Universidade de Aveiro
2024

Eva
Bartolomeu

# Recuperação de memórias através de lifelogging usando a aplicação MEMORIA

# Memory retrieval through lifelogging using the MEMORIA application

Universidade de Aveiro
2024

**Eva Bartolomeu**

# Recuperação de memórias através de lifelogging usando a aplicação MEMORIA

# Memory retrieval through lifelogging using the MEMORIA application

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Informática, realizada sob a orientação científica do Doutor António José Ribeiro Neves, Professor associado c/ agregação do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e da Doutora Josefa das Neves Simões Pandeirada, Equiparado a Investigador Principal do Departamento de Educação e Psicologia da Universidade de Aveiro.

**o júri / the jury**

| | |
|---|---|
| presidente / president | Professor Doutor António Joaquim da Silva Teixeira<br>Professor Catedrático, Universidade de Aveiro |
| vogais / examiners committee | Professora Doutora Soraia Meneses Alarcão<br>Professora Auxiliar, Universidade de Lisboa - Faculdade de Ciências |
| | Professor Doutor António José Ribeiro Neves<br>Professor Associado C/ Agregação, Universidade de Aveiro |

**agradecimentos / acknowledgements**

Agradeço toda a ajuda e apoio que os meus pais e a minha irmã ofereceram não apenas durante este trabalho, mas sim em toda a minha vida, sem eles nada disto seria possível. Estou grata pelo meu companheiro Artur me motivar durante este percurso, nos momentos de menor confiança. Um obrigada aos meus colegas do MEMORIA, Alexandre, Bernardo, Doutor Ricardo Ribeiro e Pedro, que sempre me ajudaram quando tinha dificuldades. Aos participantes que realizaram a experiência desenvolvida nesta dissertação, agradeço pela sua contribuição e pelo tempo dedicado. Por fim, e não menos importante, agradeço ao meu orientador, Professor Doutor António Neves, pela orientação e prontidão que teve sempre comigo.

**Palavras Chave**

registo de vida, memória episódica, memória autobiográfica, recuperação de imagens, processamento de linguagem natural, análise semântica latente.

**Resumo**

Lifelogging é o processo de recolha, armazenamento e análise de dados multimodais pessoais, em formato digital. A dissertação integra-se no sistema lifelogging MEMORIA, um sistema já em execução, projetado para recuperar e analisar imagens egocêntricas, com o objetivo de melhorar a capacidade do utilizador de recordar memórias passadas. O objetivo da dissertação é implementar algoritmos para extrair e organizar informação de imagens e vídeos para análise e recuperação automática de possíveis momentos importantes do utilizador, sem a necessidade de pesquisas.

Foi realizada uma recolha de dados, de forma a suportar a implementação do algoritmo. A recolha de dados consistiu em fornecer o MEMORIA a participantes que carregavam as suas imagens pessoais e classificavam a importância de eventos gerados pelo sistema, com as imagens carregadas. Foram recolhidas as anotações atribuídas aos eventos pelo sistema, com uma importância associada.

A primeira etapa foi a adaptação do MEMORIA para a recolha de dados. O sistema foi dividido em MEMORIA Light e MEMORIA Server. O Light é uma versão mais leve, sem os modelos de processamento de imagem, e o Server é um serviço externo, que inclui uma API para processamento de imagens. Assim, os participantes usaram o MEMORIA Light nos seus computadores, sem exigências de altos requisitos computacionais, com o processamento feito no Server via API.

O sistema original já permitia o carregamento e processamento de imagens, gerando várias anotações de diferentes tipos e assuntos, também agrupava imagens relacionadas através da segmentação de eventos. Estas funcionalidades foram adaptadas para suportar a nova arquitetura. Além disso, implementou-se a atribuição de anotações aos eventos e uma nova funcionalidade, a classificação dos eventos por parte do utilizador. As adaptações e as novas implementações feitas, permitiram recolher as anotações e as suas respectivas importâncias para cada participante.

Foram desenvolvidas funcionalidades de visualização de anotações importantes do utilizador, para analisar melhor o que é importante para o mesmo. Os resultados recolhidos mostraram a diversidade de anotações e as diferentes importâncias atribuídas, de pessoa para pessoa, destacando a variabilidade dos momentos importantes para cada indivíduo.

O tema abordado vai para além do conhecimento existente ao nível das ferramentas de lifelog. Espera-se que melhore a memória ao proporcionar momentos verdadeiramente significativos e adaptados às preferências individuais.

**Keywords** lifeloging, episodic memory, autobiographical memory, image retrieval, natural language processing, latent semantic analysis.

**Abstract**

Lifelogging is the process of collecting, storing and analyzing personal multimodal data in digital format. The dissertation is part of the MEMORIA lifelogging system, a system already under development, designed to retrieve and analyze egocentric images, with the aim of improving the user's ability to recall past memories. The aim of the dissertation is to implement algorithms to extract and organize information from images and videos for the analysis, automatic retrieval of possible important moments of the user, without the need for searches.

A data collection was carried out in order to support the implementation of the algorithm. Data collection consisted of giving MEMORIA to participants who uploaded their personal images and rated the importance of events generated by the system with the uploaded images. The annotations assigned to the events by the system were collected with an associated importance.

The first stage was to adapt MEMORIA for data collection. The system was divided into MEMORIA Light and MEMORIA Server. Light is a lighter version, without the image processing models, and Server is an external service, which includes an API for image processing. Participants therefore used MEMORIA Light on their computers, without demanding high computing requirements, with the processing done on Server via the API.

The original system already allowed images to be uploaded and processed, generating various annotations of different types and subjects. It also grouped related images through event segmentation. These functionalities were adapted to support the new architecture. In addition, the assignment of annotations to events and a new feature, the classification of events by the user, were implemented. This allows annotations and their importance to the participant to be collected for each participant.

Functionalities for visualizing the user's important annotations were developed, in order to better analyze what is important to the user. The results collected showed the diversity of annotations and the different importance attributed from person to person, highlighting the variability of important moments for each person.

The topic covered goes beyond existing knowledge of lifelog tools. It is hoped that it will improve memory by providing truly meaningful moments tailored to individual preferences.

**reconhecimento do uso de ferramentas IA / recognizing the use of AI tools**

Reconheço o uso de ChatGPT 3.5 (Open AI, https://chat.openai.com) para refazer frases que foram escritas, com melhores palavras e explicações. O resultado foi modificado para criar um texto menos artificial, com um toque mais humano, respeitando sempre o meu estilo.

I acknowledge the use of ChatGPT 3.5 (Open AI, https://chat.openai.com) to redo sentences that were written, with better words and explanation. The output was modified to create a less artificial text, with a more human touch, always respecting my style.

# Contents

# List of Figures

# List of Tables

# Glossary

| | |
|---|---|
| **LSC** | Lifelog Search Challenge |
| **NLP** | Natural Language Processing |
| **CBF** | Content-based Filtering |
| **LLM** | Large Language Model |
| **FW** | Feature Weighting |
| **IDF** | Inverse Document Frequency |
| **CNN** | Convolutional Neural Network |
| **QA** | Question Answering |
| **API** | Application Programming Interface |
| **PoV** | Point-of-View |
| **BIQA** | Blind Image Quality Assessment |
| **CLIP** | Contrastive Language–Image Pre-training |
| **CRAFT** | Character-Region Awareness For Text detection |
| **YOLO** | You Only Look Once |
| **SIFT** | Scale Invariant Feature Transform |
| **GRiT** | Generative Region-to-text Transformer |
| **FAISS** | Facebook Artificial Intelligence Similarity Search |
| **SWOT** | Strengths Weaknesses Opportunities Threats |
| **MEMORIA** | Memory Enhancement and MOment RetrIeval Application |
| **IEETA** | Institute of Electronics and Informatics Engineering of Aveiro |
| **WSGI** | Web Server Gateway Interface |
| **ASGI** | Asynchronous Server Gateway Interface |

CHAPTER 1

# Introduction

Over the last few years, with technological advances, different types of sensors and devices have increasingly emerged to collect data on human activities and behavior [1] [2]. For example, cameras, smartphones and wearable devices, have enabled people to store their personal data, both as special moments and day-to-day, in real time. In addition, their availability has increased, and their size and prices have decreased, thus making access to these technologies easy in society [2].

Given the wide variety of data collection devices, different types of data have emerged, which are very heterogeneous [1] [2]. A wide variety ranging from images, audio, geographical coordinates to biometric information, etc [1]. The wide accessibility of these devices by society and their easy use has led to a large volume and rapid speed of data collection [2].

This data, known as lifelogs, can be put to good use in various applications, such as: retrieving moments, understanding everyday life, diagnosing mental illnesses, searching for images/videos, etc. Their use leads to the need to store, analyze, process and retrieve them [1] [2]. However, these tasks of analysis, processing, etc. are difficult and complex due to the variety, volume and speed of the data [1] [2]. Therefore, the tasks mentioned above are carried out by a system with various algorithms and methods [1] [2].

This dissertation will be applied to such a system, the Memory Enhancement and MOment RetrIeval Application (MEMORIA) system, a system that has already been started in previous years [1]. MEMORIA is a project currently being carried out by a research team from IEETA/Department of Electronics, Telecommunications and Informatics/ University of Aveiro. It is a system for retrieving and analyzing egocentric images using multimodal datasets [1] (Figure 1.1).

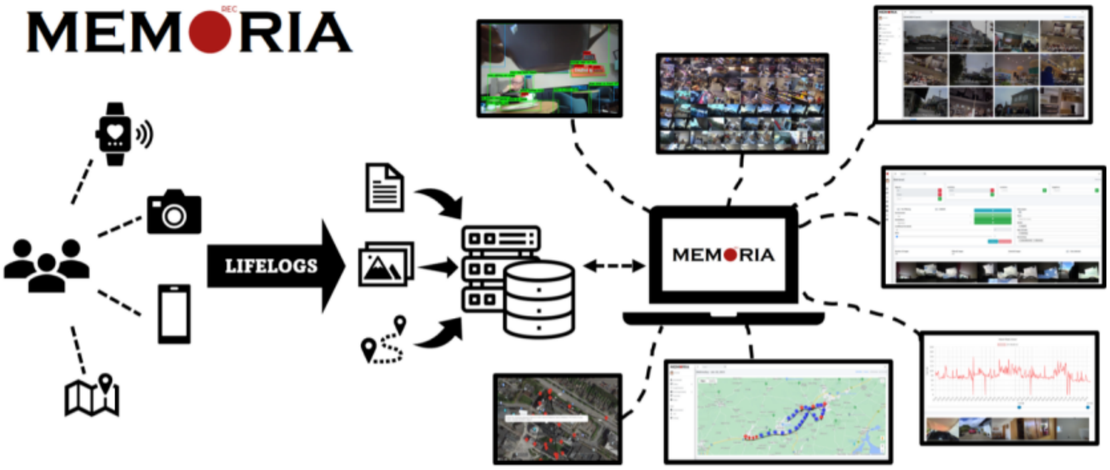



**Figure 1.1:** An overview of the MEMORIA lifelogging system [3].

Lifelogs are essentially digital diaries that document a person's everyday activities. The MEMORIA application was designed with the aim of recovering past moments and improving the recall of users' experiences and memories through a user-friendly interface. Visual lifelogs are the main emphasis because of their potential to improve individual memory [1].

The MEMORIA system allows users to upload and analyze lifelogs, explore segmented events, retrieve images and events, search for images through a text query, among other features. The system makes use of image quality analysis and image annotation techniques to enhance the lifelogs. In order to handle the growing volume of data, enhance system efficiency, and provide user events and moments for display, the event segmentation approach was created [1].

## 1.1 Objectives

The number of cases of dementia has been growing, and it is predicted that there will be even more cases in the future. Memory loss and difficulties in creating new memories are some of the problems associated with dementia, which is why we need to take precautions and look for solutions. Memories are essential for our daily lives and for a fulfilling life. With them we build our sense of identity and make decisions based on our history [4].

In order to meet this challenge, memory rehabilitation techniques have been sought and studied. With technological evolution, the idea of using technology to develop new techniques for this purpose has emerged. The use of technology to improve memory is a field that has only recently been explored, with many theoretical studies but few practical ones.

Lifelogging is an emerging technique that uses technology to improve memory. It is the process of collecting, storing and analyzing personal multimodal data in digital format. In the context of lifelogging, it is common to collect data passively, automatically without action by the photographer, thus generating large data sets. Today's lifelogging systems offer various functionalities for exploring and analyzing the large data sets collected by the process.

However, due to the large number of images, it can be difficult for the user to search for what they want to see [2].

Nowadays, everyone is bombarded with content recommendations on the internet, on social networks, streaming platforms and so on. All this content doesn't appear by chance, behind it are sophisticated algorithms that analyze user behavior to make suggestions. This gave rise to the idea of applying memory recommendations to lifelogging.

The aim of this work is to develop and implement algorithms capable of extracting and organizing information obtained from images/videos that are useful for the analysis, identification and automatic retrieval of relevant memories in the user's daily lives, without the need for research. The research question is the following: «How is it possible to extract, organize relevant information from images/videos stored over time in order to identify and retrieve important memories?».

For each person, what is most important varies. It is hoped that this feature will improve memory by providing truly meaningful moments for the user, as it could be the information they really want to see. Instead of offering a large set of images and features that require the user to search for what they want (a difficult task when there is a lot of data), as the old MEMORIA system and current lifelogging systems do. The aim is to directly suggest the content that the user wants to see without requiring any action on their part.

The tool could be aimed at regular people who love technology and are interested in accessing their photographic memories quickly and easily. People with mental health problems could also be recipients, with a tool to support them in recovering important moments. Finally, caregivers/doctors/therapists who can use MEMORIA as a diagnostic and therapeutic tool.

This thesis will explore the link between event segmentation and memory retrieval of autobiographical/episodic memories. The system already segments events, organizes and retrieves lifelog data in homogeneous temporal segments, each representing different events and personal experiences. Episodic memory is important in lifelogging because it represents the key unit for segmenting and organizing events. Episodic memories are summary records of experiences that represent the unfolding of events in a temporally compressed form, just as an event represents in the segmentation of events developed in MEMORIA [5].

To recap, a new feature will be introduced to the MEMORIA system, the delivery of relevant events to the user without requiring a search. For each person, what is most important can vary, it depends from person to person. It is hoped that this functionality can enhance memory by providing truly meaningful moments for the user, and by adapting to individual preferences.

## 1.2 Document Structure

This document is divided into six chapters. Chapter 1 deals with the context, motivation and objectives of the dissertation. Chapter 2 defines all the concepts used in the area, talks about the latest work carried out in the area, what has been developed, what conclusions have been drawn, and what future work has been identified. Also in Chapter 2, content-based recommendation tools and technologies are analyzed in order to assess their applicability in identifying and retrieving the most relevant moments for the user. Chapter 3 presents the work plan for this dissertation. Chapter 4 explains the adaptations made to MEMORIA in order to support the algorithm for recommending important events. Chapter 5 reveals and analyzes the results obtained. Chapter 6 ends the dissertation with the conclusions of this work, final considerations and suggestions for future work. Finally, Appendix A presents the paper submitted to the LSC 2024 conference [6], where some contributions made in this work are described, and Appendix B presents the written experiment protocol for an experiment carried out within the scope of this dissertation.

CHAPTER 2

# Literature Review

*This chapter covers a review of the literature, providing definitions of the concepts used in the area of this dissertation, such as lifelog, lifelogging, episodic and autobiographical memory, and event segmentation. Recent work developed in the area is also discussed, including some participants in competitions with tasks associated with lifelogging. In addition, current approaches and resources in content-based recommendation are analyzed.*

## 2.1 Lifelog and Lifelogging

**Lifelogging** can be defined as the process of collecting, storing and analyzing personal multimodal data in digital format. It consists of the collection of an individual's day-to-day activities and experiences (this individual is technically called a lifelogger). This data is collected by lifelogging devices, electronic devices such as wearable devices (Figure 2.1), or even smartphones, with digital sensors that enable this collection [1] [2]. In the practice of lifelogging, it is necessary to collect personal data over long periods of time, or even over the lifelogger's entire life [2].


**Figure 2.1:** The Microsoft Research-developed SenseCam is usually worn on an adjustable lanyard. SenseCam is a wearable camera that takes photos automatically [7].

**Figure 2.2:** First-person view / PoV images captured to reflect the user's visual perspective. Images that are normally used in the context of lifelogging. [7].

The multimodal data collected through lifelogging is called **lifelog** data. The main types of lifelog data are: visual (Figure 2.2), audio, location, physical activity and physiological signals. Because of their richness and the advances in image processing algorithms, visual data are most frequently used [2].

Lifelogging systems are lifelog analysis and retrieval systems. The lifelog functions as a surrogate memory within lifelogging systems capable of organizing and managing lifelogs. Some studies indicate that human digital memories are lifelogs, others indicate that they are the results of organizing and processing lifelogs [2].

In these systems, combining various types of lifelog data can contribute to better results (i.e. better retrieval of lifelog data). This is because, although visual lifelogs are currently the most widely used for lifelogging, due to the amount of information they can provide, and the advances in image processing that have taken place, other types of lifelog data (e.g. audio and location) can complement them, as they provide clues that visual lifelogs do not. For example, we can combine visual and location lifelogs in order to know the precise location where the image was taken [2] [8].

Lifelogs are promising tools for memory retrieval and understanding individual experiences and behaviors. Furthermore, can improve quality of life by improving memory for both those with memory problems and healthy individuals, while also providing support for individuals struggling with this type of memory problem [2].

Lifelogs in lifelogging applications may contain irrelevant or useless data. Preprocessing techniques are used to identify pertinent lifelogs and remove or fix those that might cause noise and errors in the system [2].

Annotating, organizing, and storing lifelogs with semantic concepts is crucial to enhance understanding of the lifelogger's behavior and to define events and specific moments. Since semantic concepts provide more information about the lifelogger's environment and activities. These annotations act as stand-in memories for visualizations in the future [2].

Effective memory retrieval depends on the development of algorithms and techniques capable of interpreting and analyzing lifelog data. For machines, lifelogs are data without any meaning, for example, visual lifelogs are just pixels. Algorithms that examine lifelogs are essential for interpreting and assigning semantic value to them. In addition, it is important to take into account multimodal lifelogs, which can be very heterogeneous and unstructured due to the variety of collection devices, and the long periods of time to which collection is subjected. The semantic information gathered by these algorithms makes lifelogs useful in a variety of applications, such as life event tracking, memory enhancement, health and well-being intervention, research, analysis, among others [2].

Yen *et al.* [9] presented a visual lifelog retrieval system that was developed to answer textual queries. The system aimed to assist with memory recovery for people. It incorporated visual and textual concepts, using pre-trained textual embeddings. It offered search filters such as the number of people, location, and date. Recommendations of concepts were provided to users to make queries more effective and with more appropriate terms. The scene graph generation model was implemented to annotate the relationships between objects in an image. The daily timeline visualization feature of the lifelogger was designed to provide an overview of the activities the lifelogger completed each day. In order to compare machine captions with human captions, human captions linked with the images were created using the Cloudsight API, a startup cloud service that offers an image caption annotation tool with human-robot collaboration.

All the functionalities implemented in the system of the study developed by Yen *et al.* [9] improve retrieval results of visual lifelogs, but still offers insufficient details to close the semantic gap between the query and the images. Results retrieval is hampered by the existence of irrelevant visual concepts and inaccurate machine-generated captions. Computer vision models trained on third-person images still have difficulty comprehending actions in first-person images. The human ability to filter out irrelevant information and infer complete events was evident, yet it can also be difficult for humans (who were not lifeloggers). When irrelevant or incorrect annotations were created, machines would focus more on the locations and larger objects in photos, while humans would filter out irrelevant things and deduced action specifics. It turns out that human and model captions can complement each other. The study emphasizes how crucial the lifelogger's experience timeline was.

Privacy is a concern with lifelogging, particularly when bystanders are being recorded without their consent. Lifeloggers can also have privacy issues, as the data can be powerful for companies and advertisers, which is why the need for data protection regulations is important. It is illegal to record audio or to take pictures without the express permission of the subject. There are already some techniques that have tried to get around this problem in lifelogging, such as blurring faces in visual lifelogs, but this has the disadvantage of damaging the memory aid function [2].

Our sense of reality may be distorted by lifelogs, making memories appear more recent than they actually are. However, this only documents a small portion of reality, preventing complete documentation. One of the concerns related to this topic is the fact that people trust lifelogs too much, and there may even be obsessions with memories. Ownership of lifelogs and the information they contain should be owned by the lifeloggers, who should also be conscious of how their data is used [2].

## 2.2 Episodic Memory and Autobiographical Memory

**Episodic memories** are recollections of one's own past and may contain spatial details or locations, accompanying feelings, and additional contextual information about particular events. The three basic elements that form the basis of episodic memory recall are: mental time traveling (subjective sense of time); autonoetic awareness (allows individuals to be aware of the «self» in a memory) and linking memories to an individual's «self» [10]. A person's episodic memory could be, for example, their graduation day from university.

Summaries of experiences known as episodic memories are temporally compressed representations of the events as they unfold. The temporal compression of experience in episodic memory will be detailed further in Section 2.4 [5].

**Autobiographical memory** and episodic memory share many characteristics. It happens when episodic memories are associated to autobiographical knowledge of oneself [11]. The best medium for representing autobiographical memories is a photograph [2]. An example of an autobiographical memory is the fact that a person knows their date of birth, this is an autobiographical memory of the person in question.

One of the leading causes of chronic impairment in the globe is memory loss. Some factors that lead to significant memory loss are, for example, amnesia, normal aging, and brain disorders like Alzheimer's. Particularly vulnerable to several kinds of brain degeneration is episodic memory [10]. Studies predict that the number of people with dementia will increase in the coming years (Figure 2.3) [4].

