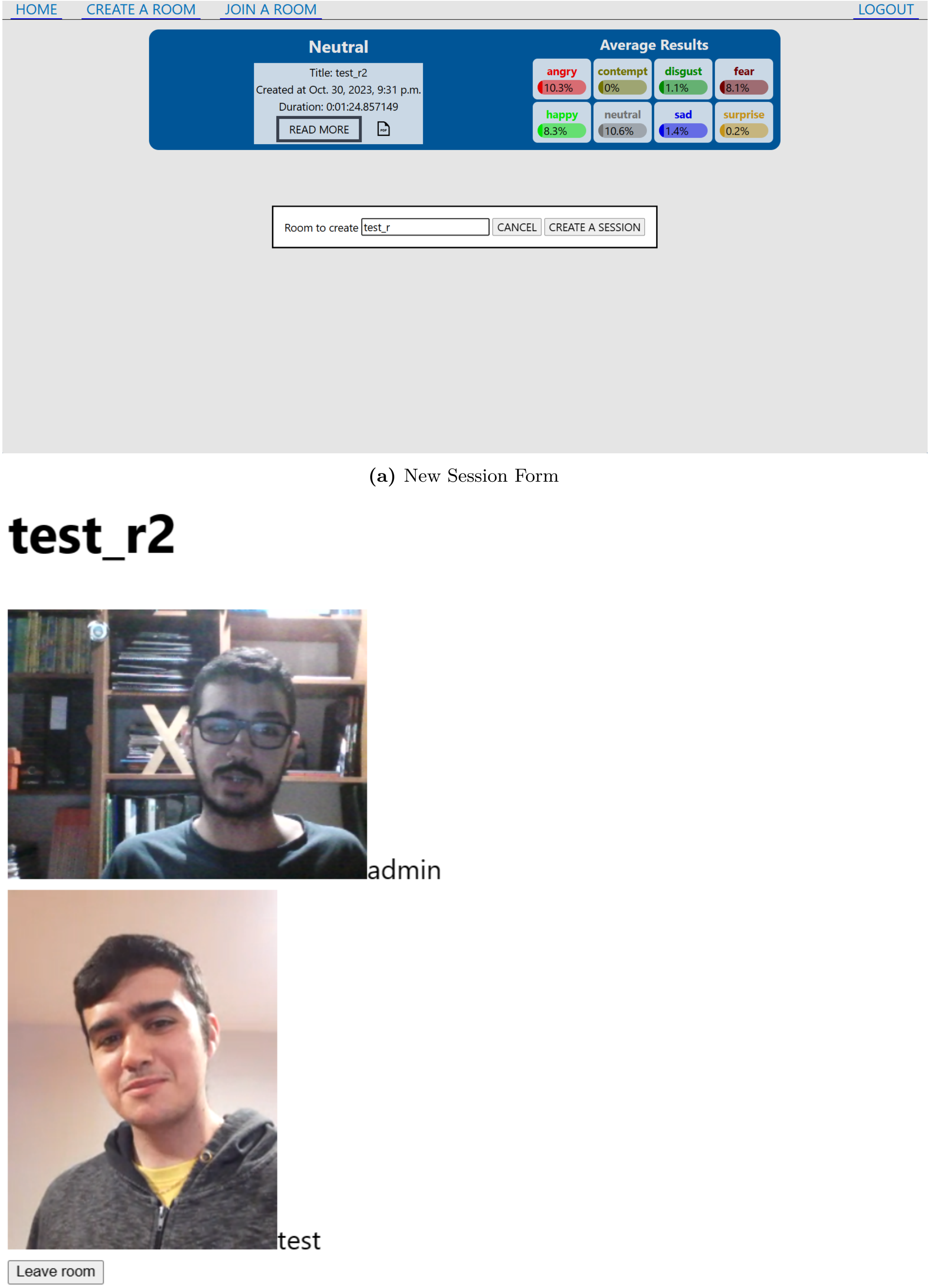
It can be observe long time execution in the Analyse and Extraction of Image frames, this derives from the high amount of frames being analyse, the computer were the system is running and the fact that the models use are not optimize. This is the a major problem of the system since the report can not be generate unless all the frames are extract and analyse, and so the system may collapse because a extreme number of frames being accumulated because they are not processed in time. This should be a focus to address when exploring this project in the future.