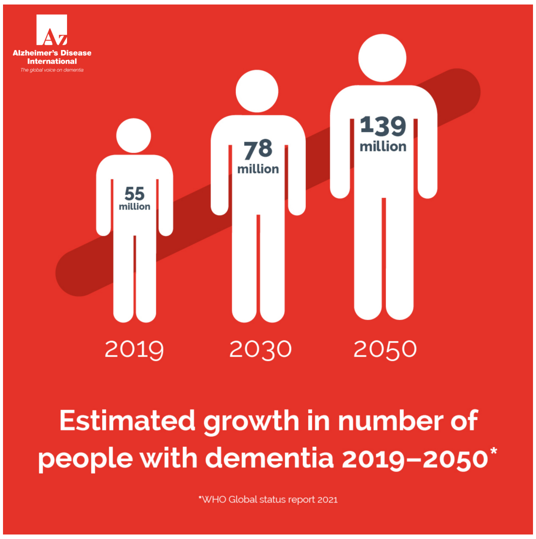



**Figure 2.3:** Number of people living with dementia in 2019 and the estimate for 2030 and 2050 (in millions) [4].

The ability to remember things is crucial for daily tasks, identification, self-awareness, and even life quality. Amnesia and severe cognitive impairment are common in Alzheimer's patients, and the condition can even require acute care. The significance of memory functions has led to a notable surge in interest in memory rehabilitation methods. These techniques can be classified as internal and external. While internal methods like cue-based learning and errorless learning work well for skill acquisition, they are not appropriate for episodic memory [10].

Lifelogging is an external technique that stands out as one of the few effective approaches to preserving episodic memory. Its success may have something to do with the lifelogging devices' ability to review lifelog photos, which offer useful hints to certain memories. Visual lifelogs are the most successful type of lifelog because they capture people's life activities comprehensively and objectively [10].

Teijlingen *et al.* [10] made a narrative review of the literature on lifelogging and memory rehabilitation, Table 2.1. They reviewed case studies in brain-damaged patients, they also reviewed clinical group studies in patients diagnosed with Alzheimer's dementia and experimental studies in healthy participants.

| **Type of studies** | Case studies | Clinical group studies | Experimental studies |
|---|---|---|---|
| **Health of participants** | brain-damaged patients | patients diagnosed with Alzheimer's dementia | healthy participants |
| **Number of studies** | 7 | 4 | 7 |

**Table 2.1:** Studies analyzed on lifelogging and memory rehabilitation by Teijlingen *et al.* [10].

In studies involving patients with mild to severe amnesia (case studies and clinical group studies), lifelogging had positive effects on subjective and objective memory measurements. In experimental studies on healthy people, lifelogging didn't always help with memory rehabilitation. But overall, in studies on healthy people, it is still concluded that lifelogging is successful in its aim of supporting memory [10] [8].

The studies that used lifelogging were more successful than personal diary writing [10] [8].

Patients have given excellent feedback regarding the usage of lifelogging devices, like the Sensecam. In addition to improving memory, have appreciated the fact that it allows them to review images and recall events. These devices have benefited patients with different brain disabilities [10] [8].

During the review sessions, events captured by the devices are repeated, which facilitates simpler retrieval of the information by the memory. Repetition increases the strength of a memory in the brain, i.e. increases the ability to recall a particular memory [10] [8].

First-person perspective images have shown better results in memory retrieval compared to third-person perspective, as the accuracy and vividness of the memories are greater [10] [8].

However, lifelogging devices have significant consequences, the usage of technologies for memory recovery is quite limited. Some lifelogging systems have been put on hold due to privacy and complex usability concerns. In spite of this, lifelogging has a great deal of promise for aiding those who suffer from memory loss. It offers a viable substitute for traditional memory rehabilitation methods, particularly in clinical settings [10] [8].

Woodberry *et al.* [11] carried out an experiment involving six participants diagnosed with

Alzheimer's disease, aged between 64 and 84, all of whom had mild to moderate stages of the disease. The research was conducted in two distinct phases.

In the first phase, which lasted two weeks, each participant recorded significant events using either SenseCam or a written diary. During this period, recall tests were carried out, involving reviewing images for those who used the SenseCam and reviewing notes for those who opted for the diary. Recall performance was measured and recorded [11] [8].

The second phase of the study took place in two iterations, one after one month and the other after three months. During this phase, the participants were given new recall tests and their recall performance was assessed. The participants who used the SenseCam did not review the images during this stage. In addition, some significant events collected in the first phase were purposely not reviewed during the initial two weeks. This approach was adopted to allow these events to be subsequently subjected to recall tests in the second phase, contributing to the evaluation of recall performance and establishing a baseline [11] [8].

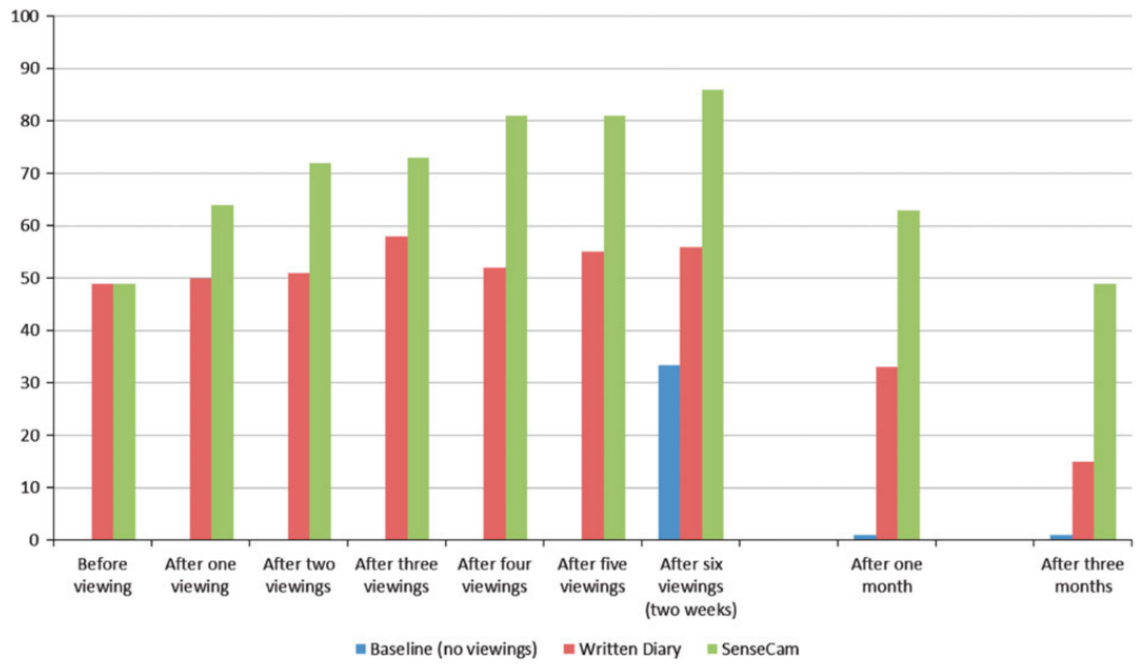



**Figure 2.4:** Results of the experiment carried out by Woodberry *et al.* [11]. The average percentage of autobiographical events remembered over a two-week period, as well as tests conducted at one and three months later for the written diary, SenseCam and baseline conditions.

In Figure 2.4 we can see the results of the experiment. The SenseCam outperformed the diary in recalling autobiographical events during the first two weeks for the majority of participants. In the long term, after one and three months, the SenseCam continued to be more effective in retaining memories. But it should be noted that the diary is more effective than the baseline condition (events that were not aided by memory) in the long term. After two weeks, for the baseline events, the average recall was 33%, and after one and three months, participants had almost no recollection of the original events [11].

In the literature review carried out by Ribeiro *et al.* [2], a study of a patient with amnesia was examined. In this study, it was observed that wearable devices and smartphones have a positive impact on autobiographical memory, improving both short-term and long-term memory. The written diary, on the other hand, only showed improvements in the short term.

### 2.3 Lifelog Search Challenge 2023

In recent years, workshops and challenges have emerged on tasks associated with lifelogging, in order to talk about methods, analyzes and uses, recognizing potential and difficulties in this field [1] [2].The Lifelog Search Challenge (LSC) is one of the challenges that has emerged.

Every year, international teams are encouraged to participate in the LSC, which aims to create effective interactive lifelog retrieval systems that can search vast lifelog databases [12] [2]. Once these systems have been implemented, the LSC offers a challenge. Starting from the same set of lifelog data, teams compete in real time to solve the same lifelog retrieval tasks performed by the lifelogger [12] [1] [2]. In this challenge, the performance of the search is evaluated, who retrieved the right images as quickly as possible in each task [12].

The sixth edition of the LSC took place in 2023 [13], with twelve teams developing interactive lifelog search engines. Of these teams, eight had already taken part in previous editions, presenting new versions of their systems in 2023, while the remaining four teams had never taken part.

In the sixth edition, the teams received a multimodal lifelog dataset collected by a lifelogger over 18 months, between January 2019 and June 2020 [13] [14]. This dataset contains Point-of-View (PoV) images, images that show what the lifelogger sees, captured frequently throughout the day by a wearable camera, the Narrative Clip [13] [14]. Through an automated procedure, all faces and screens were edited from the images. The images were chosen manually by the lifelogger, removing those of poor quality or that he chose not to share. The images were collected in conjunction with UTC time and accompanied by other metadata, such as biometrics, geolocation and music listening history [13] [14]. Each image was enriched with visual concepts extracted by the Microsoft Computer Vision API and OCR text from the images.

In 2023, the contest had 24 tasks, each task consisting of finding the answer to a query [13]. The tasks were grouped into three categories: ad-hoc/conventional, allowing one (or more) lifelogs to be submitted as an answer; single/few known item, where one (or a few) lifelogs could be used as an answer; and Question Answering (QA), which included detailed questions about lifelogs, requiring a textual answer in addition to the lifelogs [13] [14]. The ad-hoc and QA tasks were evaluated in real time by the judges [13]. Each task lasted 3 minutes, with additional information for the query every 30 seconds [1] [8].

LifeXplore [12] was the winning team of LSC 2023, having participated since 2018. The team's focus was on creating an intuitive interface and a robust backend, allowing free text search and temporal filtering. The architecture of the system incorporated previously created techniques as object detection (YOLOv7), concept detection (EfficientNet) and text recognition (CRAFT). It combined these models developed in previous years with the new method developed in 2023, an approach that extracts text-embeddings from each image using a vision transformer, OpenCLIP ViT-H/14. The new version allows the use of special commands in queries. If the query has special commands, the results of the methods from previous years are merged with the new method. Otherwise, only the results from the new model are used, i.e. only the embeddings from OpenCLIP ViT-H/14 are used for the search. The methods from previous years remain, and temporal context filtering has been added for queries in which CLIP does not identify the target scene [8].

MyEachtra [15], which came second in the LSC 2023, was the winner in three consecutive editions, 2022, 2021 and 2020. After substantial upgrades, the system is now an event-based lifelog retrieval system. The main motivation came from the speculation that the temporal nature of lifelogs can make systems more robust. A new user interface emphasizing significant events was implemented. In addition, an event-centered approach was adopted, segmenting events based on location metadata and visual and temporal changes between subsequent images. Data processing was also modified, using CLIP embeddings to determine image similarity, modifying the previous method with VGG16 and SIFT. In the updated system, a method was developed to generate embeddings of events, using the average of the embeddings of the images belonging to the event in question. The system highlights essential details such as location, time and significant images when showing classified events. Using the FrozenBiLM model improves the response to QA queries. With the new event-based method, performance was not compromised, and it originated in a reduced search space [8].

Memento [16] took third place, improved the search and ranking functionality and modified the user interface to serve new types of queries. Improved image-text embeddings have been implemented to better comprehend the semantics of the data. With an approximation search methodology based on clusters in the image embedding space, the system now functions faster. The embeddings come from OpenCLIP (more extensive, trained on around five times more data) and OpenAI CLIP models. In spite of the larger model size and training set, OpenCLIP did not perform appreciably better than the OpenAI CLIP model. In view of this, the user interface was modified to add the functionality of the user being able to dynamically switch the backend model related to the search. Using an inverted file index from the FAISS library improved the vector similarity search [8].

The second version of the MEMORIA [1] replaces the relational database with a graph database, offering a more sophisticated search engine with support for location processing and free text search. A sequential and more explicit representation of temporal moments was achieved by using a hierarchical approach to event segmentation, which was based on semantic locations and additional image annotations. The system has improved image annotation and pre-processing, as we can see in the Figure 2.5, integrating new methods, deep annotation models (GRiT), optical character recognition, object detection (YOLOv7) and CLIP. Additionally, a more straightforward upload process and the addition of thumbnails have enhanced the system's user experience [8].

| Types of annotations | MEMORIA at LSC'22 | MEMORIA at LSC'23 |
|---|---|---|
| Object detection | [(car, 0.638) , (car, 0.399)] | [(car, 0.621), (truck, 0.663)] |
| OCR | ——————— | [] |
| Caption generation | ——————— | "view from the driver's seat." |
| Deep annotation | ——————— | [("red truck in rear", 0.554), ("the rear view mirror on the car", 0.554)] |

| Types of annotations | MEMORIA at LSC'22 | MEMORIA at LSC'23 |
|---|---|---|
| Object detection | [(bottle, 0.904), (couch, 0.663), (potted plant,0.389), (refrigerator, 0.367)] | [(tv, 0.471), (couch, 0.906), (cup, 0.432), (cup, 0.353), (bottle, 0.922)] |
| OCR | ——————— | [('flora', 0.836),( '10o', 0.387), ('gooune', 0.063)] |
| Caption generation | ——————— | "view from the kitchen into the living room." |
| Deep annotation | ——————— | [("a bottle of water", 0.607), ("a white bucket on the floor", 0.737), ("a calendar page with scenery on it", 0.6803), ("a skylight in the ceiling", 0.699), ("a white and blue label on a refrigerator", 0.615), ("an electrical outlet on the wall", 0.595), ("a newspaper on the wall", 0.620), ("green cap on a water bottle", 0.5839)] |

| Types of annotations | MEMORIA at LSC'22 | MEMORIA at LSC'23 |
|---|---|---|
| Object detection | [(person , 0.499), (car, 0.370)] | [(person , 0.43), (car , 0.46)] |
| OCR | ——————— | [("centra", 0.166)] |
| Caption generation | ——————— | "driving a car in the rain." |
| Deep annotation | ——————— | [("a man driving a car", 0.640), ("person in the car", 0.564), ("the hand of a person", 0.626)] |

| Types of annotations | MEMORIA at LSC'22 | MEMORIA at LSC'23 |
|---|---|---|
| Object detection | [(car, 0.938), (car, 0.914), (car, 0.313)] | [(car, 0.963), (car, 0.916), (car, 0.298)] |
| OCR | ——————— | [('o6di', 0.258), ('58377', 0.433)] |
| Caption generation | ——————— | "property image # detached holiday home in quiet location with garden" |
| Deep annotation | ——————— | [("a blue car parked on the street", 0.702), ("black car parked in street", 0.670), ("license plate on the car",0.707), ("red tail light on a car", 0.582), ("back tire of the car", 0.614), ("the back window of the car", 0.589), ("window on a brick building", 0.606), ("window on a building", 0.626), ("window on a building", 0.587), ("cars parked in front of a building", 0.609), ("a red brick building", 0.551), ("window of a blue van", 0.585), ("window in a brick building", 0.645), ("window on a building", 0.622)] |

**Figure 2.5:** Enhancements to the annotations extracted from the images in MEMORIA in the LSC'22 dataset [1].

The fourth iteration of the FIRST system [17] incorporated tools to help develop textual queries, based on the hypothesis that users may have difficulty formulating queries. Through an iterative dialog with the machine, the system suggests relevant contextual cues using generative models like GPT-3.5 Turbo. Alternatives to the initial query that are contextually appropriate are shown to the user, enabling selection and modification as needed. Temporal segmentation of events was introduced, which made it possible to index sequences of images as events. This approach improved the interface and made searching more efficient and contextualized [8].

Voxento 4.0 [18], a prototype voice-controlled interactive lifelog retrieval system, aimed to improve voice interaction, interface and retrieval techniques. In this version, the dataset underwent data processing and cleaning, and the CLIP model was used to extract image attributes and increase the accuracy of the text image similarity comparison. The revised interface offers enhanced visualization and interaction features, such as better voice interaction registration and workflow optimization with the addition of more voice commands [8].

E-LifeSeeker [19] presented the fifth version of the system, including the latest embedding models - CoCa, CLIP, BLIP and ALIGN - which have been pre-trained on large amounts of data. The main search engine model can now be chosen by the user. Additionally, Differential Networks have been constructed for the new QA task. After the initial query, users can use the Fusion model based on Differential Networks to evaluate the images retrieved in the query, and thus help answer the question. With added features and visual improvements, the user interface is now more intuitive, especially for novices, providing a more productive experience [8].

LifeGraph 3 [20] presented an approach to organizing lifelogs in a multi-modal knowledge graph based on cluster hierarchies, according to temporal, geographical and visual variables. The resulting clusters comprise the semantic structure of a graph with all lifelogs as entries, together with semantic metadata and OpenCLIP-based augmentations. Before storing the information in the graph, the data was pre-processed. The interface provides interactive navigation and exploration capabilities by allowing users to filter the set of entries that are pertinent to a given query and navigate through cluster hierarchies [8].

All the systems described earlier are teams that had already taken part in the LSC. Now let's look at the new systems taking part in 2023.

Spiess *et al.* [21] developed a new multimedia retrieval system based on the vitrivr stack. This hybrid system combines the immersive navigation offered by vitrivr-VR, which runs in virtual reality, with the speed at which queries may be quickly created using the vitrivr-ng desktop interface [8].

MemoriEase [22] is another new system, which uses concept-based retrieval and embedding approaches. MemoriEase extracts embeddings from images and clusters images into segments using the segmentation technique, through BLIP model. Using Elasticsearch's full-text search, the concept-based retrieval strategy finds images whose visual concepts correspond to the query's keywords. In addition, the system combines the relevance found by the approaches, to deliver a balanced score in the classification of images, for a given query. The system strives to make the procedure easier for both inexperienced and seasoned users using an intuitive user interface [8].

LifeInsight [23] incorporates the BLIP embedding model and provides information about the lifelogger's routine, using spatial data to support QA tasks. It has visual similarity search and uses Elastic Search as a filtering technique to exclude pointless images. Explicit relevance feedback, reformulation of the query description based on artificial intelligence and the creation of visual examples to improve the query are features that have also been developed [8].

LifeLens [24], built on the E-LifeSeeker [19] recovery engine, stands out for offering a simple, user-centered user experience. Its minimalist and intuitive interface has been developed to improve usability and interaction, designed especially for lifelogs. The system offers the following features: a text search bar for searching images, a timeline for browsing lifelog data, a feedback function that allows users to find similar images from a selection of other images, a filter for filtering results by time, location and person, a grid search results page, and an additional details page for a specific image [8].

In general, almost all participants used text-image embedding models, and most used the CLIP model or models derived from it. Event segmentation has also been incorporated into many of the systems, which can reduce retrieval time by optimizing the search process and freeing up screen space. In addition, they have enhanced the user interface, since making the search for lifelogs easier to use also requires an intuitive and well-organized layout.

### 2.4 Event Segmentation

The concept of an **event** can be a very subjective definition. Generally, events are segments where something happens at a certain place and time, with a beginning and an end. In the context of lifelogging, the definition of event can become more flexible, focusing on the key components of personal experience [3].

Within lifelogging, **event segmentation** plays a crucial role by organizing, structuring, indexing and retrieving lifelog data into homogeneous temporal segments, each representing distinct events (Figure 2.6). The constant increase in the collection of lifelogs has driven research in this domain, resulting in a growing need for efficiency in data organization and retrieval. This scenario has led to the emergence of studies that have adopted event segmentation to meet this emerging demand. Although event segmentation is a relevant process, it faces a critical issue in detecting event boundaries, which reflect the transitions between different activities [3].

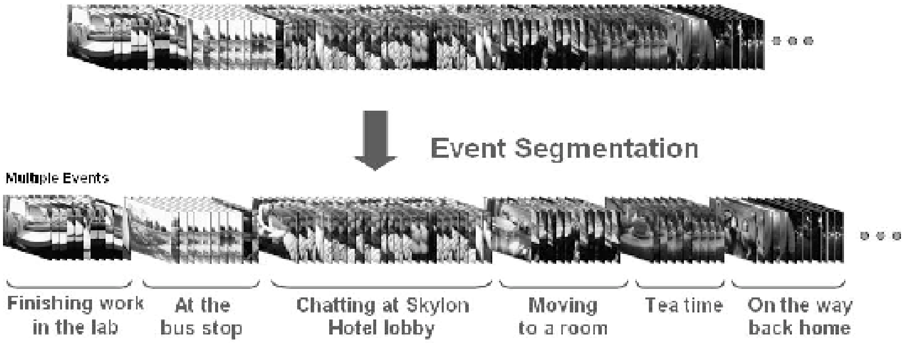



**Figure 2.6:** Example of an event segmentation (Segmenting images into events) [25].

Segmenting events can save screen space and speed up the search process, reducing the time it takes to retrieve lifelogs [1]. This is crucial in a context where there are many lifelogs collected. In addition, event segmentation has the potential to improve memory organization and retrieval in lifelogging systems [3].

Understanding episodic memory and its application in lifelogging has proven to be a field that is constantly being explored. However, these domains still need further exploration and more detailed study [5]. Episodic memory acts as the fundamental unit for segmenting and organizing events, playing a crucial role in memory enhancement [3] [8].

Recent studies suggest that episodic memory encapsulates the continuous flow of information that constitutes the events of daily life, compressing them temporally [5]. Event boundaries structure the encoding and organization of information in episodic memory, allowing temporal compression to remember events in a shorter period than their actual duration [5].

Temporal compression in episodic memory occurs due to discontinuities in the representation of the unfolding of events, manifesting itself as a succession of moments called «units of experience» (Figure 2.7). Longer time intervals between remembered moments are associated with higher compression rates [5].

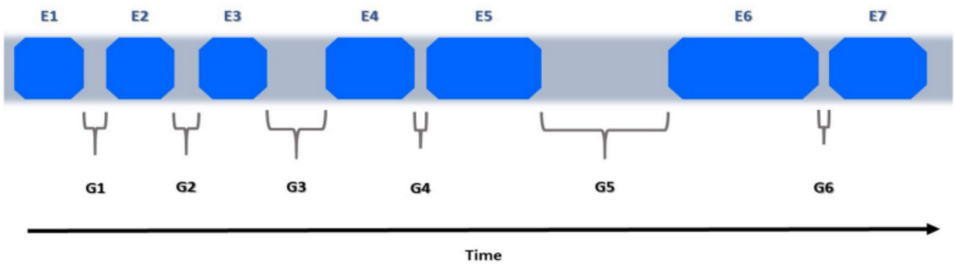



**Figure 2.7:** Potential process behind the events in episodic memory being compressed in time [5].

The process of visualizing the segmentation of events in a lifelogging system, that is, the analysis of the continuous flow of experiences into events and sub-events, can play a crucial role in the formation of these units in episodic memory [5].

Prior research has demonstrated the important effects of event segmentation on long-term memory, where event boundaries are encoded more richly, determining organization in episodic memory [5]. Monitoring goal processing and adjusting temporal compression rates to preserve goal-relevant information is a crucial role of episodic memory [5]. Changes in goal processing are often correlated with event borders, indicating that a hierarchical structure of goals and sub-goals may serve as the foundation for the formation of episodic memories [5].

Jeunehomme1 and D'Argembeau [5] carried out an experiment with the aim of exploring whether temporal compression in episodic memory depends on the segmentation of events. Initially, participants were instructed to take a walk on a university campus while their experience was recorded by a wearable camera. Along the way, they had daily activities to carry out. The activities and their order were the same for all the participants. In addition, the activities were goal-directed actions («hand events»), or spatial displacements («foot events»), or neither. The participants were not informed that their memory would be tested, so as not to influence the results.

After the walking phase, the memory task followed, individuals were required to mentally relive the walk and describe every moment in as much detail as they could, using a digital audio recorder to capture their descriptions. The participants then listened to the audio recording and had to identify the moments in the video corresponding to each audio description [8]. Thus identifying different units of experience. For each unit of experience, the participants were asked to choose, from a set of images derived from the video recording, the one that best reflected their mental representation of that moment.

Finally, the participants performed a segmentation task, indicating when they thought the end of one event and the beginning of another occurred. They were told that the events could be divided into smaller parts. In this segmentation task, participants were asked to divide the series of images into the smallest events that made sense to them. After completing the exercise, all participants were asked if they expected their memory for the walk to be tested.

The findings suggest that there is a considerable correlation between the event segmentation's grain size and the episodic memory's temporal compression. Segmentation grain sizes and temporal compression rates are four to five times smaller in goal-directed action events than in other action types (spatial displacements and activities involving neither actions nor displacements). Goal-directed behaviors had a higher experience unit density, which suggests less temporal compression in the events in episodic memory, leading to shorter temporal gaps. Additionally, participants recognized a greater number of sub-events for actions with specific goals, and a finer segmentation of events results in a smaller compression in episodic memory [8].

Both the event borders and the grain size of the event segmentation predicted temporal compression in memory. Parts of the event were not as frequently recalled as its borders. Actions with specific objectives are essential for creating units of experience in memory. The findings of the experiment are crucial to understanding how episodic memories depict the progression of events in a temporally compressed way.

The system into which this work will be integrated, MEMORIA [1], already has event segmentation based on hierarchical events, using previously processed lifelog data, such as temporal information, location, environment, image similarity and semantic annotations [3]. The hierarchical approach includes five layers: Days, Parts of Day, Locations, Environment and Image Similarity (Figure 2.8). Temporal segmentation can be adjusted based on the user of the lifelogging system [3].

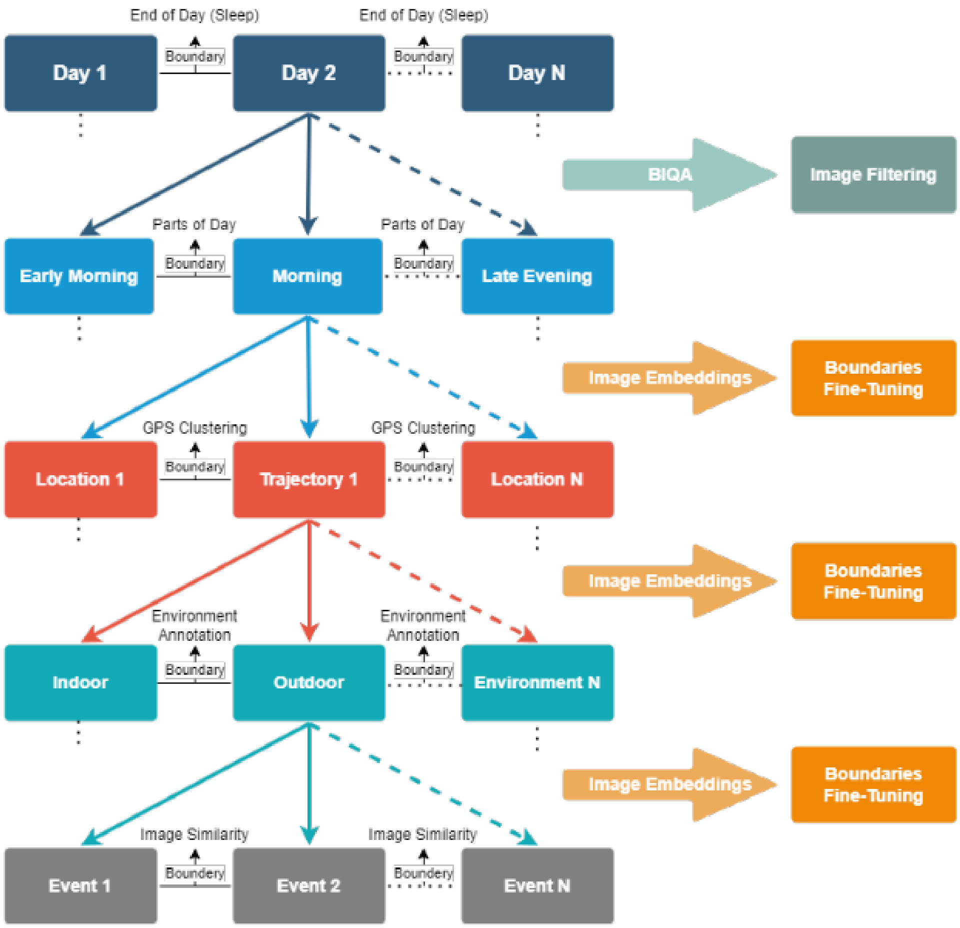



**Figure 2.8:** Event segmentation approach based on hierarchical events, in MEMORIA aplication [3].

Blind Image Quality Assessment (BIQA) is a sub-layer that sits between the Day and Parts of the Day layers. Its purpose is to remove low-quality images and those that don't have pertinent contextual information [3].

Between the other layers, there is a sub-layer that automatically fine-tunes the boundaries of the events in the previous hierarchical layer. This proposal aims to identify two distinct boundaries between two adjacent events, using image content similarity to dynamically adjust the event boundaries. The approach depends on the layer we are working on, and allows for a more meaningful organization of events [3].

The efficient hierarchization of lifelog data facilitates intuitive navigation through events, improving visualization. This proposed organization of images considerably reduces the volume of data, keeping important content the same, and improves performance in retrieving memories and moments. Future work includes exploring natural language processing techniques to generate textual descriptions of events, to enable queries based on these descriptions [3].

### 2.4.1 Event Annotations

Although the MEMORIA system [1] has not yet implemented event annotation, some systems that have addressed event segmentation have already done so, as in the case of MyEachtra [15]. In this system, event embeddings are created by joining the embeddings of each event image in a matrix. The embeddings of each image are previously created by the CLIP model.

After joining, an aggregation function is applied to the matrix to create a global event representation. Four aggregation functions were tested: average of the embeddings of the images belonging to the event; clustering of the events with selection of the cluster centers as representative embeddings; emphasize important images using transform encoders and learning a self-attention method, the outputs are mean pooled; and finally, passing through a linear layer to generate picture weights that represent the importance of each one, and then using the results to build a weighted average of embeddings. The average embedding of images was the approach that proved most suitable for MyEachtra.

## 2.5 Recommendations using Natural Language Processing

The vast amount of information available on systems such as social networks, streaming platforms, e-commerce services, and so on, makes it difficult for users to find data that is specific and relevant to their needs. A recommendation system can solve this problem by providing users with personalized and relevant information [26].

Content-based Filtering (CBF) is one of the categories of recommendation systems [26], Figure 2.9. CBF recommends items based on the user's preferences, i.e. items similar to those the user has already liked or rated highly. It also takes into account information from the user's profile or previous actions [8]. These recommendations are personalized for each user and can result in more accurate suggestions [26] [27]. Similarity between items is calculated based on specific characteristics of the items, attributes/features. One of the most successful techniques uses correlations between content [26]. As the user classifies more items, performs actions based on the recommendations and offers new information, the system improves its accuracy [27].

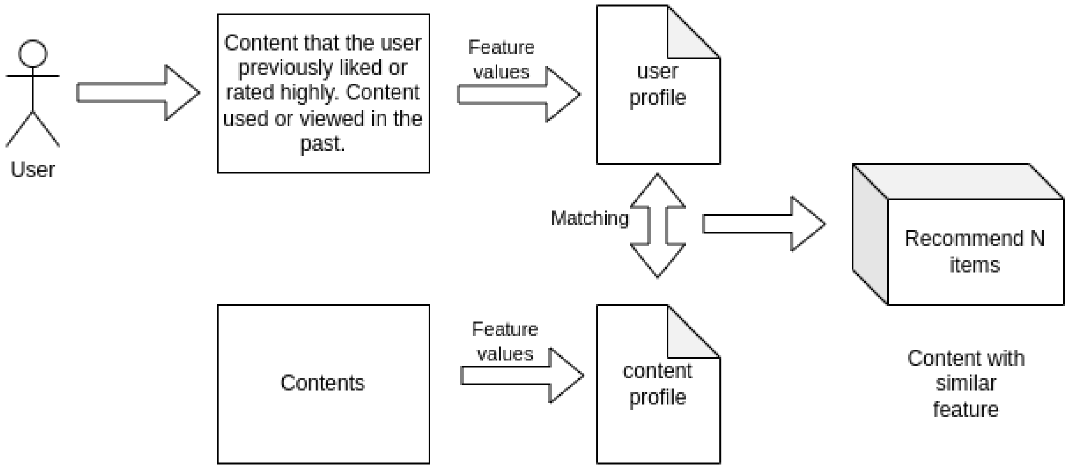



**Figure 2.9:** CBF process [28].

There are many obstacles in the way of CBF in recommendation systems. When there is no appropriate information in the content examined, it is difficult to categorize it, and in turn, difficult to offer pertinent suggestions [26]. In order to solve this problem, the Feature Weighting (FW) approach emerged, which consists of assigning different levels of importance to different features/attributes/resources [26] [27] [8].

However, the FW approach can give rise to the problem of overspecialization, i.e. a tendency to provide biased results. This is because when calculating the similarity of items they use only a few attributes, not using the most appropriate information [26]. The incorporation of ontological information into semantic analysis has been proposed by some studies with the aim of correcting this limitation. However, scalability and sparsity problems can arise when huge amounts of data are used in matrix calculations [26].

CBF systems are limited since they only take into account direct item associations. Network analysis is a viable strategy for raising these systems' performance. The relationships that bind objects together form patterns called networks. Examining and comprehending a range of relational and structural elements is part of network analysis. Network analysis clearly measures the variation in connection based on individual attributes. An advantage of this approach is that it allows conclusions to be drawn about connections on a large scale. Furthermore, because network analysis considers all kinds of relationships, it can extract meaningful information from even little amounts of data that goes beyond what an individual may perceive [26] [8].

Network analysis has been applied to recommendation systems, for example in social networks and e-commerce, resulting in better performance. Since similarity is only computed for users inside the target cluster, the utilization of clusters in the network lessens the possibility of information overload and enhances system scalability [8]. Studies demonstrate that the accuracy of forecasting user preferences is increased by clustering algorithms [26].

In a study conducted by Son and Kim [26], proposed a new CBF technique that employs a multiattribute network to properly reflect various attributes while determining correlations to recommend items to users. In addition, the suggested approach makes use of clustering and centrality algorithms to take into account the links between objects and identify the structural patterns of these interactions.

The approach in the previous study uses a network analysis, enhances the system's performance by improving both accuracy and robustness by recommending various items to the user based on varying criteria. Because the objects are characterized using a set of attributes, this method addresses the problem of overspecialization. This technique offers a plethora of information by taking into account the relationships among all the elements and analyzing the structural and indirect linkages between them, which aids in resolving sparsity issues in the recommendation process [26].

Kumar Sharma *et al.* [27] conducted a study where they developed several algorithms to find out which algorithm is suitable for the challenge of recommending products based on a selected product. The algorithms developed use Natural Language Processing (NLP) and Deep Learning to predict similar products.

The model selected to create the vector that holds information about the main feature, the title, was IDF-Weighted Word2Vec. Word2Vec is a technique that uses neural networks and reveals word embeddings. Google's pre-trained Word2vec model was used, employing only a subset of the matrix in order to include only words present in the titles, thus optimizing the process [8]. In addition, they applied Inverse Document Frequency (IDF) weights to each word and calculated the average to give greater importance to rare terms [27].

The Convolutional Neural Network (CNN) was used to create a feature vector from a product image. This was done using the pre-trained VGG-16 model, the architecture used to extract the features from the images [27].

After creating the vectors for each feature (e.g. title vector, image vector,...), the respective vectors for a product are combined, resulting in one large vector representing a single product. The recommendation of similar products was based on comparing the Euclidean distances between the vectors of all the products and returning the products with the smallest distance [27] [8].

In conclusion, CBF offers notable advantages in recommendation systems by taking into account the user's interests and preferences, making a personalized recommendation for the user [8].

The network analysis approach shows potential in expanding CBF's capacity by considering more detailed and interconnected binding patterns, including direct and indirect associations of items, and measuring variation in binding. This strategy, adopted in the study by Son and Kim [26], improved accuracy and robustness and resolved issues such as overspecialization, sparsity and scalability.

In addition, the study by Kumar Sharma *et al.* [27] was analyzed, which provides an innovative approach to predicting similar products using NLP and Deep Learning. To create the vector of annotations for each item, they combined the IDF-Weighted Word2Vec model with CNN's VGG16 model, and to recommend items, they compared the Euclidean distances between the vectors of the respective items. This method stands out as a particularly useful approach for comparing and representing various attributes, offering more refined recommendations [8].

Advances in recommendations, driven by the use of NLP and network analysis, increase accuracy and adaptability, allowing recommendation systems to better meet the wide range of user needs [8].

CHAPTER 3

# Methodology

This dissertation tells a narrative that ranges from acquiring data from people's daily lives, annotating it, segmenting events (grouping related data) and suggesting relevant moments.

The platform will be adapted so that it learns what is relevant to an individual. To do this, it is necessary to make event annotations that summarize the annotations on the images. The event annotations will then be analyzed and related to those that the user said were most relevant. After the analysis, it will be possible to predict possible important moments for the user, based on the information collected, and retrieve them.

The first tasks were to identify the research question, analyze the literature, write documentation related to the Literature Review and the methodology. On a more technical level, install the MEMORIA tool, explore event segmentation on the MEMORIA platform and annotate events.

An event contains a large amount of data, including several images, making it impractical to store annotations of all the images. It is therefore necessary to create a summary for the task of event annotation. In this context, the following approaches were analyzed:

- frequency of each concept in the images belonging to the event;
- the use of Large Language Model (LLM) tools;
- concepts occurrences average of the images belonging to the event (more information in Section 2.4.1);
- clustering of events with selection of cluster centers as representative embeddings (more information in Section 2.4.1);
- emphasize important images using transform encoders and learning a self-attention method, the outputs are mean pooled (more information in Section 2.4.1);
- passing through a linear layer to generate picture weights that represent the importance of each one, and then using the results to build a weighted average of embeddings (more information in Section 2.4.1).

After annotating the events, the next task is to prepare for data collection. In the data acquisition stage, some participants will be invited to use the MEMORIA application. They will be instructed to upload some of their personal images and then to classify the events generated according to how important they think the event is to them. In this way, a statistical collection is carried out that allows events to be associated with the user's rating (Figure 3.1).

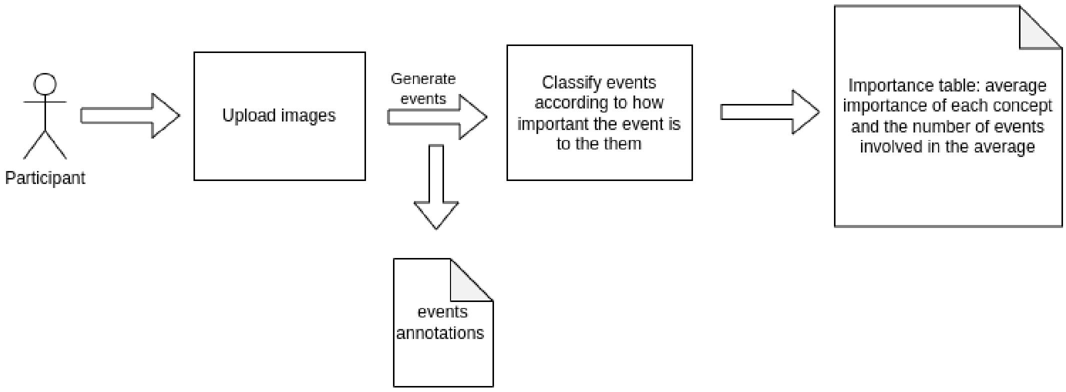



**Figure 3.1:** Data collection process.

In preparation for data collection, an experiment protocol will be written to guide participants during the experiment. Furthermore, it is necessary to adapt the system to the experiments to be carried out. A lighter version of MEMORIA will be developed, exclusively for data collection, excluding the heavier processing models (Figure 3.2). This way, participants won't have to install such a heavy application on their computers.

Basically, this new version will keep the login and account creation features of the current version of the system. The feature for uploading images will be a little different. A webservice will be developed for when the user uploads the images, and instead of processing the images on the computer itself, it will send them via an API to IEETA's computer for processing. The images will not be saved on IEETA's computer disk in order to preserve users' privacy, nor will any information from the images. The endpoint will return a file with the annotations of the images, the result of the processing. This webservice is designed to avoid computational demands on participants' devices [8].

Event segmentation functionality will also be available in this version. The computational requirements of the current algorithm will have to be examined, and ways will have to be found to deal with the requirements and with the possible absence of location information in the images [8].

Still in the light version, a new feature will be introduced in which the user will be able to classify the events in the last layer of event segmentation, the Image Similarity layer (Figure 2.8), based on how relevant each event is to them. The definition of event segmentation in lifelogging, as well as the application of the process in MEMORIA can be consulted in Section 2.4. This feature will store the average importance of each concept and the number of events involved in the average, taking into account all the classifications made by the user; and the importance attributed to each event by the user [8].

In addition, a new page will be added, the most important moments page. This page will display the events previously classified by the user, ordered by degree of importance.

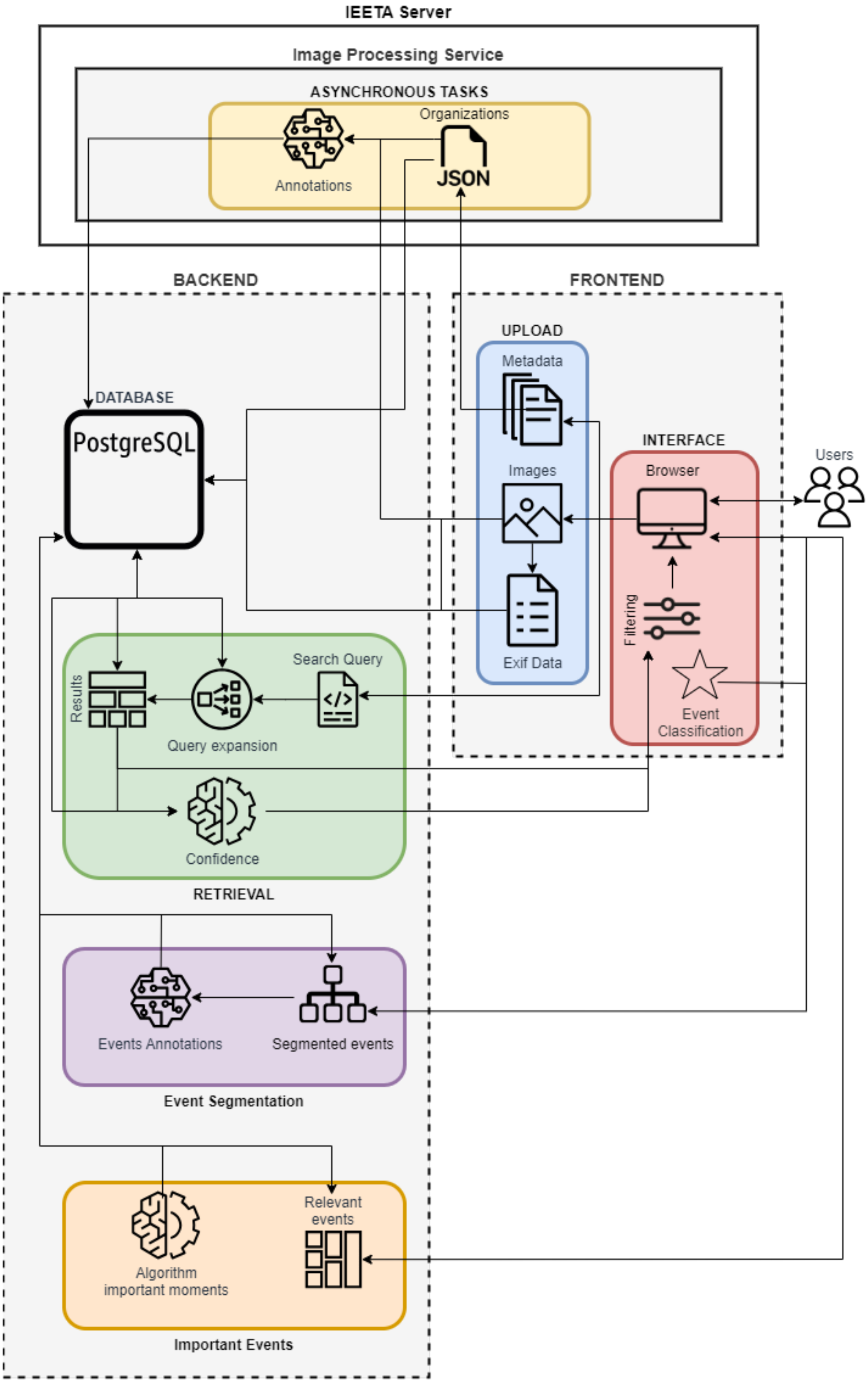



**Figure 3.2:** MEMORIA architecture, with existing services and those that will be added or adjusted during this work (Events Annotations, Important Events and Image Processing Service) [29].