Video Length	Media	Number of Frames	Extraction Time	Analyse Time
15 minutes	Audio	139	5 seconds	32 seconds
15 minutes	Image	27000	6 minutes and 55 seconds	2 hours 14 minutes and 11 seconds
30 minutes	Audio	277	10 seconds	53 seconds
30 minutes	Image	54000	15 minutes and 56 seconds	4 hours 7 minutes and 46 seconds
1 hour	Audio	554	19 seconds	1 min 53 seconds
1 hour	Image	108000	28 minutes and 50 seconds	8 hours 5 minutes and 25 seconds

Table 6.2: Results Time Performance

6.3 Use Case Results

In this section a sequence of images will show the process a user has to do to fulfill the Use Case mention in early 3.



Neutral

Title: test_r2
Created at Oct. 30, 2023, 9:31 p.m.
Duration: 0:01:24.857149

Average Results

angry	contempt	disgust	fear
10.3%	0%	1.1%	8.1%
happy	neutral	sad	surprise
8.3%	10.6%	1.4%	0.2%

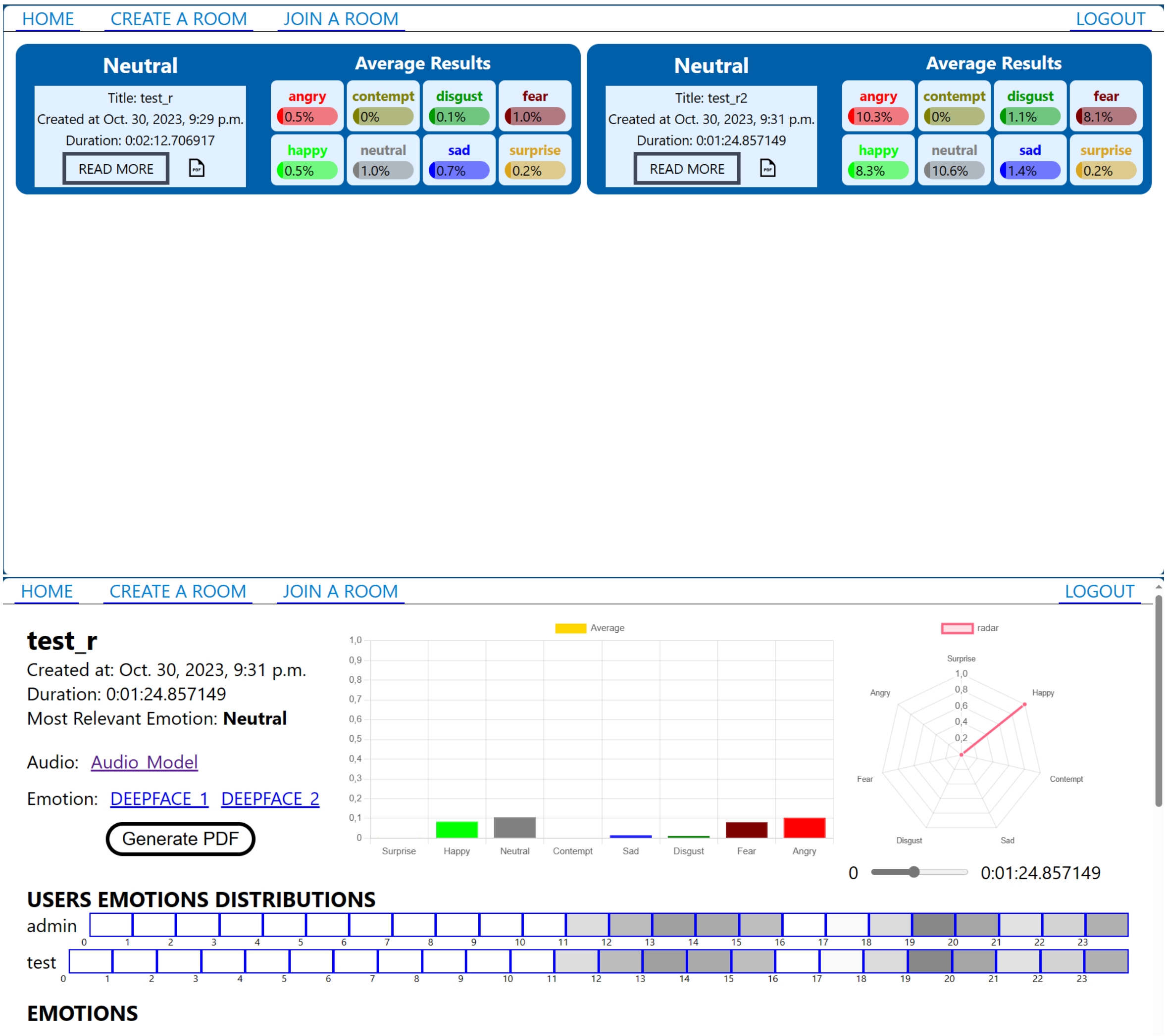
Room to create test_r

(a) New Session Form

test_r2

(b) Video Session

Figure 6.5: Creation of a new Session



HOME CREATE A ROOM JOIN A ROOM LOGOUT

Neutral

Title: test_r
Created at Oct. 30, 2023, 9:29 p.m.
Duration: 0:02:12.706917

READ MORE

Average Results

angry	contempt	disgust	fear
0.5%	0%	0.1%	1.0%
happy	**neutral**	**sad**	**surprise**
0.5%	1.0%	0.7%	0.2%

Neutral

Title: test_r2
Created at Oct. 30, 2023, 9:31 p.m.
Duration: 0:01:24.857149

READ MORE

Average Results

angry	contempt	disgust	fear
10.3%	0%	1.1%	8.1%
happy	**neutral**	**sad**	**surprise**
8.3%	10.6%	1.4%	0.2%

HOME CREATE A ROOM JOIN A ROOM LOGOUT

test_r

Created at: Oct. 30, 2023, 9:31 p.m.
Duration: 0:01:24.857149
Most Relevant Emotion: **Neutral**

Audio: Audio Model

Emotion: DEEPFACE 1 DEEPFACE 2

Generate PDF

USERS EMOTIONS DISTRIBUTIONS

EMOTIONS

Figure 6.6: Observe The Results

CHAPTER 7

Conclusion

7.1 Conclusions

In this dissertation we have described the development of a system to create video conference sessions, analyse and present the evolution of the emotional state of the users that participate in the sessions. For that three main tasks were require, a simple but robust video conference system, the machine learning models to analyse the emotion state of users from video and audio, and a pipeline that process the videos from the video conference, analyse them through the machine learning models and records the results presenting them to the users. In this work we preferred the use of house accessible machine learning models to overcome the restrictions associated to commercial services for video