Finally, in order to make it even easier to install the new version of MEMORIA on participants' computers, a multi-container docker will be developed. This docker will include the configuration of all the essential services, ensuring a quick and easy installation of all the dependencies needed for the system to work.

At the end of data acquisition, we will have the average importance of all the concepts for each user. Next, some tasks were planned in order to achieve the algorithm for recommending important moments, which were not carried out due to the limited time required to develop the work. The next expected task would then be processing the collected data, analyzing the results and making any changes to better suggest events.

The next step would be to implement the algorithm for recommending important events, Figure 3.3, with a training, testing and parameter adjustment phase. It would compare the annotations of the events previously classified as relevant with the annotations of the remaining events, returning those with the greatest similarity. For this comparison, some NLP techniques that are used in CBF-type recommendation systems were collected:

- CBF algorithm based on a multiattribute network (more information in Section 2.5);
- for the creation of the vector of annotations for each item: combination of the IDF-Weighted Word2Vec model with the CNN VGG16 model; for the recommendation: comparison of the euclidean distances between the vectors. (more information in Section 2.5);

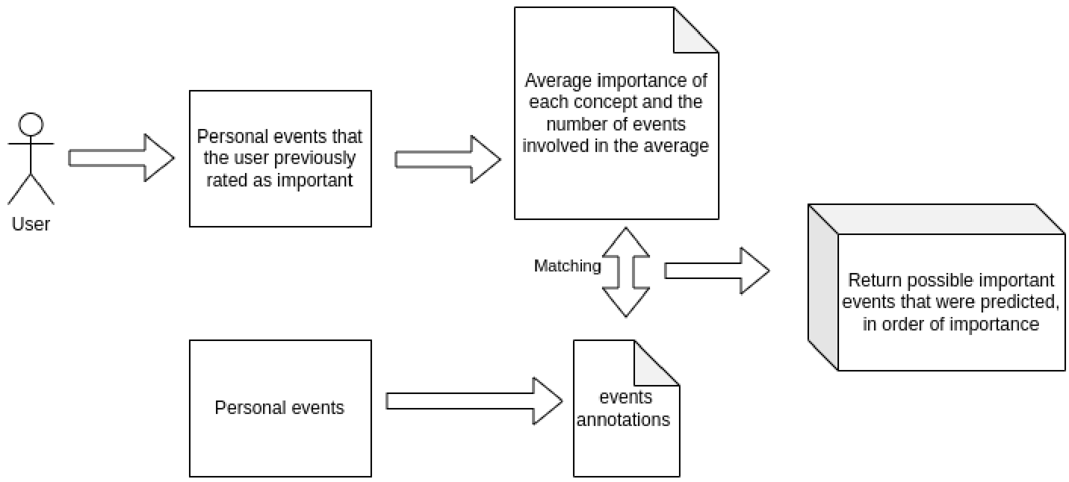



**Figure 3.3:** Process of recommending events (with CBF).

The system to be developed is supposed to be just the start of identifying and recovering the user's most important moments, creating the conditions for more detailed work in the future.

Over the course of the work, adaptations will be made to the frontend and documentation will be drawn up. Participation in the LSC 2024 competition is planned.

## 3.1 SWOT Analysis

Finally, a SWOT analysis of the dissertation will be presented to identify its strengths, opportunities, weaknesses and threats.

**Strengths** [8]:

- The application already provides easy access to daily photos, as well as events/moments, helping with efficient memory retrieval. These created events will be the moments used for classification by the user, as well as for retrieving the important moments, thus already having the retrieval unit;
- The user will be able to see the moments that are really relevant to them, that they really want to remember, and not insignificant moments;
- Integration of episodic and autobiographical memory, contributing to memory improvement;
- A new organization of lifelog data to improve the user experience.

**Weaknesses** [8]:

- Current technologies don't allow us to select the important moments, which can result in events being displayed that the user doesn't consider relevant, or even wants to forget;
- Potential privacy issues because lifelog data is intimate in nature.

**Opportunities** [8]:

- There are currently many advances in NLP and Artificial Intelligence, areas in which this dissertation will fall. These advances can help make the selection of important moments more efficient;
- Possibility of increasing interest and creating a new challenge related to memory aid, in this context of lifelogging, creating conditions for future more detailed work.

**Threats** [8]:

- Ethical issues with data privacy;
- Potential psychological effects of revisiting specific memories, requiring careful navigation.

CHAPTER 4

# MEMORIA Adaptation for Research

The first stage of development for this dissertation was the adaptation of the MEMORIA system for the data collection that will be carried out. Data collection is necessary, in turn, to help implement the algorithm for recommending possible important moments (objective of the study).

The objectives of adapting the MEMORIA application are to create a system that can be used by outsiders on their own computers, minimizing technical requirements and ensuring that it is as lightweight as possible. To adapt the system to provide the best possible interaction with the user, and to have functionalities for analyzing and recovering memories. The aim is to make the necessary adaptations in order to facilitate and simplify data collection as much as possible, making the process more accessible and efficient for participants.

This stage is not only crucial for the development of the algorithm, it is also important for bringing MEMORIA to interact for the first time to an outside audience. It is essential to evaluate how the application behaves for end users, checking that it is user-friendly and that users know how to navigate the system without doubts and without needing assistance. After all, the product is aimed at end users, and it is vital to provide an interaction experience that is fluid and easy [8].

MEMORIA was separated into two distinct services, MEMORIA Light and MEMORIA Server (Figure 4.1). MEMORIA Light is a lighter version of MEMORIA, without the image processing models that demand high GPU memory capacity and require high computational requirements. This version is designed to run easily on any computer used by experiment participants. MEMORIA Server is a service available on a server, where images can be sent via an API to be processed on the server, by complex image processing models. Processing results could also be obtained via the API. Images are never stored on the server, ensuring the privacy of user data.

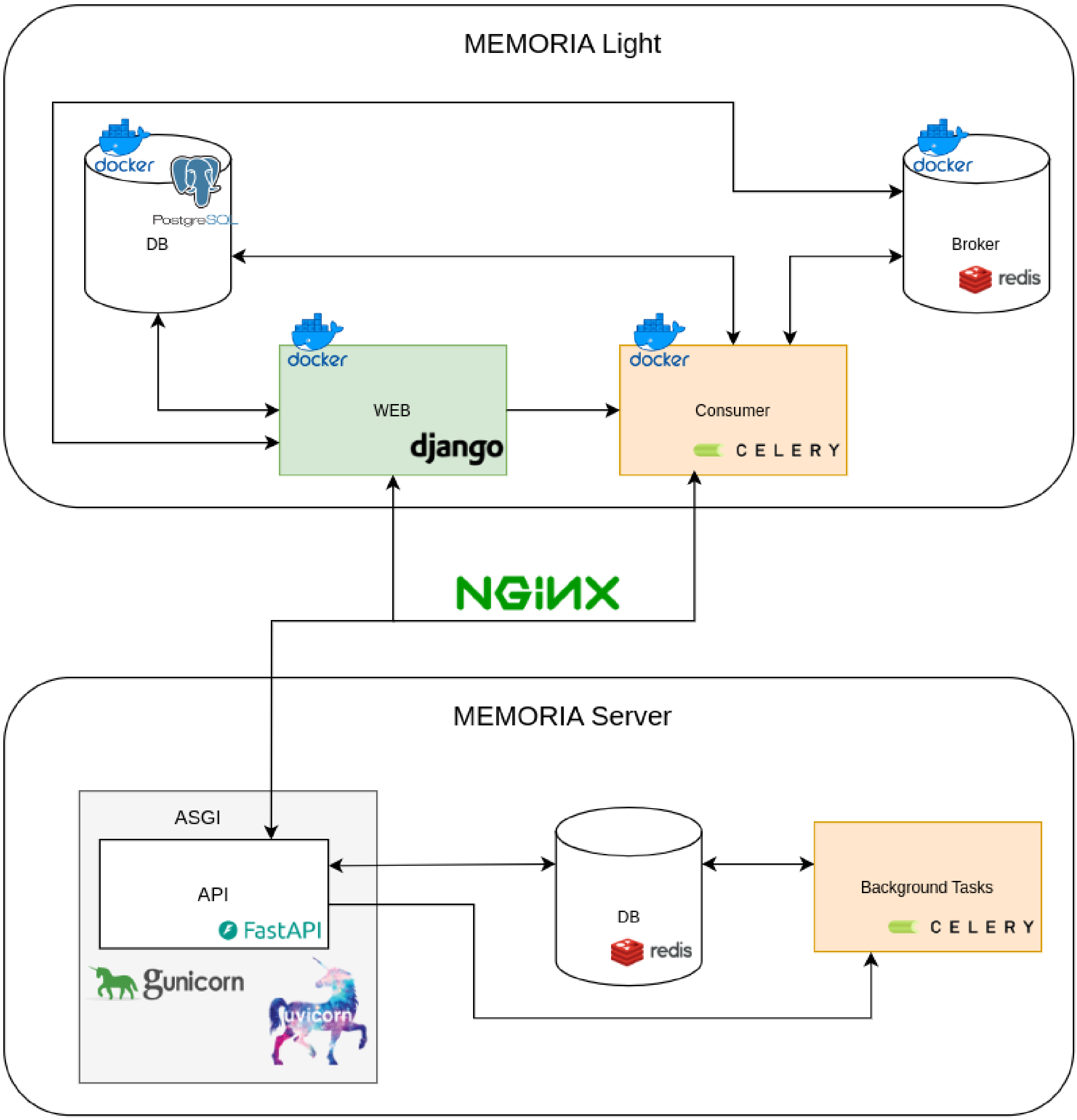



**Figure 4.1:** MEMORIA architecture.

## 4.1 MEMORIA Server

On the MEMORIA Server side an API was developed together with a database (Figure 4.1).

After a careful analysis of the technologies that could be used in the development of the API, FastAPI [30] was chosen. The main programming language used in the MEMORIA backend is Python. That's why it was decided that the API should also be developed in Python, for consistency and compatibility. In addition, all the code for the image processing models is in Python, and as this code will now be on the MEMORIA Server side, it makes sense for the API to also use this language to facilitate integration [8]. Three technologies were therefore considered for this purpose: Flask [31], FastAPI and Django Rest Framework [32].

| | Flask | FastAPI | Django REST Framework |
|---|---|---|---|
| **Type** | Micro web framework | Asynchronous web framework | Powerful Django extension for building RESTful APIs |
| **Asynchronous** | No | Yes / concurrency | No |
| **Performance** | Efficient for small and medium applications | High performance, ideal for data-heavy APIs | Moderate, suitable for most use cases |
| **Scalability** | Less than FastAPI and Django | Highly scalable | Strong scalability |
| **Ease** | Very simple, light and very easy to use | Easy to get started, especially for those who know Python | More complex |
| **Project size** | Small and medium | Medium and large | Larger projects with complex requirements |

**Table 4.1:** Comparison of the characteristics of Flask, FastAPI and Django REST Framework [33].

Table 4.1 shows the characteristics collected for the technologies considered for the new API. The choice of FastAPI was motivated by its high performance and scalability, essential characteristics for dealing with large volumes of complex data, which is the case where we are going to deal with images, which will be processed by extremely heavy image processing models. FastAPI's ability to support asynchronous operations was a decisive factor, as it allows image processing tasks to run in the background, simultaneously with other tasks, optimizing the time it takes to obtain results. In contrast, both Flask and Django have no asynchrony. In addition, Flask is not known for its performance and scalability, while Django, although scalable, has moderate performance. FastAPI is therefore the technology that most meets the necessary requirements, and is also recognized for its ease of use and automatic generation of documentation [8]. It is recommended for medium and large projects and for real-time and high-concurrency projects, where MEMORIA fits in [33].

Since image processing results are not determined instantaneously, but are processed in the background due to the possibility that they may take some time to complete, a Redis [34] database was created on the MEMORIA Server side. Thus, the results are temporarily saved while they are not sent to MEMORIA Light. It's important to note that at no point are the images stored on the MEMORIA Server, only the results of the analysis, such as annotations and labels, are kept temporarily [8].

Redis was the technology chosen to temporarily store the results of image processing, as a database was needed that would allow data to be added, accessed, edited and deleted quickly and accessibly [34]. These requirements are fundamental, considering that image processing itself can be a time-consuming task, so agile manipulation of the data in the database is crucial to the system's efficiency. Redis offers excellent performance, providing data access with low latency and high throughput. In addition, it is flexible, simple to use, and offers high availability and scalability, meeting the requirements of the necessary database [34].

### 4.1.1 Implementation

The API developed has 4 asynchronous endpoints, the endpoint «/process-image/{user_key}/{img_id}», «/get-annotations/{user_key}», «/confirm-receipt» and «/get-task-status/{user_key}/», allowing the endpoints to be invoked simultaneously and executed concurrently (Figure 4.2).

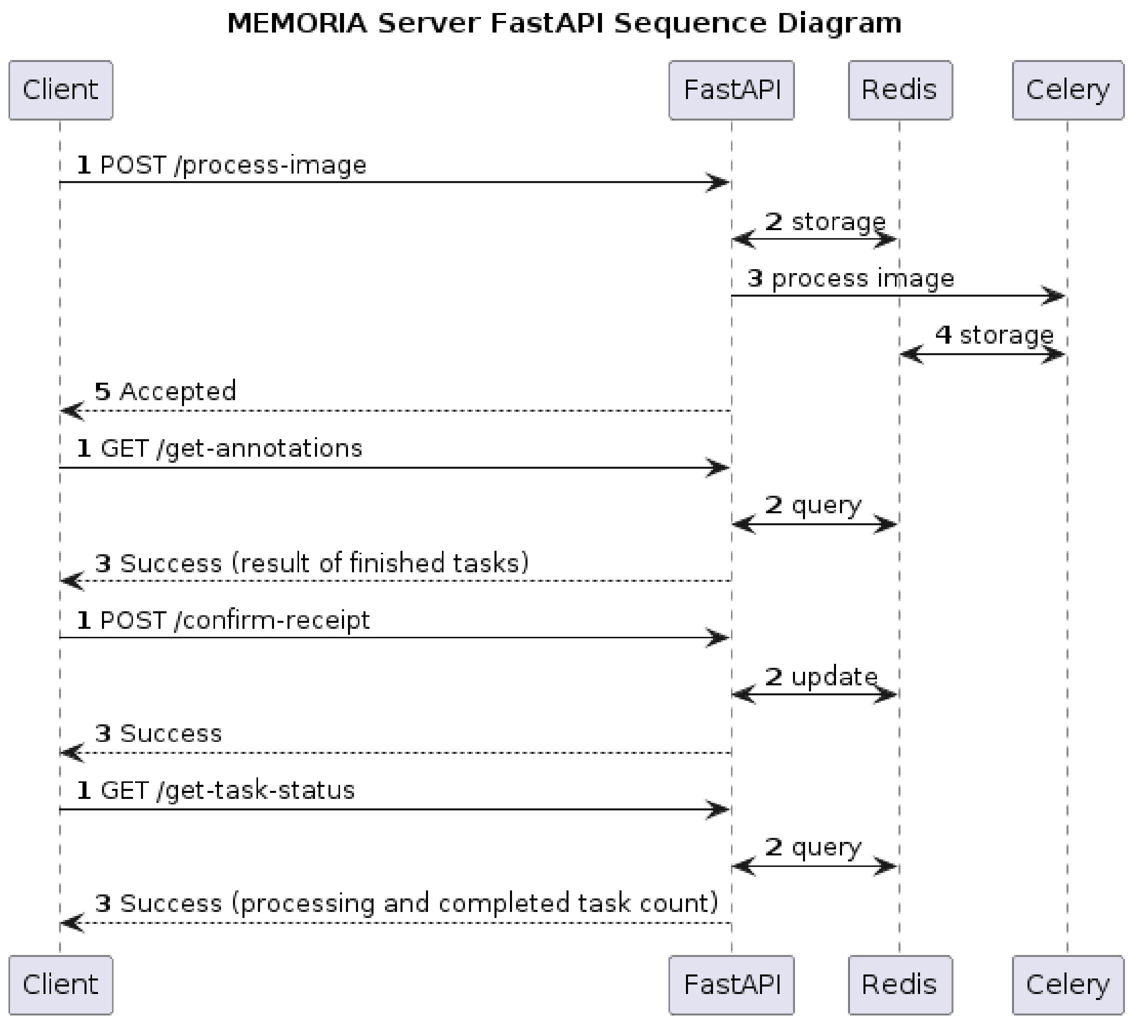



**Figure 4.2:** MEMORIA Server FastAPI Sequence Diagram.

The «/process-image/{user_key}/{img_id}» POST endpoint (Figure 4.3) was created for a user to send an image they own to be processed. This endpoint receives three parameters, in the URL path «user_key» which is a unique identifier of the user and «img_id» which is a unique identifier of the image of the given user. It also receives in the body of the request the content of the image in bytes to be processed (Figure 4.2).

The image processing logic was implemented in Celery tasks. Celery [35] is a background-running asynchronous task job queue system. The use of Celery is essential when analyzing images, as it is a process that costs a lot of time and resources. There are seven Celery tasks, each responsible for processing the image using a specific model (i.e., there are seven distinct models):

- BIQA, pre-trained model, predicts the Mean Opinion Score of the image quality;
- Places 365, model using a CNN for scene recognition and generic features of deep scenes for visual recognition;
- YOLOv7, model for object detection;
- facebookresearch_WSL, ImageNet Image classification model;
- Image Caption CLIP model, model for generating captions for images;
- CRAFT, model for text detection in images;
- GRiT model for Object Understanding.

The image processing mechanism is practically the same as that of the previous version, it uses the same processing models, with only a few adaptations to split MEMORIA into MEMORIA Server and MEMORIA Light. In the Image Caption CLIP model task, the result not only has the caption generated, but also the CLIP embeddings, the encoded features that the model has determined for the respective image. This addition is necessary for event segmentation, so it won't be necessary to run the CLIP model again when segmenting events, thus avoiding another call to the MEMORIA Server.

The «/process-image/{user_key}/{img_id}» endpoint processes the image asynchronously, running the Celery tasks in parallel and in the background. The tasks are asynchronous, so they can be executed simultaneously without blocking the main execution. During this process, Redis stores the number of tasks being processed, the number of tasks completed, the status of each task (whether it is «processing», «succeeded» or «failed») and the results of the tasks (model processing results) as they finish. After starting processing, it returns a JSON response with a message indicating that the image processing has started and a status code of 202 (Accepted) (Figure 4.2 and 4.3).

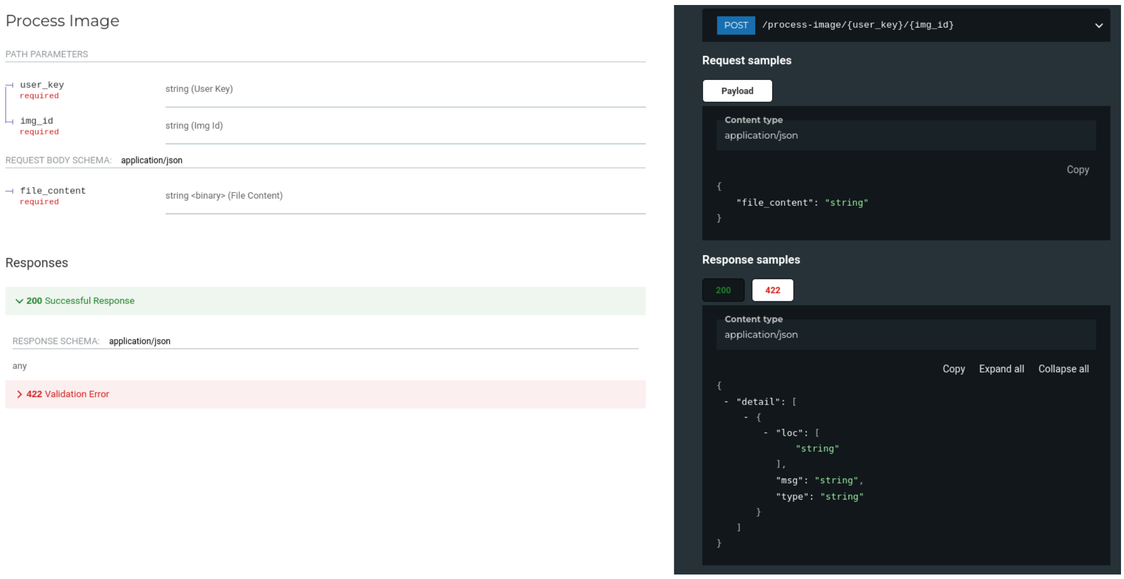



**Figure 4.3:** POST endpoint documentation «/process-image/{user_key}/{img_id}».

The GET endpoint «/get-annotations/{user_key}» (Figure 4.4) returns the results of the user's tasks in question, which have already finished. The endpoint receives the user's unique identifier as a parameter and provides the annotations generated by the user's successfully completed tasks, as well as the results of the failed tasks. This information is collected in Redis. It also returns a list of the identifiers of the tasks it is returning (Figure 4.2).

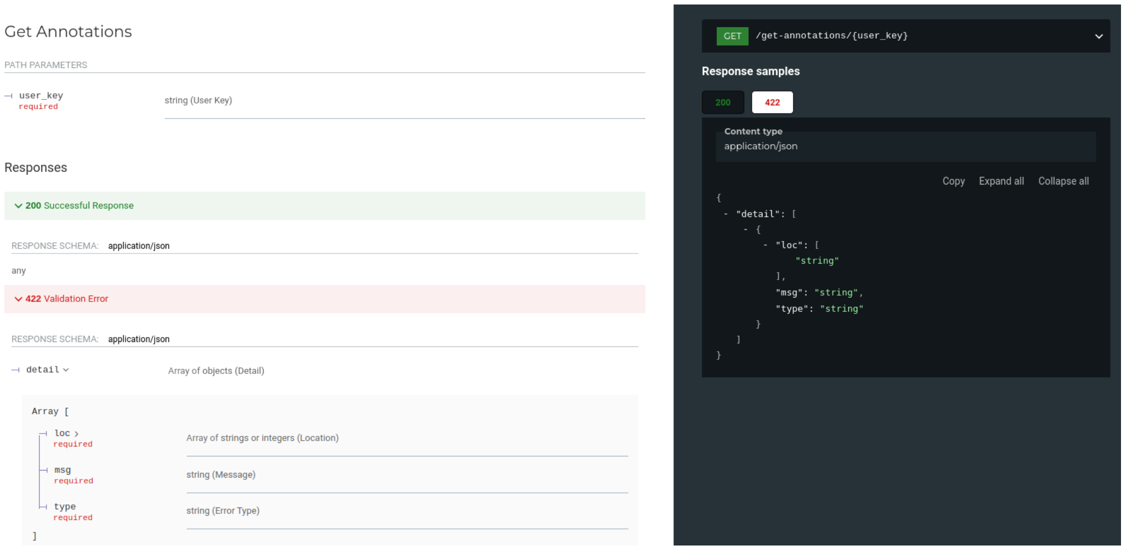



**Figure 4.4:** GET endpoint documentation «/get-annotations/{user_key}».

The «/confirm-receipt» POST endpoint (Figure 4.5) is used to confirm receipt of the annotations from the «/get-annotations/{user_key}» endpoint, and to delete the received tasks from the Redis. It receives as parameters «received», a boolean indicating whether it has received the annotations or not, and the parameter «keys_completed», the list of tasks received at the «/get-annotations/{user_key}» endpoint. If the value of «received» is true, the received tasks are deleted from Redis and the number of completed tasks in Redis is updated (Figure 4.2).

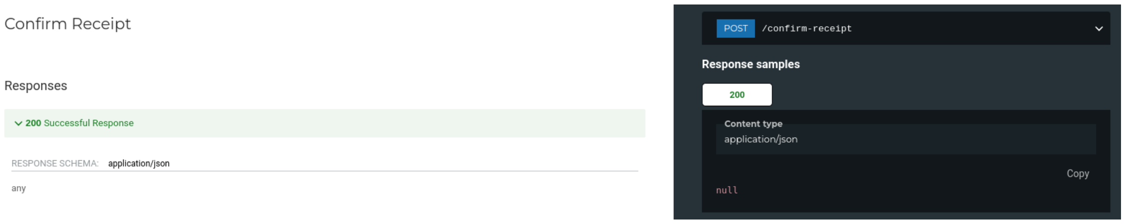



**Figure 4.5:** POST endpoint documentation «/confirm-receipt».

The GET endpoint «/get-task-status/{user_key}/» (Figure 4.6) returns the count of completed and in-process tasks for the user with the identifier passed in the parameter. The endpoint allows the user to see in MEMORIA Light how many tasks are complete and in process on the MEMORIA Server. In this way, the user knows if they have annotations to obtain on the server, or if there are still images to be processed (Figure 4.2).

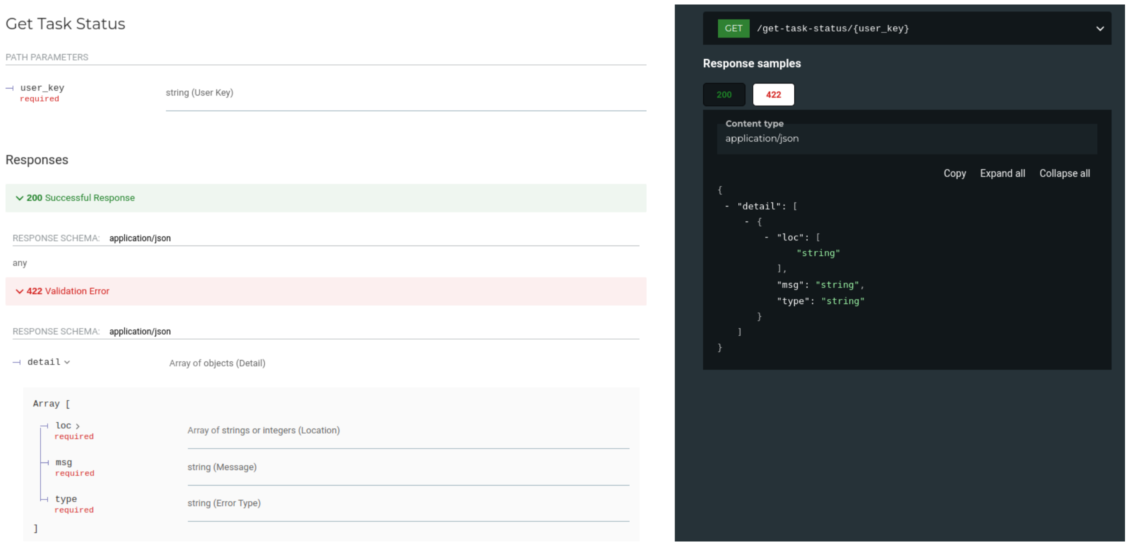



**Figure 4.6:** GET endpoint documentation «/get-task-status/{user_key}/».

Overall, the API endpoints developed enable efficient and controlled interaction with the scalable MEMORIA Server. This ensures that users can send image content, monitor the processing of these images, retrieve the results and confirm receipt of these results in an organized and robust manner.

### 4.1.2 Deployment

After developing the MEMORIA Server, the next step was to deploy the FastAPI with NGINX [36] and Gunicorn [37], and generate and use the free SSL certificate (Figure 4.1). This step is necessary for the service to be available for data collection, allowing participants to send their images for processing to a server capable of processing the images and returning the results.

Gunicorn is a Python Web Server Gateway Interface (WSGI) HTTP server designed to run Python web applications. Python programmers may easily create asynchronous, high-performance web apps with Uvicorn, an implementation of the Asynchronous Server Gateway Interface (ASGI) server. NGINX is a high-performance web server and reverse proxy server for HTTP and HTTPS traffic, also used for delivering static content and load balancing. Python web applications are typically deployed using NGINX, Gunicorn, and Uvicorn in tandem as an ASGI server and reverse proxy [38].

The first configuration was an ASGI server using a combination of Gunicorn and Uvicorn (Figure 4.1). The FastAPI application can be processed and asynchronous requests can be handled by the configured ASGI server. In a production setting, this combination provides an effective means of supporting the application [38] [8].

Incoming requests are forwarded to the FastAPI application server via the NGINX server, which is set up as a reverse proxy (Figure 4.1). This makes it possible to divide up duties, which enhances performance and security [38] [8].

Finally, a domain name was created for the MEMORIA Server API. A free SSL certificate was generated for this domain using Let's Encrypt [39] and Certbot [40]. The SSL certificate allows HTTPS requests to be made successfully [38] [8].

## 4.2 MEMORIA Light

MEMORIA Light, as previously mentioned, is a lighter version of the previous MEMORIA. It stands out for not having image processing models and, in turn, not requiring high requirements for its execution. This section describes the changes made to the old functionalities, as well as new functionalities implemented to make data collection possible.

### 4.2.1 Upload Images

In the development of MEMORIA Light, one of the first changes was to the image upload functionality (Figure 4.7). When the user uploads an image, the system creates a unique identifier for it, based on the name of the image and the name of the user. It then attempts to extract the image creation date and time from the image's EXIF metadata information. Given the extracted date and time, the system associates the image with a specific part of the day (morning, afternoon, evening, etc.). The image is then stored in the database along with the identifier created, the date and time extracted and the part of the day determined.

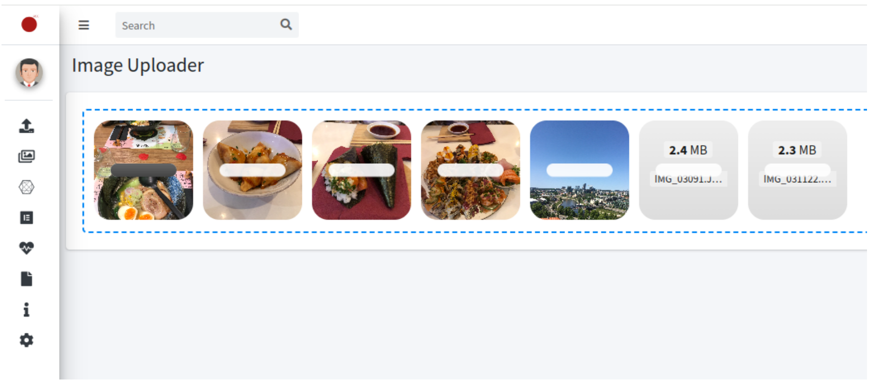



**Figure 4.7:** Image upload page saving images.

This described process, remains mostly the same as in the previous version of MEMORIA. However, below are the steps that have undergone changes.

A new Celery task has been added to assign location to the image. This task runs asynchronously and tries to extract the geographical coordinates (latitude and longitude) from the EXIF metadata. After obtaining the geographical coordinates, the task determines the address of the location and its timezone, associating both properties with the image. Based on the timezone, the system stores the local date and time (in the timezone of the image location) and also the date and time in Coordinated Universal Time.

Also in the Celery task, a search is made for images that took place on the same day as the image that has just been uploaded, in order to create connections between them. The connections are reflected in the storage, in each image, of the image that happened before and after. This process of creating connections between images already existed in the previous version, but the difference is that before the search was limited to images that occurred 4 hours before and 4 hours after, now the search covers all images from the same day.

This change was necessary because the previous version was designed for a context of passively collecting images at short intervals, for example, every 30 seconds. In this scenario, it was efficient to limit the search to 4 hours before and after, as we were sure to have photos, and so avoid dealing with a large number of images unnecessarily, speeding up the process. In this case, for the experiment, we are in a context where the collection of images is carried out intentionally in principle, in an active way, where the time intervals between images can be longer than 4 hours. To make sure that all pertinent links are made, the search has been expanded to include all photos taken on the same day.

Finally, still in the Celery task of the locations, another Celery task was created and called up here. This last task is used to create connections between locations, for each location to save the previous and next location.

After the Celery task for the locations, a new Celery function designed to send the contents of the image to the MEMORIA Server, which in turn processes the image, is executed. This function replaces the local execution of extremely heavy image processing models. In order to optimize the image's size for transmission without sacrificing its visual quality, it is first resized to a maximum size of 1024x1024 pixels. It then makes a call to an external API, the MEMORIA Server API, via the POST endpoint «/process-image/{user_key}/{img_id}» (Figure 4.3). Sending the image data in byte format as JSON in the body of the request, along with the username and the image instance information in the URL path. The MEMORIA Server receives this request, starts processing the image in the background and responds with a status code of 202 (Accepted).

This new adaptation to image uploading makes it possible to run the system on any computer, without the need for high GPU memory capacity or other demanding computing requirements. This is because the image processing models have been migrated to a separate system, MEMORIA Server.

### 4.2.2 Get Image Processing Results

Another adaptation was the procedure for obtaining and saving the image processing results determined on the MEMORIA Server. A card has been added to the daily image display page for this purpose (Figure 4.8).

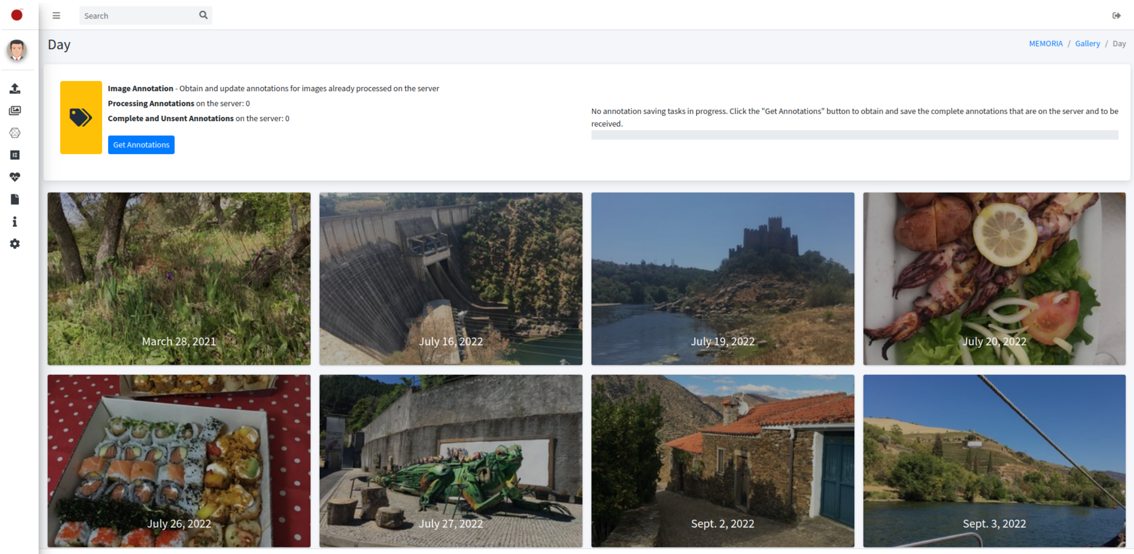



**Figure 4.8:** Daily image display page.

The card shows two values «Processing Annotations» and «Complete and Unset Annotations», as shown in Figure 4.9. The «Processing Annotations» value represents the number of annotations that are still being processed on the server, i.e. results that have not yet been finalized. The «Complete and Unset Annotations» value indicates the number of annotations that have been fully processed on the MEMORIA Server and already have results, but which the user has not yet received in the system. These values are updated periodically by sending requests to the GET endpoint «/get-task-status/{user_key}/» (Figure 4.6) of the MEMORIA Server API, passing the username of the authenticated user as a parameter to obtain the status of the tasks. The counts of the user's completed and in-process tasks are then updated and displayed on the daily photo gallery page.

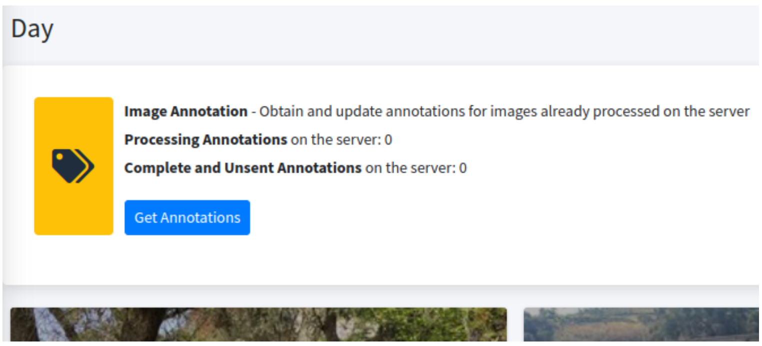



**Figure 4.9:** Information about the annotations on the server, on daily image display page.

These two values are essential for the user to have an idea of what is happening on the MEMORIA Server. To know how many processing tasks are still in progress, and to know which tasks have already been completed and whose results are already available for obtaining.

A «Get Annotations» button has been developed (Figure 4.9), which when pressed invokes a Celery function that has been developed to obtain and store, in parallel with the main execution of the system, the analysis and annotations of the images already determined in MEMORIA Server. In this Celery function, the first step is to call the GET endpoint «/get-annotations/{user_key}», from the MEMORIA Server API, passing the username of the authenticated user as a parameter and waiting for a response.

If the response code is 200 (ok status code), it extracts from the response the annotations and results sent and the list of completed task identifiers. After this extraction, a POST is made to the «/confirm-receipt» endpoint (Figure 4.5) of the MEMORIA Server API, sending as a parameter a boolean true in the «received» field, and the «keys_completed» parameter with the list of tasks received. This POST confirms the reception of the annotations to the MEMORIA Server.

If the annotations received are empty, the Celery task ends up sending an alert to the interface, with the message «No annotations to get and save on the server!». Otherwise, i.e. if the results received have content, a new Celery task is created for each result, allowing parallel execution. This new Celery task, in turn, generates another task that varies according to the model that originated the result. This last task is responsible for saving the result in the database according to the specific model. The way it saves the results of each model, and what it saves, is the same as in the old version of MEMORIA, differing only for the CLIP model.

In the CLIP model, the result not only has the caption generated (as in the previous version) but also the CLIP embeddings. For this reason, it was necessary to edit the database table that stores the information for each image, adding the «emb_clip_encode_img» field. Thus, the Celery function for saving the results of the CLIP model was adjusted to update the «emb_clip_encode_img» field of the corresponding image with the CLIP embeddings received.

Throughout this process, as Celery tasks are created, the status of each task is stored in the database, whether it is in progress, failed or completed. These status, together with the use of websockets, make it possible to keep track of the progress of tasks in obtaining and saving annotations, which is updated and communicated in real time to the user interface. The progress in question is shown on the interface via a progress bar (Figure 4.10).

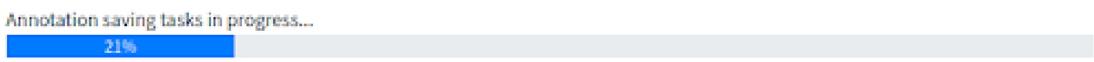



**Figure 4.10:** Progress bar for saving annotations tasks, on the daily image display page.

At the end of each Celery task, a message is sent to the websocket indicating the progress of the tasks, calculated based on the number of tasks in each state. The websocket, in turn, updates the progress bar on the interface. This functionality allows users to monitor the status of the tasks for obtaining and saving annotations and to be aware of any problems that may arise during the process.

With these three functionalities, the display of MEMORIA Server's «Processing Annotations» and «Complete and Unset Annotations» values, the «Get Annotations» button to obtain and save the results and the progress bar indicating the progress of the tasks to save the results obtained allow the user to be in tune with what is happening on MEMORIA Server. They allow the user to start obtaining and saving finished results, to know when there are no more results to be received and to check that all the results have already been analyzed and saved in MEMORIA Light.

In short, these features keep the user constantly informed, in real time, about the current status of the processing of annotations on the server and tasks in MEMORIA Light. The transparency of the information and the continuous reporting of the entire process (through the visualization of data, status, progress, ...) are key to enlightening the user about what is happening, improving interaction with the system. In addition, running tasks in the background is efficient and allows the user to continue navigating the system without interruption, offering a more fluid experience.

### 4.2.3 Create Events

The event segmentation process has undergone some changes, the description of the process in the previous version can be read again in Section 2.4. On the event creation page (Figure 4.11), it is possible to press a button to start segmenting events by creating Celery tasks, just like in the previous version. First, a Celery task is created to create the events in the first layer, «Day», at the end of which another task is created to create the events in the «Parts of Day» layer, at the end of which a task is created to segment the events in the «Location» layer, then «Environment» and «Similarity». At the end of the «Similarity» event creation task, the MEMORIA Light version added the creation of a new task, invoking a new Celery function designed to annotate all similar events.

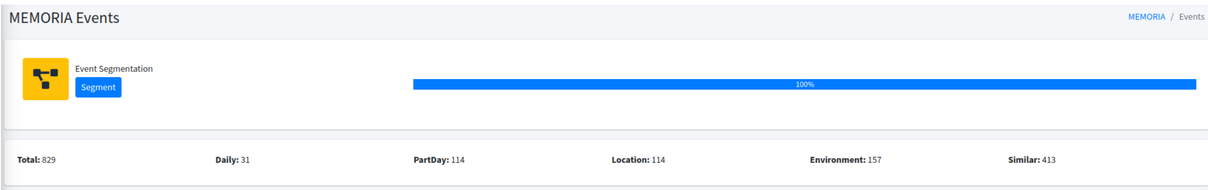



**Figure 4.11:** Event creation page.

The event segmentation logic is practically the same as in the previous version. One of the adjustments made was to access the CLIP embeddings of an image, known as encoded features. In the previous version, the CLIP model was re-run to obtain the image embeddings, in MEMORIA Light it's only need to access the «emb_clip_encode_img» attribute of the image in question in the table that stores the information for each image.

Another adjustment was in the calculation of the cosine similarity between two images, using embeddings. In the old version, this calculation was done using a heavy library that required a complex installation process and consumed a lot of GPU memory. In the current version, the cosine similarity is calculated manually, applying the formula directly, without access to any library.

The last adjustment was the addition of a new Celery task which annotates all similar events. This task is carried out after the task of creating similar events, and is the last task in event segmentation.

In the methodology chapter, Chapter 3, some different approaches to annotating events were collected and listed. The approach adopted for annotating similar events was to save the frequency of each concept for each event, i.e. the number of images of the event in which the concept appears. This was the method chosen as it is the simplest, which is ideal for the initial phase of the problem under study, a problem that is being studied and analyzed for the first time. The annotation of events is stored in a database table created for this purpose, the «EventAnnotation» table. This table has the attributes «concept», «frequency» and the unique identifier of the event to which it refers (Figure 4.12).