classification in the cloud In the early stages of developing, the main objective of the pipeline was to provided feedback in real time as well as the report generated at the end of the conference session. This idea was not follow through because it require the creation of a custom video conference system or the changes of one already existing, something that wasn't on the scope of the project. To finish, although the system had to change from their original idea, a good base system was develop and here presented.

7.2 Future Work

Although the system developed is a good base for the final objective of the project, a lot can be improve. Future work proposals are as follow:

- Extraction of image and audio could use a better strategy to either extract only the necessary frames to be analyse or improve the speed of extraction,
- Add and improved the machine learning models to return better results in a faster rate or alternatives in the detection of emotion, for instance movement and poses.
- Increase the ways the resulting data is visualize. This includes adding functionality to a real time display of results.

References

[1] B. C. Ko, "A brief review of facial emotion recognition based on visual information", *Sensors (Switzerland)*, vol. 18, 2 Feb. 2018, ISSN: 1424-8220. DOI: `10.3390/s18020401`.

[2] E. Union, *Regulating facial recognition in the eu*, May 2021. [Online]. Available: `https://www.europarl.europa.eu/RegData/etudes/IDAN/2021/698021/EPRS_IDA%282021%29698021_EN.pdf`.

[3] R. W. Picard, *Affective Computing.* The MIT Press, 1997.

[4] R. Reisenzein, E. Hudlicka, M. Dastani, *et al.*, "Computational modeling of emotion: Toward improving the inter- and intradisciplinary exchange", *IEEE Transactions on Affective Computing*, vol. 4, no. 3, pp. 246–266, Jul. 2013, ISSN: 1949-3045. DOI: `10.1109/T-AFFC.2013.14`.

[5] M. F. Alsharekh, "Facial emotion recognition in verbal communication based on deep learning", *Sensors*, vol. 22, 16 Aug. 2022, ISSN: 14248220. DOI: `10.3390/s22166105`.

[6] J. S. R. Arya and A. Kumar, "A survey of multidisciplinary domains contributing to affective computing", *Elsevier Science Publishers B. V. Netherlands*, p. 778, May 2021, ISSN: 1574-0137. DOI: `10.1016/j.cosrev.2021.100399`.

[7] R. B. Prameela Naga Swamy Das Marri, "Facial emotion recognition methods, datasets and technologies: A literature survey," *Materials Today: Proceedings*, vol. 80, 2023, ISSN: 2214-7853. DOI: `10.1016/j.matpr.2021.07.046`.

[8] Y. Khaireddin and Z. Chen, *Facial emotion recognition: State of the art performance on fer2013*, 2021. arXiv: `2105.03588 [cs.CV]`.

[9] A. S. Aljaloud, H. Ullah, and A. A. Alanazi, "Facial emotion recognition using neighborhood features", *International Journal of Advanced Computer Science and Applications*, vol. 11, 2020. [Online]. Available: `https://api.semanticscholar.org/CorpusID:211031549`.

[10] M. Jianhua, J. Hai, T. Y. Laurence, and J. T. Jeffrey, *Ubiquitous Intelligence and Computing.* Springer Berlin Heidelberg, 2006. DOI: `10.1007/11833529`.

[11] M. Farooq, F. Hussain, N. K. Baloch, F. R. Raja, H. Yu, and Y. B. Zikria, "Impact of feature selection algorithm on speech emotion recognition using deep convolutional neural network", *Sensors*, vol. 20, no. 21, 2020, ISSN: 1424-8220. DOI: `10.3390/s20216008`. [Online]. Available: `https://www.mdpi.com/1424-8220/20/21/6008`.

[12] Y. H. Z. Yang, "Algorithm for speech emotion recognition classification based on mel-frequency cepstral coefficients and broad learning system", *Evolutionary Intelligence*, 2022, ISSN: 1864-5917. DOI: `10.1007/s12065-020-00532-3`.

[13] S. Narayanan and P. G. Georgiou, "Behavioral signal processing: Deriving human behavioral informatics from speech and language", *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1203–1233, 2013. DOI: `10.1109/JPROC.2012.2236291`.

[14] J. Rashid, Y. W. Teh, F. Hanif, and G. Mujtaba, "Deep learning approaches for speech emotion recognition: State of the art and research challenges", *Multimedia Tools and Applications*, vol. 80, 2021, ISSN: 1573-7721. DOI: `10.1007/s11042-020-09874-7`.

[15] G. for Developers. "Web real-time communication". (2023-09-17), [Online]. Available: `https://webrtc.org/`.

[16] B. Sredojev, D. Samardzija, and D. Posarac, "Webrtc technology overview and signaling solution design and implementation", in *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2015, pp. 1006–1009. DOI: 10.1109/MIPRO.2015.7160422.

[17] W. contributors. "Real-time communication for the web". (2023-10-17), [Online]. Available: https://en.wikipedia.org/wiki/Real-time_communication (visited on 10/17/2023).

[18] K. Technologies. "Kurento". (2020), [Online]. Available: https://doc-kurento.readthedocs.io/en/latest/.

[19] F. Bellard. "Ffmpeg library for media processing". (2023-11-10), [Online]. Available: https://www.ffmpeg.org/about.html (visited on 12/10/2023).

[20] M. Boers. "Pythonic bindings for ffmpeg's libraries." (2017), [Online]. Available: https://pyav.org/docs/stable/index.html#overview (visited on 09/17/2023).

[21] D. S. Foundation. "Django". (2023), [Online]. Available: https://www.djangoproject.com/.

[22] Oracle. "Mysql". (2023), [Online]. Available: https://www.mysql.com/.

[23] I. ReportLab. "Reportlab". (2023), [Online]. Available: https://docs.reportlab.com/.

[24] "Restapi". (2023), [Online]. Available: https://www.ibm.com/topics/rest-apis.

[25] "Springboot". (2023-10-30), [Online]. Available: https://spring.io/projects/spring-boot.

[26] S. Ramírez. "Fastapi". (2023), [Online]. Available: https://fastapi.tiangolo.com/.

[27] W. contributors. "Actor model". (2023-11-6), [Online]. Available: https://en.wikipedia.org/wiki/Actor_model.

[28] K. Quick. "Thespian". (2020-03-10), [Online]. Available: https://thespianpy.com/doc/.

[29] D. Inc. "Docker". (2023), [Online]. Available: https://www.docker.com/.

[30] W. contributors. "Transmission control protocol". (2023-10-29), [Online]. Available: https://pt.wikipedia.org/wiki/Protocolo_de_Controle_de_Transmiss%C3%A3o.

[31] S. I. Serengil. "Deepface". (2023-12-8), [Online]. Available: https://github.com/serengil/deepface (visited on 12/10/2023).

[32] W. Kaisiyuan, W. Qianyi, S. Linsen, *et al.*, "Mead: A large-scale audio-visual dataset for emotional talking-face generation", *ECCV*, 2020. [Online]. Available: https://paperswithcode.com/paper/mead-a-large-scale-audio-visual-dataset-for.