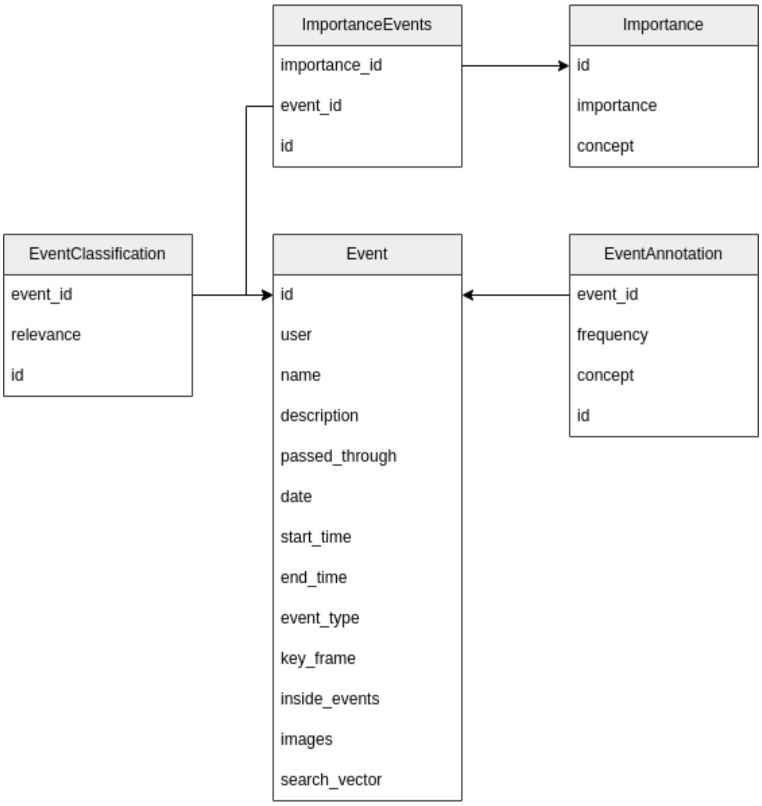



**Figure 4.12:** Tables in the database relating to Events.

Just like the tasks for obtaining and saving annotations (Section 4.2.2), on the event creation page it is possible to view the progress of event segmentation in real time via a progress bar (Figure 4.11) located next to the event creation button. This functionality is possible thanks to the use of websockets and by monitoring the status of each event segmentation task, which is stored and updated in the database during the segmentation process.

### 4.2.4 Event Classification

We now move on to a new feature implemented in MEMORIA Light, the classification of similar events. Similar events belong to the last layer of the event hierarchy (Figure 2.8), and are believed to be the most similar to real life moments. Because of this similarity, the ability for the user to classify a similar event according to how important the event is to them was implemented.

On the similar events page, a page that already existed in the previous version that displays all similar events from the authenticated user, was where the classification functionality was added. In each similar event, there is a star rating bar available for the user to assign an importance from one to five, if desired (Figure 4.13). The user can edit the rating by clicking on the new importance desired and can also remove a rating by clicking again on the star corresponding to the event's current rating.

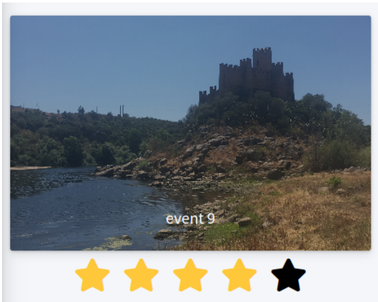



**Figure 4.13:** Example of a similar event with the rating bar.

This functionality has also been implemented, on the similar event page, which shows a specific similar event, what is inside a similar event, for example, the images belonging to that event, the name of the event, etc. The functionality is also carried out by a star rating bar, also allowing to add, edit and remove a rating (Figure 4.14).

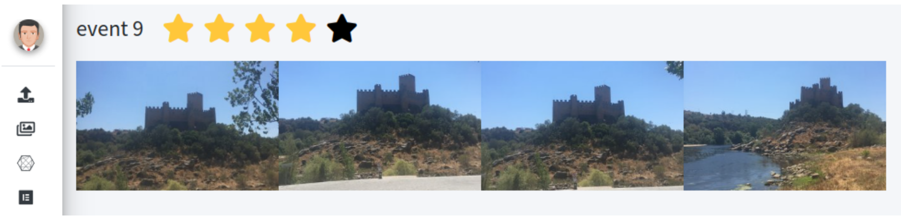



**Figure 4.14:** Example from inside a similar event with the classification bar.

In order to support the new classification feature implemented in MEMORIA Light, it was necessary to create three tables in the database: «EventClassification», «Importance» and «ImportanceEvents» (Figure 4.12).

The «EventClassification» table is only used to store the classifications that the user assign to events. The absence of an event's classification does not imply that its classification is 0. As long as a user does not rate an event, it is not assumed that the event's rating is 0, it simply has no value assigned to it. The attributes of the «EventClassification» table are «relevance» and the unique identifier of the classified event (Figure 4.12).

The «Importance» table serves to store the importance of each concept for the user, taking into account the classifications made in the events. The importance of each concept, for a given user, is calculated by the following formula:

$$\mathrm{IM}(c) = \frac{\sum(\mathrm{Classification}_i \times \mathrm{Frequency}_i^c)}{\sum \mathrm{Frequency}_i^c} \tag{4.1}$$

where $c$ is the concept being evaluated. The variable $i$ is a classified user event that contains concept $c$. The term $\mathrm{Classification}_i$ is the classification given to event $i$ by the user. $\mathrm{Frequency}_i^c$ is the frequency of concept $c$ in event $i$. $\mathrm{Frequency}_i^c$ is equivalent to the number of images of event $i$ that contain concept $c$, and can be accessed in the «EventAnnotation» table (Figure 4.12).

In other words, the metric represents the weighted average of the importance of the concept, considering its frequency in the classified events. It is calculated by multiplying the classification of the event by the frequency of the concept in that event, summing it up for all the user events classified and containing the concept. Finally, it is divided by the sum of the frequencies of the concept in the classified events.

The divisor is the sum of the frequencies of the concept in the classified events, and not in all events, to ensure that unclassified events do not influence the metric. There are several reasons why a user may not have classified an event, and it cannot simply be assumed that the unclassified event is not important. In addition, only annotations that have at least one classified event have an importance value, again, we shouldn't assume that an annotation has 0 importance just because it isn't in classified events.

The «Importance» table therefore contains two attributes, the «concept» attribute and the «importance» attribute. The «importance» attribute is the value calculated in the metric described above (Equation 4.1) (Figure 4.12).

The «ImportanceEvents» table results from the many-to-many relationship between the «Event» and «Importance» tables. An event can be associated with several importance calculations, and an importance can have several events involved. For this reason, the «ImportanceEvents» table is used to store associations between classified events and importances in the «Importance» table. That is, to associate with an importance, the classified events that are involved in the calculation of that importance. This table has the unique identifier of an event and the unique identifier of an importance as attributes (Figure 4.12).

The «EventClassification», «Importance» and «ImportanceEvents» tables (Figure 4.12) are updated whenever a user adds, edits or deletes the classification of an event.

A new page has been developed for MEMORIA Light, the important events page (Figure 4.15). This page displays the user's events that have been classified, allowing them to view the classifications assigned to each event previously and to edit or remove classifications if they wish.

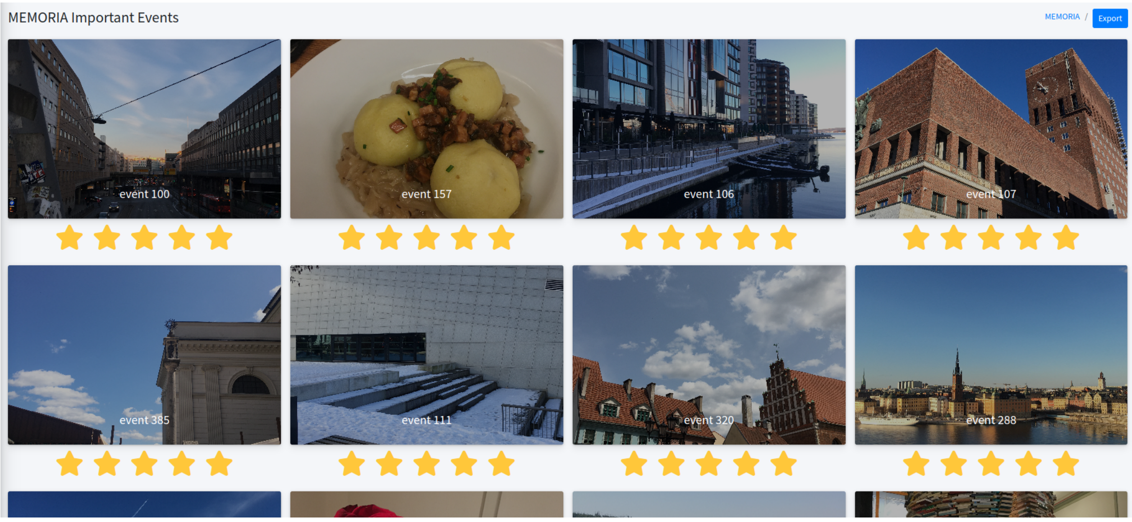



**Figure 4.15:** Important Events page.

On the Important Events page, there is a button called «Export» located in the top right-hand corner of the screen (Figure 4.15). The button extracts a CSV file with the annotations of the events classified by the user and the importance of the respective annotations calculated during the classification process (Figure 4.16). The values included in the CSV file are taken directly from the «Importance» table.

```
Annotation,Importance
dish is a quick and easy dish that's ready in minutes!.,4.0
Saská,3.8333333333333
a picture hanging on the wall,2.0
shopping mall,3.25
a black speaker next to a wall,4.5
property image # - at the top of the world with mountain views,3.5
the tv is on the wall,2.0
the long row of windows on the building,3.0
Brzegi,4.4375
hot spring,3.8333333333333
a white porcelain toilet bowl,3.0
the chandelier in the dining room.,2.0
```

**Figure 4.16:** Example of an exported file, with event annotations classified and the degree of importance of each annotation.

Still on Important Events page (Figure 4.15), it is possible click on an event to be redirected to a page with more detailed information about the selected event, as shown in Figure 4.17.

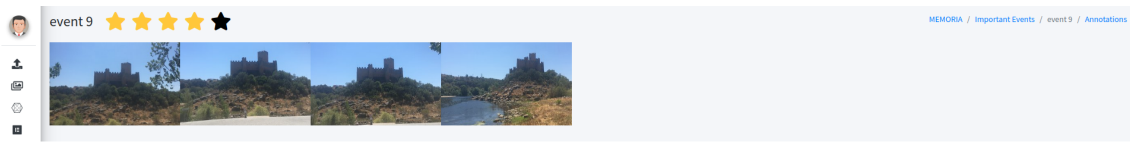



**Figure 4.17:** Important event page.

The redirected page, called the important event page, shows all the images belonging to the event clicked on, the rating given to the event by the user, and a navbar. On this page, the user can also edit or delete the event's rating, if desired. The navbar enables users to return to the important events page and navigate to the important event annotations page (Figure 4.17). The important event annotations page displays the annotations for the event in question (Figure 5.7), along with other information we'll cover in more detail in Section 5.2.3.

### 4.2.5 Docker Setup

Finally, a multi-container docker was created for MEMORIA Light, consisting of four containers: one for the PostgreSQL database, one for the Redis server service, a third for running the main application in Django, and finally a container designed to run the Celery worker. No services have been added or removed from MEMORIA Light compared to the old version of MEMORIA, only the changes mentioned throughout this chapter have been made.

CHAPTER 5

# Results

During this work, a contribution was made to the development of the paper «MEMORIA: A Memory Enhancement and MOment RetrIeval Application at the LSC2024» [41], which can be consulted in Appendix A. This paper was created for the purpose of taking part in the LSC 2024 competition [6]. The annual LSC workshop, previously explained in Section 2.3, is an event where challenges in the development of lifelogging systems are discussed.

One of the pieces of work described in this paper is the event classification functionality, created with the aim of automatically predicting and recovering possible relevant memories for the user, based on previous classifications. This work was carried out in the context of this dissertation on the MEMORIA system.

The paper describes the main changes made to MEMORIA compared to the version that took part in the last edition of the LSC [1]. The new version features a revamped user interface, designed to provide more accessible navigation, improving the user experience.

Once the application was prepared for the data collection procedure, the subsequent steps involved collecting and visualizing the data.

## 5.1 Data Acquisition

Data collection, which involved giving the MEMORIA system to different people to use, was fundamental both for the system itself and for this dissertation. For the system, this step was crucial, as it was the first time that MEMORIA was tested by external people. It was also important for the dissertation, since the data collected is essential for the development of the algorithm for recommending relevant moments.

An experiment protocol was written to guide the participants in the experiment (data collection), which can be seen in Appendix B. The protocol includes a brief introduction explaining what the experiment is for, its objectives and a summary of the activities to be carried out. In addition, it details the requirements that participants must meet, the equipment and material needed, the steps to be followed during the experiment, and the ethical considerations that are being taken into account.

In the experiment, the inclusion and exclusion criteria for individuals were defined to include people with knowledge of technology, specifically the basic sciences of Computer Engineering (Software Engineering and Information Systems). Participants needed to own a computer, have access to the internet and have some personal images of their own (as mentioned in Appendix B). Considering these criteria, the experiment only included participants from one of the three groups of possible recipients of the tool, the group of regular people, technology lovers and photographic recordings. The system is still too underdeveloped to be tested on individuals from the other two groups of possible recipients: people with mental health problems and caregivers/doctors/therapists (the possible groups of recipients of the tool have been clarified in Section 1.1).

Six people carried out and successfully completed the experiment. Each participant received the experiment protocol and used it to carry out the proposed activity. Each person installed the MEMORIA system on their computer and uploaded their personal images. The images were processed, annotated and grouped into events by the system. The participants rated the events they wanted, according to their personal importance, using a scale of one to five.

In the end, each participant returned a CSV file (Figure 4.16). Generating the CSV file is the last step asked of the participants, which is generated by clicking the «Export» button on the Important Events page (Figure 4.15), as described in Section 4.2.4. Each participant's CSV file contains the annotations of the events classified by the user and the importance assigned to these annotations during the classification process, on a scale of one to five. The file has two columns: «Annotation» and «Importance».

Table 5.1 shows some statistics of the results obtained in each participant's experiment. Each column represents a participant, allowing us to see the average importance relative to all the importance ratings obtained for a given participant. The table also shows the total number of annotations and the number of annotations within different importance ranges. These statistics provide a comprehensive view of the participants' behavior and evaluations throughout the study [8].

Looking at table 5.1, the average importance scores range from 3.21 to 4.01, while the total number of annotations ranges from 6924 to 1267. The range of annotations and importance ratings reflects the personal aspect of memory and illustrates the difference in what individuals think meaningful. The analysis and development of algorithms to enhance memory retrieval, including those that suggest pertinent moments, will be based on this data in the future [8].

In addition, it should be noted that there is little data with a low rating, which can be explained by the fact that the images used were taken intentionally by the participants, i.e. if they took the photo on purpose, it was because they were already finding that moment important.

| Participant | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| **Mean Importance** | 4.01 | 3.53 | 3.62 | 3.21 | 3.24 | 3.71 |
| **Total annotations** | 5098 | 2127 | 3704 | 6924 | 2287 | 1267 |
| **Annotations count in the importance range [1,2]** | 449 (8.81%) | 338 (15.89%) | 345 (9.31%) | 1925 (27.80%) | 585 (25.58%) | 101 (7.97%) |
| **Annotations count in the importance range ]2,3]** | 729 (14.30%) | 615 (28.91%) | 860 (23.22%) | 1568 (22.65%) | 577 (22.23%) | 264 (20.84%) |
| **Annotations count in the importance range ]3,4]** | 1488 (29.19%) | 679 (31.92%) | 1967 (53.10%) | 1960 (28.31%) | 730 (31.92%) | 655 (51.70%) |
| **Annotations count in the importance range ]4,5]** | 2432 (47.70%) | 495 (23.27%) | 532 (14.36%) | 1470 (21.23%) | 395 (17.27%) | 247 (19.49%) |

**Table 5.1:** Importance and annotation statistics for participants.

## 5.2 Viewing Important Annotations

Some functionalities for visualizing the results of the user's event classification tasks have been developed. In order to show the user what really happens after their event classification.

In addition to improving interaction and the user's understanding of the system, these features were also designed to serve as a basis for the future development of an algorithm for recommending important events (moments) for the user. Visualizing data before implementing an algorithm is an important step in improving the quality of the algorithm and the results it produces. This process makes it possible to understand the distribution of data, identify patterns, characteristics and possible relationships between different types of data [42].

### 5.2.1 Word Cloud

The first feature implemented to display the user's important annotations was a Word Cloud. This display shows the user's 2000 most important annotations (Figure 5.1).

The Word Cloud was built using the «importance» and «concept» data from the database's «Importance» table, which can be seen in Figure 4.12. During the user's classification of the relevance of events, this table is updated with the importance of each annotation. The explanation of how the relevance of each concept is stored and calculated can be found again in Section 4.2.4. The data was sorted in reverse order of importance, so that the most relevant data came first. Next, the 2000 most important annotations, with the most relevant words larger, were uploaded to Word Cloud. The «concept» of each annotation is the text displayed on the interface, and the size of each word corresponds to the «importance» of each annotation.

The annotations provide information about objects, activities, places, themes, people, animals, actions, events, scenarios, among other aspects. Examples of annotations can be viewed in the Word Cloud illustrated in Figure 5.1 [8].


**Figure 5.1:** Important Annotations WordCloud page.

The Word Cloud was built using the D3.js [43] JavaScript library, which is used to create different types of data visualizations in web browsers.

Building the Word Cloud can be a time-consuming process when there are many annotations. For this reason, only the 2000 most relevant annotations are displayed. In addition, a Word Cloud with too many words could become visually confusing and difficult to interpret. In other words, it is better to display the most important annotations and give a clear and useful view than all the annotations, especially since the most relevant annotations are probably the information the user wants to see the most.

Even with just the 2000 most important annotations, the Word Cloud takes around 15 seconds to appear on the screen. In view of this, while the Word Cloud does not appear on the screen, a loading indicator is displayed to inform the user that the Word Cloud is being built, to wait for it to load, as the process can be lengthy (Figure 5.2). The loading indicator can make all the difference to the user's experience with this feature, providing information about what is happening and why the page won't load.

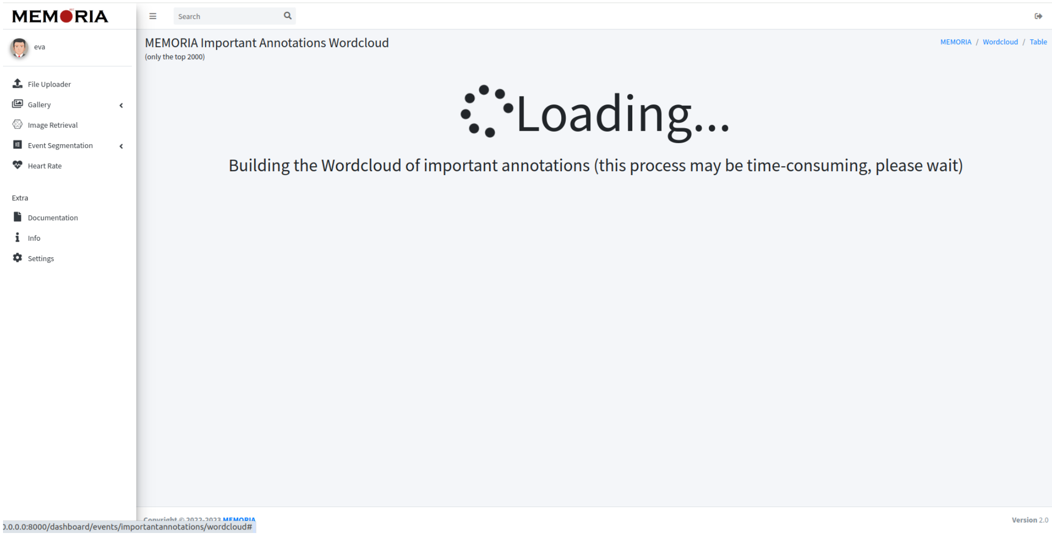



**Figure 5.2:** Loading indicator in WordCloud page.

### 5.2.2 Table

Another page was developed to visualize the results of event classification, the Important Annotations Table page (Figure 5.3). This page displays a table showing the user annotations that have an associated importance, together with the respective importance to the user, determined previously by the system (during the act of classifying events). The table also shows the total frequency of each annotation, i.e. the number of user images that contain the annotation, and the classified events that include the annotation. The data is ordered first by importance and, in case of a tie, by the alphabetical order of the annotations. The event information displayed is the name, the classification of the event given by the user, and the frequency of the annotation in the event, the number of images of the event that contain the annotation.

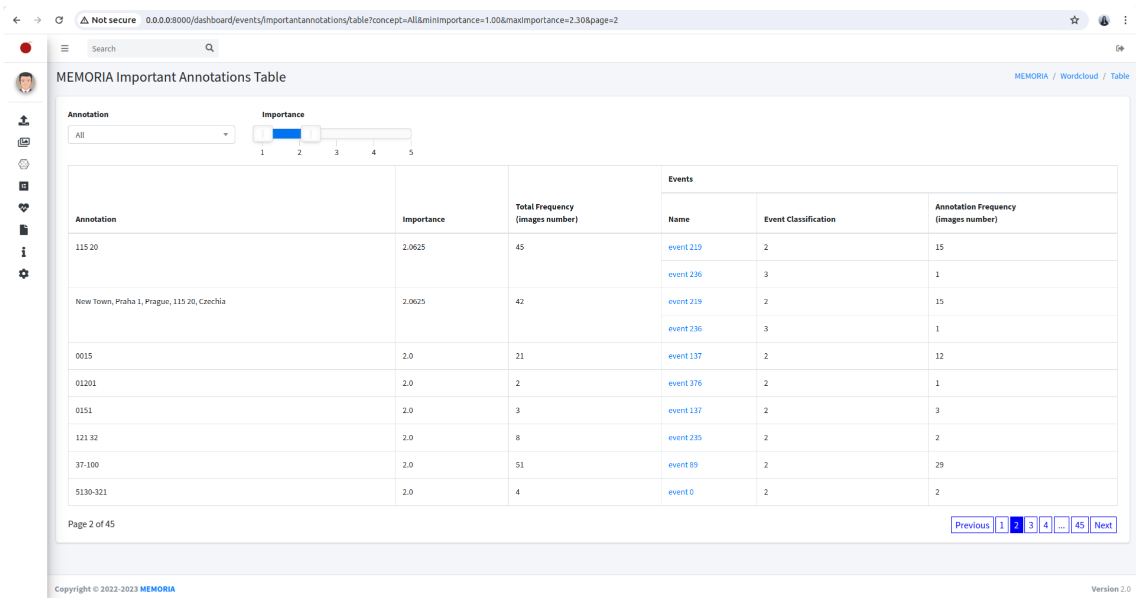



**Figure 5.3:** Important Annotations Table page.

Since important annotations can be a large data set, depending on the number of images the user has in the system and the number of classified events, the pagination technique was implemented in the table presented. This technique ensures that data is displayed instantly, even with large volumes, improving the usability and performance of the system [44].

The data can be explored easily through page navigation, thus displaying one page of results at a time. Page navigation is carried out via buttons containing the number of the page to which they navigate, and buttons for «Previous» and «Next» pages are also included. The pagination controls are displayed at the bottom of the table, along with an indication of the current page (Figure 5.3) [8].

A potentially enormous set of annotations is divided into smaller, easier-to-read pages using the adopted approach. There are eight fixed annotations displayed on each page, which speeds up loading times, boosts efficiency, and enhances user experience overall [8].

Also on the same page, two filters were used to filter the table results. The «Annotation» dropdown allows filtering the results by a given annotation (Figure 5.4), and the «Importance» slider enables adjustment of the desired minimum and maximum importance of the annotations, ranging from values of one to five (including decimals). The filters can be applied together or individually (Figure 5.5).

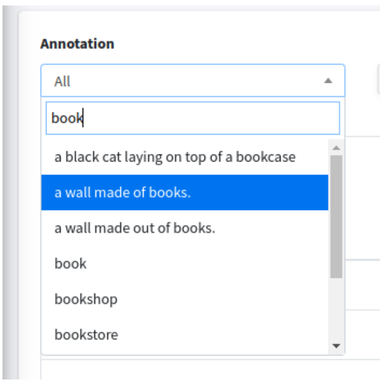



**Figure 5.4:** Annotation filter dropdown on Important Annotations Table page.

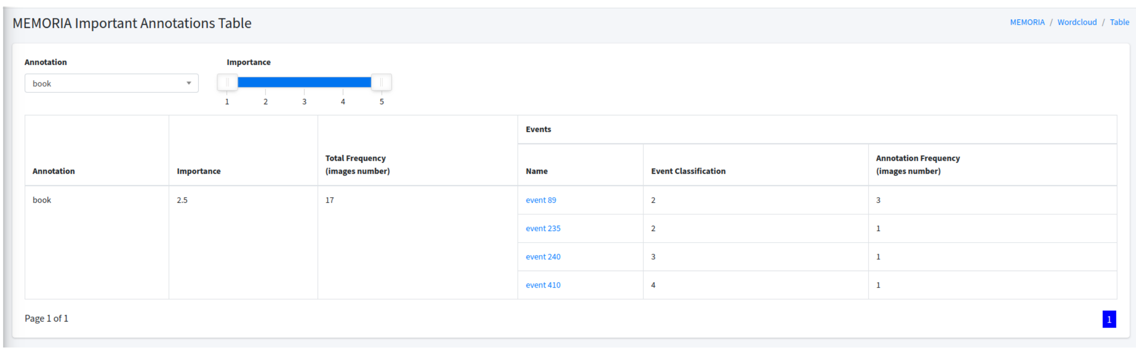



**Figure 5.5:** Table filtered by annotation on Important Annotations Table page.

When the user applies a filter or changes pages, only the table data and pagination controls are dynamically updated. In addition, the filter settings and page number are reflected in the URL, allowing the user to save the URL in case they want to return to a particular filter.

The fact that the entire page is not reloaded and the filters are applied immediately, once again improves performance and user interaction.

The last visualization feature developed on this page was the visualization of the event itself in a pop-up window, triggered by clicking on the name of a given event in the table. In Figure 5.3 the names of the events are colored blue, which differs from the other data in the table, which is colored black. This blue color is intended to make the user understand that the name of the event is clickable and that it will direct to something, in this case to the pop-up window (Figure 5.6).

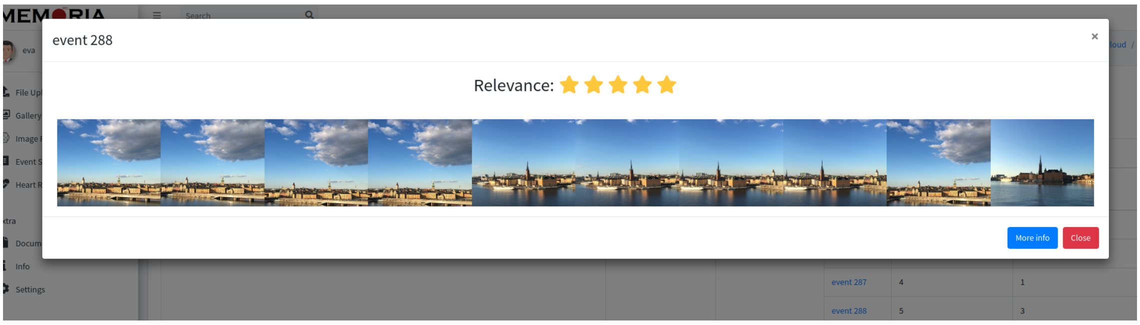
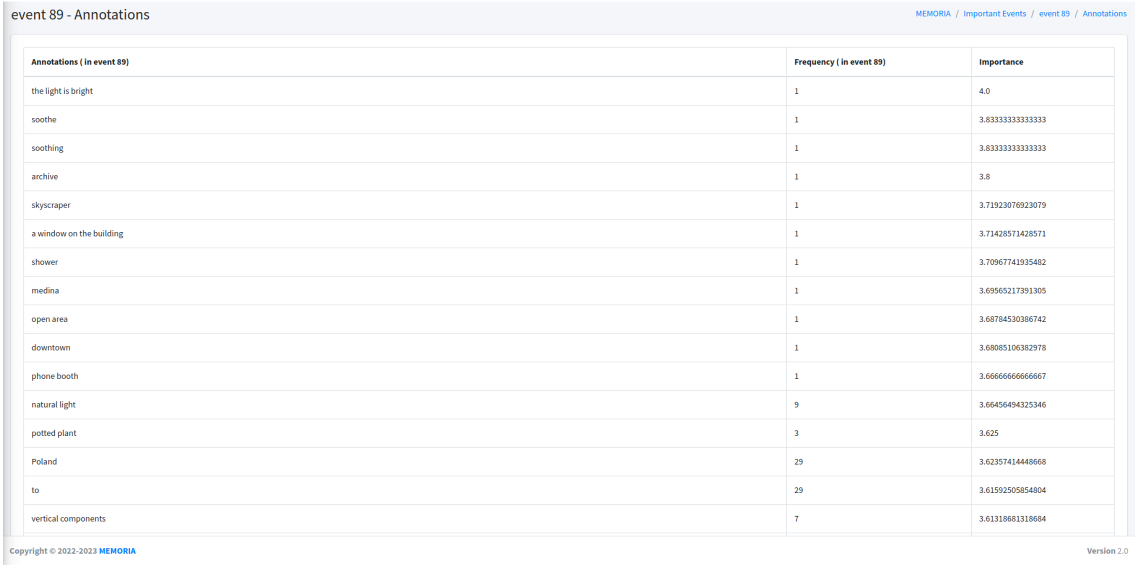



**Figure 5.6:** Visualization of a given event in a pop-up window on the Important Annotations Table page.

The pop-up shows the importance that the user has attached to the selected event, images of the event and has the name of the event as its title. It also shows a «More info» button, which redirects to the important event page. This page, previously mentioned at the end of Section 4.2.4, provides more detailed information about the event in question (Figure 4.17).

### 5.2.3 Page of Annotations from an Important Event

Within the important event page, discussed in Section 4.2.4 and illustrated in Figure 4.17, the page where information about a given important event is shown, the «Annotations» tab has been placed in the top right-hand corner of the page. The tab redirects to a page that shows the annotations of the event in question, the frequency of each annotation in the event, i.e. the number of images of the event that have that annotation and the importance of the annotation calculated for the user (Figure 5.7).


event 89 - Annotations

MEMORIA / Important Events / event 89 / Annotations

| Annotations ( in event 89) | Frequency ( in event 89) | Importance |
|---|---|---|
| the light is bright | 1 | 4.0 |
| soothe | 1 | 3.83333333333333 |
| soothing | 1 | 3.83333333333333 |
| archive | 1 | 3.8 |
| skyscraper | 1 | 3.71923076923079 |
| a window on the building | 1 | 3.71428571428571 |
| shower | 1 | 3.70967741935482 |
| medina | 1 | 3.69565217391305 |
| open area | 1 | 3.68784530386742 |
| downtown | 1 | 3.68085106382978 |
| phone booth | 1 | 3.66666666666667 |
| natural light | 9 | 3.66456494325346 |
| potted plant | 3 | 3.625 |
| Poland | 29 | 3.62357414448668 |
| to | 29 | 3.61592505854804 |
| vertical components | 7 | 3.61318681318684 |

Copyright © 2022-2023 MEMORIA

Version 2.0

**Figure 5.7:** Important Event Annotations page.

The system thus provides different and complementary views. One view focuses on a specific event, showing information about the event, such as all the event annotations, their frequency and importance. Another view focused on the annotation/importance, allowing the associated events and other information for each one to be seen, and even enabling the filtering out of a specific annotation/importance.

CHAPTER 6

# Conclusion

This dissertation focused on the development and adaptation of the MEMORIA application to support an algorithm for recommending relevant moments for the user. Although the algorithm was not developed due to limited time, the work carried out is substantial and important, providing a solid basis for future research and development in this field [8].

The dissertation was laborious and productive, as there was no previous work on MEMORIA related to this topic. In the context of lifelogging, no studies have been found that address the automatic identification and retrieval of possible moments that are really important to the user based on their preferences, without the need for active searches. This dissertation therefore proposes a new feature that could be interesting in a lifelogging system, retrieval of relevant memories that could also improve memory and the ability to remember.

The dissertation was enriching, it transformed an extremely heavy system, which couldn't be run on any computer, into a much lighter system. Capable of running on almost any machine, communicating with an external service to carry out more complex and heavy tasks. In addition, for the first time, MEMORIA was used by an outside audience, allowing the collection and visualization of essential data for training a future model, an algorithm for recommending relevant moments.

The division of MEMORIA into two services, MEMORIA Light and MEMORIA Server, was extremely necessary for data collection. Originally, MEMORIA had many heavy models, required high GPU capacity and other demanding requirements, making it unfeasible to run on ordinary computers. To solve this limitation, all the heavy image processing models were removed from MEMORIA Light and transferred to MEMORIA Server. In MEMORIA Server, an API was developed to enable communication with MEMORIA Light. This solution allowed MEMORIA Light, a lightweight system with minimal requirements, to be used by experiment participants on their own computers, enabling and facilitating data collection.

Various adaptations have been made to make it possible to split up the system, allow MEMORIA Light to be used by external users, and make it possible to collect the desired data. Old features were adapted and improved in order to offer the best possible user experience, making the system user-friendly and informative. New features have also been implemented (which did not exist in the other version), such as event annotation (associating annotations with events, summarizing the annotations of the event's images) and event classification by the user.

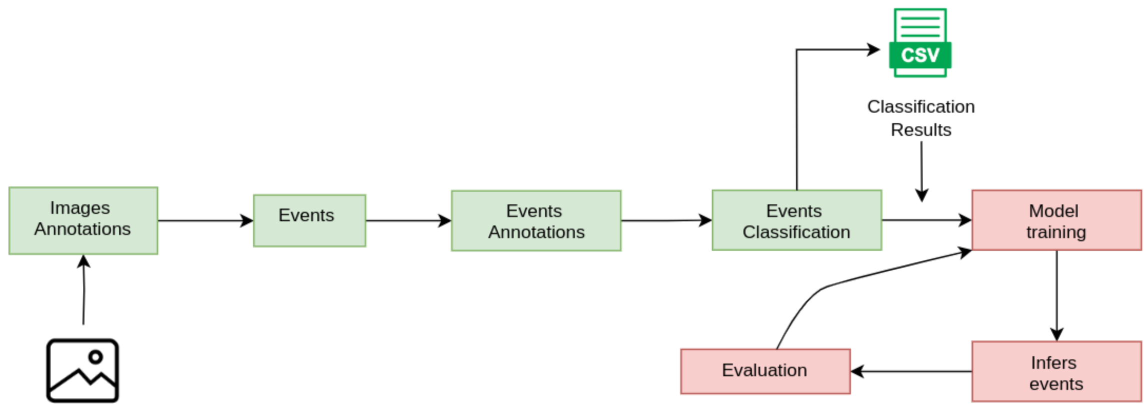



**Figure 6.1:** Development steps for the moment recommendation algorithm. Steps in green are already completed, steps in red are future work.

An experiment was carried out and MEMORIA was used by 6 participants. Each participant was asked to install MEMORIA Light on their computer, upload personal images, ensuring that the images were not saved on any device other than the participant's own. MEMORIA Server processed the images, generating annotations, then MEMORIA Light segmented the images into events, grouping related images and annotating the events. The participant then ranked the events they wanted, from 1 to 5, based on how important the event was to them, and finally downloaded a CSV file (Figure 6.1).

An important element of the data collection process involved participant feedback and real-world applicability. Gave vital information about system performance and user interaction [8].

The data collection resulted in a CSV file for each participant, containing the columns «Annotation» and «Importance». These files record the annotations and their respective importance to the person, calculated during the participant's event classification task. The CSV files are essential for the future development of the recommendation algorithm. In each file, «Diagnostic Clues» can be detected that indicate what might be important to the person in question. From the «Diagnostic Clues», it will be possible to analyze and recommend relevant moments.

At the end of this dissertation, work is delivered that is ready to begin the development of the important moments recommendation algorithm. The system is prepared to collect more data, if necessary, and can be used by external people. Collected data, different sets of annotations and importances, each referring to a specific participant, are provided. Different data visualization functionalities are also provided, to better identify «Diagnostic Clues».

The data collected is now ready to be used to train an event recommendation model. One proposal was for the model, after training, to infer relevant events, and then the results could be submitted to an evaluation for further training. A continuous cycle of improvement is thus proposed: training the model, inferring events and evaluating it, with the aim of progressively improving the model (Figure 6.1).

The development and integration of an algorithm for recommending moments is therefore identified as future work. This can identify «Diagnostic Clues» from the data collected in this work, in order to suggest relevant moments to users. All of the data and infrastructures that have already been prepared for this dissertation will be useful for this advancement. It is suggested to test and validate the algorithm on actual users in order to continuously refine and enhance it. While increasing the amount of data gathered might be beneficial, it is not considered to be a top priority right now [8].

The objectives of the study were achieved by acquiring and analyzing data, using the system with an outside audience and developing user-friendly functionalities that seek to offer the user the best possible experience [8].

The variety of annotations, and the different importance given to annotations by people in data collection, shows how personal and unique memories are, and highlights the difference in what is considered relevant to each person [8]. The development of a personalized algorithm that recommends significant moments for the user, adapting to their preferences, could be a feature that has a positive impact, both in lifelogging and in the context of capturing images on purpose.

In short, although the development of the algorithm has not been completed, this dissertation represents a significant step forward in preparing MEMORIA for this purpose. Adapting MEMORIA to be used by external people, through the creation of a lighter version, MEMORIA Light, which communicates with MEMORIA Server, an external service responsible for image processing, was a crucial step. The new functionalities implemented to carry out data collection and analysis are also essential. The contributions made enable future research to begin immediately on the development of the algorithm for recommending relevant moments.

# References


[1] R. Ribeiro, L. Amaral, W. Ye, A. Trifan, A. J. R. Neves, and P. Iglésias, «Memoria: A memory enhancement and moment retrieval application for lsc 2023», in *Proceedings of the 6th Annual ACM Lifelog Search Challenge*, ser. LSC '23, Thessaloniki, Greece: Association for Computing Machinery, 2023, pp. 18–23, ISBN: 9798400701887. DOI: 10.1145/3592573.3593099. [Online]. Available: https://doi.org/10.1145/3592573.3593099.

[2] R. Ribeiro, A. Trifan, and A. J. R. Neves, «Lifelog retrieval from daily digital data: Narrative review», *JMIR Mhealth Uhealth*, vol. 10, no. 5, e30517, May 2022, ISSN: 2291-5222. DOI: 10.2196/30517. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/35499858.

[3] R. Ribeiro, A. Trifan, and A. J. R. Neves, «Hierarchical event segmentation for lifelogging memory retrieval and organization», (Under Review), 2023.

[4] *Dementia statistics*, Alzheimer's Disease International, Accessed: 2024-01-14. [Online]. Available: https://www.alzint.org/about/dementia-facts-figures/dementia-statistics/.

[5] O. Jeunehomme1 and A. D'Argembeau, «Event segmentation and the temporal compression of experience in episodic memory», *Psychological Research*, vol. 84, pp. 481–490, 2020. DOI: 10.1007/s00426-018-1047-y. [Online]. Available: https://doi.org/10.1007/s00426-018-1047-y.

[6] C. Gurrin, L. Zhou, G. Healy, *et al.*, «Introduction to the seventh annual lifelog search challenge, lsc'24», in *Proceedings of the 2024 International Conference on Multimedia Retrieval*, ser. ICMR '24, Phuket, Thailand: Association for Computing Machinery, 2024, pp. 1334–1335, ISBN: 9798400706196. DOI: 10.1145/3652583.3658891. [Online]. Available: https://doi.org/10.1145/3652583.3658891.

[7] S. Hodges, E. Berry, K. Wood, K. Woodberry, and S. Hodges, «Sensecam: A wearable camera that stimulates and rehabilitates autobiographical memory», *Memory*, vol. 19, no. 7, pp. 685–696, Jan. 2011. [Online]. Available: https://www.microsoft.com/en-us/research/publication/sensecam-a-wearable-camera-that-stimulates-and-rehabilitates-autobiographical-memory/.

[8] OpenAI, *Chatgpt-3.5*, Large language model, 2023. [Online]. Available: https://chat.openai.com.

[9] A.-Z. Yen, M.-H. Fu, W.-H. Ang, *et al.*, «Visual lifelog retrieval: Humans and machines interpretation on first-person images», *Multimedia Tools and Applications*, vol. 82, pp. 37 757–37 787, 2023. DOI: 10.1007/s11042-023-14344-x. [Online]. Available: https://doi.org/10.1007/s11042-023-14344-x.

[10] T. van Teijlingen, E. Oudman, and A. Postma, «Lifelogging as a rehabilitation tool in patients with amnesia: A narrative literature review on the effect of lifelogging on memory loss», *Neuropsychological Rehabilitation*, vol. 32, no. 10, pp. 2646–2672, 2022, PMID: 34533426. DOI: 10.1080/09602011.2021.1974891. eprint: https://doi.org/10.1080/09602011.2021.1974891. [Online]. Available: https://doi.org/10.1080/09602011.2021.1974891.

[11] E. Woodberry, G. Browne, S. Hodges, P. Watson, and N. K. K. Woodberry, «The use of a wearable camera improves autobiographical memory in patients with alzheimer's disease», *Memory*, vol. 23, no. 3, pp. 340–349, 2015, PMID: 24528204. DOI: 10.1080/09658211.2014.886703. eprint: https://doi.org/10.1080/09658211.2014.886703. [Online]. Available: https://doi.org/10.1080/09658211.2014.886703.

[12] K. Schoeffmann, «Lifexplore at the lifelog search challenge 2023», in *Proceedings of the 6th Annual ACM Lifelog Search Challenge*, ser. LSC '23, Thessaloniki, Greece: Association for Computing Machinery, 2023, pp. 53–58, ISBN: 9798400701887. DOI: 10.1145/3592573.3593105. [Online]. Available: https://doi.org/10.1145/3592573.3593105.

[13] C. Gurrin, B. Þ. Jónsson, D. T. D. Nguyen, *et al.*, «Introduction to the sixth annual lifelog search challenge, lsc'23», in *Proceedings of the 2023 ACM International Conference on Multimedia Retrieval*, ser. ICMR '23, Thessaloniki, Greece: Association for Computing Machinery, 2023, pp. 678–679, ISBN: 9798400701788. DOI: 10.1145/3591106.3592304. [Online]. Available: https://doi.org/10.1145/3591106.3592304.

[14] C. Gurrin, L. Zhou, G. Healy, *et al.*, «Introduction to the fifth annual lifelog search challenge, lsc'22», in *Proceedings of the 2022 International Conference on Multimedia Retrieval*, ser. ICMR '22, Newark, NJ, USA: Association for Computing Machinery, 2022, pp. 685–687, ISBN: 9781450392389. DOI: 10.1145/3512527.3531439. [Online]. Available: https://doi.org/10.1145/3512527.3531439.

[15] L. D. Tran, B. Nguyen, L. Zhou, and C. Gurrin, «Myeachtra: Event-based interactive lifelog retrieval system for lsc'23», in *Proceedings of the 6th Annual ACM Lifelog Search Challenge*, ser. LSC '23, Thessaloniki, Greece: Association for Computing Machinery, 2023, pp. 24–29, ISBN: 9798400701887. DOI: 10.1145/3592573.3593100. [Online]. Available: https://doi.org/10.1145/3592573.3593100.

[16] N. Alam, Y. Graham, and C. Gurrin, «Memento 3.0: An enhanced lifelog search engine for lsc'23», in *Proceedings of the 6th Annual ACM Lifelog Search Challenge*, ser. LSC '23, Thessaloniki, Greece: Association for Computing Machinery, 2023, pp. 41–46, ISBN: 9798400701887. DOI: 10.1145/3592573.3593103. [Online]. Available: https://doi.org/10.1145/3592573.3593103.

[17] N. Hoang-Xuan, T.-L. Nguyen-Ho, C. Gurrin, and M.-T. Tran, «Lifelog discovery assistant: Suggesting prompts and indexing event sequences for first at lsc 2023», in *Proceedings of the 6th Annual ACM Lifelog Search Challenge*, ser. LSC '23, Thessaloniki, Greece: Association for Computing Machinery, 2023, pp. 47–52, ISBN: 9798400701887. DOI: 10.1145/3592573.3593104. [Online]. Available: https://doi.org/10.1145/3592573.3593104.

[18] A. Alateeq, M. Roantree, and C. Gurrin, «Voxento 4.0: A more flexible visualisation and control for lifelogs», in *Proceedings of the 6th Annual ACM Lifelog Search Challenge*, ser. LSC '23, Thessaloniki, Greece: Association for Computing Machinery, 2023, pp. 7–12, ISBN: 9798400701887. DOI: 10.1145/3592573.3593097. [Online]. Available: https://doi.org/10.1145/3592573.3593097.

[19] T.-N. Nguyen, T.-K. Le, V.-T. Ninh, *et al.*, «E-lifeseeker: An interactive lifelog search engine for lsc'23», in *Proceedings of the 6th Annual ACM Lifelog Search Challenge*, ser. LSC '23, Thessaloniki, Greece: Association for Computing Machinery, 2023, pp. 13–17, ISBN: 9798400701887. DOI: 10.1145/3592573.3593098. [Online]. Available: https://doi.org/10.1145/3592573.3593098.

[20] L. Rossetto, O. Inel, S. Lange, F. Ruosch, R. Wang, and A. Bernstein, «Multi-mode clustering for graph-based lifelog retrieval», in *Proceedings of the 6th Annual ACM Lifelog Search Challenge*, ser. LSC '23, Thessaloniki, Greece: Association for Computing Machinery, 2023, pp. 36–40, ISBN: 9798400701887. DOI: 10.1145/3592573.3593102. [Online]. Available: https://doi.org/10.1145/3592573.3593102.

[21] F. Spiess, R. Gasser, H. Schuldt, and L. Rossetto, «The best of both worlds: Lifelog retrieval with a desktop-virtual reality hybrid system», in *Proceedings of the 6th Annual ACM Lifelog Search Challenge*, ser. LSC '23, Thessaloniki, Greece: Association for Computing Machinery, 2023, pp. 65–68, ISBN: 9798400701887. DOI: 10.1145/3592573.3593107. [Online]. Available: https://doi.org/10.1145/3592573.3593107.

[22] Q.-L. Tran, L.-D. Tran, B. Nguyen, and C. Gurrin, «Memoriease: An interactive lifelog retrieval system for lsc'23», in *Proceedings of the 6th Annual ACM Lifelog Search Challenge*, ser. LSC '23, Thessaloniki, Greece: Association for Computing Machinery, 2023, pp. 30–35, ISBN: 9798400701887. DOI: 10.1145/3592573.3593101. [Online]. Available: https://doi.org/10.1145/3592573.3593101.

[23] T.-T. Nguyen-Dang, X.-D. Thai, G.-H. Vuong, *et al.*, «Lifeinsight: An interactive lifelog retrieval system with comprehensive spatial insights and query assistance», in *Proceedings of the 6th Annual ACM Lifelog Search Challenge*, ser. LSC '23, Thessaloniki, Greece: Association for Computing Machinery, 2023, pp. 59–64, ISBN: 9798400701887. DOI: 10.1145/3592573.3593106. [Online]. Available: https://doi.org/10.1145/3592573.3593106.

[24] M. T. Hordvik, J. S. T. Østby, M. Kesavulu, T.-N. Nguyen, T.-K. Le, and D.-T. Dang-Nguyen, «Lifelens: Transforming lifelog search with innovativenbsp;ux/uinbsp;design», in *Proceedings of the 6th Annual ACM Lifelog Search Challenge*, ser. LSC '23, Thessaloniki, Greece: Association for Computing Machinery, 2023, pp. 1–6, ISBN: 9798400701887. DOI: 10.1145/3592573.3593096. [Online]. Available: https://doi.org/10.1145/3592573.3593096.

[25] A. R. Doherty and A. F. Smeaton, «Automatically segmenting lifelog data into events», in *2008 Ninth International Workshop on Image Analysis for Multimedia Interactive Services*, 2008, pp. 20–23. DOI: 10.1109/WIAMIS.2008.32.

[26] J. Son and S. B. Kim, «Content-based filtering for recommendation systems using multiattribute networks», *Expert Systems with Applications*, vol. 89, pp. 404–412, 2017, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2017.08.008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417417305468.

[27] A. Kumar Sharma, B. Bajpai, R. Adhvaryu, S. Dhruvi Pankajkumar, P. Parthkumar Gordhanbhai, and A. Kumar, «An efficient approach of product recommendation system using nlp technique», *Materials Today: Proceedings*, vol. 80, pp. 3730–3743, 2023, SI:5 NANO 2021, ISSN: 2214-7853. DOI: https://doi.org/10.1016/j.matpr.2021.07.371. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214785321052305.

[28] I. Hossain, M. A. H. Palash, A. T. Sejuty, *et al.*, *A survey of recommender system techniques and the ecommerce domain*, 2023. arXiv: 2208.07399 [cs.IR].

[29] R. Ribiero, A. Trifan, and A. J. R. Neves, «Memoria: A memory enhancement and moment retrieval application for lsc 2022», in *Proceedings of the 5th Annual on Lifelog Search Challenge*, ser. LSC '22, Newark, NJ, USA: Association for Computing Machinery, 2022, pp. 8–13, ISBN: 9781450392396. DOI: 10.1145/3512729.3533011. [Online]. Available: https://doi.org/10.1145/3512729.3533011.

[30] *Fastapi documentation*, FastAPI, Accessed: 2024-04-12. [Online]. Available: https://fastapi.tiangolo.com/.

[31] *Flask documentation*, Flask, Accessed: 2024-04-12. [Online]. Available: https://flask.palletsprojects.com/en/3.0.x/.

[32] *Django documentation*, Django, Accessed: 2024-04-12. [Online]. Available: https://docs.djangoproject.com/en/5.0/ref/class-based-views/.

[33] *Django, flask, fastapi: A comparative analysis of python web frameworks*, Capital Numbers, Accessed: 2024-04-12. [Online]. Available: https://www.capitalnumbers.com/blog/django-vs-flask-vs-fastapi/.

[34] *Redis documentation*, Redis, Accessed: 2024-04-12. [Online]. Available: https://redis.io/docs/latest/.

[35] *Celery documentation*, Celery, Accessed: 2024-04-13. [Online]. Available: https://docs.celeryq.dev/en/stable/getting-started/introduction.html.

[36] *Nginx documentation*, NGINX, Accessed: 2024-04-13. [Online]. Available: https://nginx.org/en/.

[37] *Gunicorn documentation*, Gunicorn, Accessed: 2024-04-13. [Online]. Available: https://gunicorn.org/.

[38] *How to securely deploy a fastapi app with nginx and gunicorn*, Dylan Castillo, Accessed: 2024-04-13. [Online]. Available: https://dylancastillo.co/fastapi-nginx-gunicorn/#configure-supervisor.

[39] *Let's encrypt documentation*, Let's Encrypt, Accessed: 2024-04-13. [Online]. Available: https://letsencrypt.org/getting-started/.

[40] *Certbot documentation*, certbot, Accessed: 2024-04-13. [Online]. Available: https://certbot.eff.org/.

[41] A. Gago, B. Kaluza, E. Bartolomeu, J. Pandeirada, R. Ribeiro, and A. J. R. Neves, «Memoria: A memory enhancement and moment retrieval application at the lsc2024», in *Proceedings of the 7th Annual ACM Workshop on the Lifelog Search Challenge*, ser. LSC '24, Phuket, Thailand: Association for Computing Machinery, 2024, pp. 99–104, ISBN: 9798400705502. DOI: 10.1145/3643489.3661129. [Online]. Available: https://doi.org/10.1145/3643489.3661129.

[42] A. Unwin, «Why Is Data Visualization Important? What Is Important in Data Visualization?», *Harvard Data Science Review*, vol. 2, no. 1, Jan. 2020, https://hdsr.mitpress.mit.edu/pub/zok97i7p.

[43] *D3.js documentation*, D3, Accessed: 2024-05-16. [Online]. Available: https://d3js.org/.

[44] *Pagination*, Katie Terrell Hanna, Ivy Wigmore, TechTarget, Accessed: 2024-05-16. [Online]. Available: https://www.techtarget.com/whatis/definition/pagination.

APPENDIX A

# Paper MEMORIA at the LSC2024

*This chapter presents the paper «MEMORIA: A Memory Enhancement and MOment RetrIeval Application at the LSC2024» [41] submitted to the LSC 2024 conference [6]. The paper describes the main changes made to MEMORIA compared to the version that took part in the last edition. Some contributions made in this dissertation are described in the paper.*

# MEMORIA: A Memory Enhancement and MOment RetrIeval Application at the LSC2024

Alexandre F. P. Gago
IEETA/DETI, Universidade de Aveiro
Portugal
alexandre.gago@ua.pt

Bernardo F. M. Kaluza
IEETA/DETI, Universidade de Aveiro
Portugal
bernardokaluza@ua.pt

Eva P. Bartolomeu
IEETA/DETI, Universidade de Aveiro
Portugal
evabartolomeu@ua.pt

Josefa N. S. Pandeirada
WJCR, Universidade de Aveiro
Portugal
josefa@ua.pt

Ricardo Ribeiro
IEETA/DETI, Universidade de Aveiro
Portugal
rfribeiro@ua.pt

António J. R. Neves
IEETA/DETI, Universidade de Aveiro
Portugal
an@ua.pt

## ABSTRACT

The emergence of lifelogging, recording, storing and organizing personal data, underscores the need for efficient digital systems. Lifelog Search Challenge (LSC) promotes a space for discussing the challenges in developing these systems. MEMORIA, in its third version for LSC'24, stands out mainly for its renewed user interface, which aims for more intuitive and accessible navigation, offering a better user experience. Adding more layers and sub-layers has improved event segmentation based on hierarchical events. Another new feature is searching using an image and redesigning the home page, which offers a general summary of the user's lifelogs. Event navigation has been simplified, making navigating on a single page possible. Enhanced personalized search capabilities are made possible by a new labels page that offers a consolidated view of the system's annotations. Also, the image retrieval page has been updated with filters and informative pop-ups. Using Large Language Models (LLMs) to annotate images and events is a significant development in progress that could provide richer and more contextual descriptions. An event classification feature was created to retrieve pertinent memories automatically. The new album feature allows users to organize and explore image collections, while event search will enable users to find more specific moments using text queries. These new features are expected to make MEMORIA a relevant tool in lifelogging, with innovative solutions to the complex challenge of capturing, organizing, and accessing digital memories.

## CCS CONCEPTS

• **Information systems** → **Search interfaces**; • **Human-centered computing** → *Interactive systems and tools.*

## KEYWORDS

lifelog, lifelogging, image processing, image annotation, data retrieval, object detection, Machine Learning, Information Systems, Large Language Models



*LSC '24, June 10, 2024, Phuket, Thailand*

ACM ISBN 979-8-4007-0550-2/24/06
https://doi.org/10.1145/3643489.3661129

**ACM Reference Format:**
Alexandre F. P. Gago, Bernardo F. M. Kaluza, Eva P. Bartolomeu, Josefa N. S. Pandeirada, Ricardo Ribeiro, and António J. R. Neves. 2024. MEMORIA: A Memory Enhancement and MOment RetrIeval Application at the LSC2024. In *The 7th Annual ACM Lifelog Search Challenge (LSC '24), June 10, 2024, Phuket, Thailand.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3643489.3661129

## TEAM NAME

MEMORIA - A Memory Enhancement and MOment RetrIeval Application

## 1 INTRODUCTION

Thanks to recent technological advancements, there has been a proliferation of sensors and gadgets intended to gather data on human behavior and activities [9]. This includes cameras, smartphones, and wearable devices. Furthermore, the cost and size of these technologies are falling, making it easier for society to adopt them widely [10].

With the great diversity of devices available, the data collected is equally diverse, including images, videos, audio, physiological signals, geographical coordinates, and biometric information, among others [9]. Due to the accessibility and availability of these devices, data is being collected quickly, producing a massive amount of data [10].

This data is indispensable in numerous applications, including information retrieval, comprehension of daily patterns, detection of mental disease, and image/video search, among others. This usefulness has led to the emergence of lifelogging, which is the act of collecting, storing, analyzing, processing, and retrieving multimodal personal data in digital format, which in this context is called lifelog. However, lifelogging tasks are difficult and complex due to the data's diversity, volume, and speed of acquisitions. As a result, lifelogging systems are typically used for these tasks and use various algorithms and methods to manage data [9] [10].

The Lifelog Search Challenge (LSC) is an annual event that invites participants to face challenges related to lifelogging, with the aim of discussing approaches and conclusions and recognizing potential and difficulties [9] [10]. Teams compete in real-time during the challenge by completing the same lifelog retrieval tasks and using the same dataset. The efficiency of the search in finding the right information in the least amount of time is used to measure performance [4].

This paper presents the third version of the Memory Enhancement and MOment RetrIeval Application (MEMORIA) intending to participate in the LSC24 event [4]. The updated MEMORIA interface makes navigation easier for users and enhances the overall experience with a more user-friendly design. Additionally, changes have been made to event segmentation, which now has five layers and two sublayers (BIQA and boundary fine-tuning) between them to provide a more sophisticated way to organize lifelog data. A new image search option has also been introduced, allowing searching for similar images by uploading an image. The Dashboard page has been transformed to provide quick access to features and an overview of user statistics, number of images, and events. The navigation of events has been simplified for a more fluid exploration of moments on a single page. A newly added labels page provides an interactive and consolidated view of the system's annotations. Furthermore, the image retrieval page has been redesigned with filters and informative pop-ups to provide a more straightforward search experience. An important prospective development is the use of Large Language Models (LLMs) to annotate images and events, enabling richer and more contextual descriptions. An event classification functionality was developed to automatically retrieve relevant memories. Users can now manage and view image collections more easily with the help of the new album function, and event search enables more focused moment discovery based on text queries.

This paper is organized into five sections, starting with Section 1 with an introduction and then Section 2 analyzing the latest studies related to lifelogging. Section 3 describes an overview of MEMORIA and the innovations developed in the new version. This is followed by Section 4, which reports work currently in progress. Finally, the last section, Section 5, reveals some conclusions.

## 2 RELATED WORK

As lifelong systems are being developed worldwide, developers can test, analyze, and improve their systems through participation in several lifelogging events. In the LSC, several teams compete against one another through lifelogging tasks to determine their system's efficiency while searching for possible improvements and further advancing lifelogging and retrieval technology.

Memento 3.0 [1] uses image-text embeddings derived from two different CLIP[8] models to create a ranking mechanism that aims to merge the similarity scores between an image and the query; this is achieved through the aggregating of ratings produced by multiple models' output. Additionally, it also reduces The query processing time can be determined by employing a cluster-based search technique.

LifeXplore [11] provides a system based on the extraction of visual concepts from images. To achieve this, this system uses several models. The first, CLIP [8], is responsible for the generation of text embedding and image similarity. The second CRAFT [3] is responsible for text extraction from images. The third, YOLOv7 [13], is responsible for the detection of objects. And lastly, an EfficientNet B2 model is responsible for generating semantic concepts. All these models work hand in hand to generate tags that help with the image retrieval task.

Voxento 4.0[2] is a system that prioritizes voice queries as the primary way of interacting with the system. This approach allows the user to communicate with the system through voice commands and even special voice commands that the system can recognize during retrieval. Regarding image retrieval, the system uses a CLIP model to generate embeddings for the images and then searches for the most similar images using cosine similarity.

MyEachtra [12] is an event-based lifelog retrieval system. It segments events based on location metadata, visual metadata, and temporal changes between images. It has a simple and user-friendly interface. It evaluates the similarity of images with CLIP embeddings and integrates the FrozenBiLM model [15] to improve responses to question-answer queries.

Collectively, these works contribute to the ongoing enhancement of lifelogging retrieval technology, paving the way for further advancements in the field.

## 3 MEMORIA SYSTEM

MEMORIA is a web-based application designed to store, organize, analyze, visualize, and retrieve lifelog data. The platform comprises several modules that allow and enhance a user's lifelog retrieval experience. Each module contributes to managing the lifelog data, including storage, organization, annotation, retrieval, and visualization functions. With a new and improved interface, Figure 1 shows the current MEMORIA system overview.

The platform is in permanent improvement to enhance its current functionalities of visualization and analysis of lifelogs so that users have responsive and intuitive tools to interact with that data. This enhances the user's capability to interpret and understand lifelog data originating from their daily lives. This chapter provides a quick overview of the system's main modules, their functions, and their workings.

### 3.1 Image Annotation

The best way to retrieve lifelog images is to search by typing a text query. To do so, there is an important step to allow such a feature: the automatic extraction of meaningful information from these images. This extracted information can help to close the gap between images and text queries. This a complex task as an image can have many different types of information that the system can harness. MEMORIA uses several models and algorithms to extract information from the lifelog images to achieve this goal.

The current system iteration uses the same models and algorithms to annotate the lifelog images as its previous version. It offers object detection via YOLOv7[13], object understanding via GRiT[14], Optical Character Recognition using Craft[3], Scene Understanding, Automatic Caption Generation, and Semantic Location Annotation.

### 3.2 Event Segmentation

Event segmentation organizes lifelogs and divides them into homogeneous temporal segments, each representing different events. This process is relevant to memory enhancement, as events more accurately capture vivid memories compared to isolated lifelogs

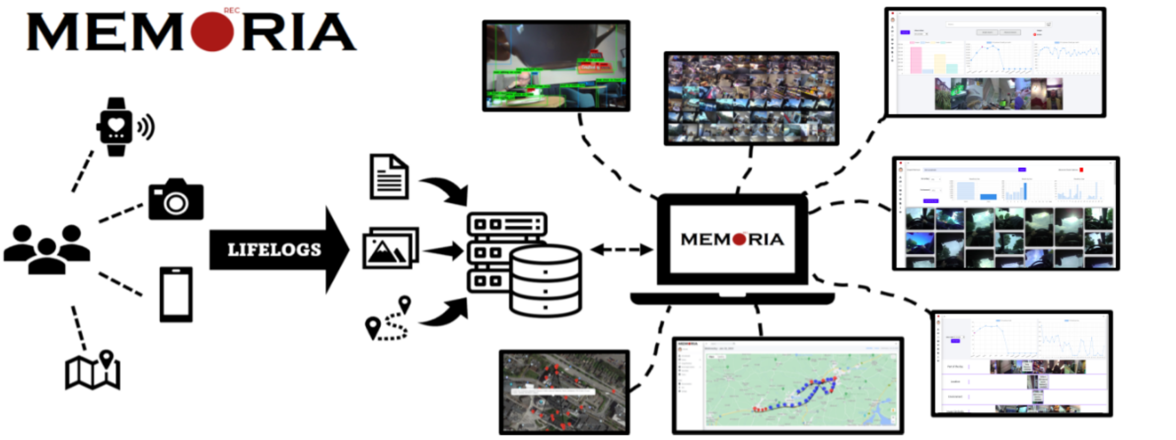



**Figure 1: Overview of the MEMORIA lifelog system architecture. The system comprises interconnected blocks representing the key components described in this paper.**

[5]. It also improves search efficiency and saves space in the visual interface [9].

MEMORIA maintains the hierarchical structure for segmenting events presented in the previous version but with some improvements [9]. Five layers comprise the current hierarchical technique (Figure 2). These layers are based on pre-processed data, which is covered in Section 3.1 and comprises temporal information, location, environment, image similarity, and semantic annotations. To start, the algorithm cluster the images into "Day Events", which are collections of images taken on the same day. Subdividing each "Day Event" into "Parts of Day", such as morning or afternoon, reflects the time of day the images were taken. The images are then divided into "Location Events" clusters according to the precise locations where they happened, and into "Environment" clusters that differentiate between indoor and outdoor areas. Lastly, similar images that might depict the same moment are grouped together in the "Similarity" layer.

Blind Image Quality Assessment (BIQA) is a sub-layer located between the "Day" and "Parts of Day" layers, designed to remove low-quality images or those that don't contain relevant contextual content. A sub-layer sits between the other levels and automatically adjusts the bounds of the events in the preceding hierarchical layer.

## 3.3 Image Retrieval

The MEMORIA system has enhanced capabilities of image retrieval due to the implementation of some advancements in text-query image retrieval, building upon the foundation of MEMORIA's previous iteration, while introducing a simplified interface and additional retrieval features. Moreover, a novel method "Image Upload Search" has been introduced, enabling users to search for images by simply dragging or selecting an image file into a designated drop zone on the dashboard.

*3.3.1 Text-query Image Retrieval.* The image retrieval algorithm is mostly the same as MEMORIA's previous iteration. However, the interface is simplified, and additional retrieval features and search options were added. These changes will be talked about in the interface subsection.

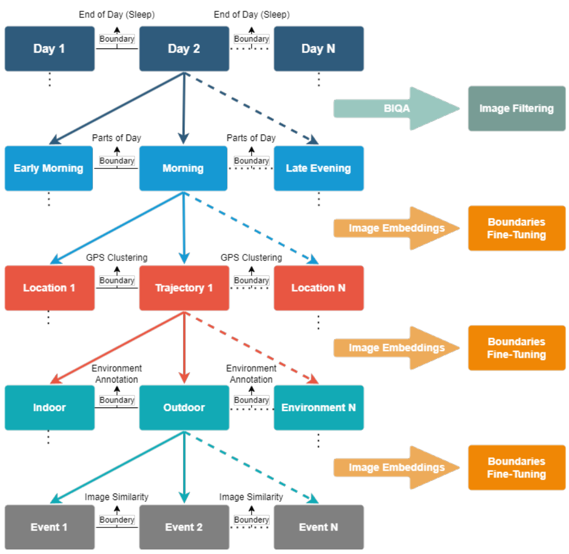



**Figure 2: Hierarchical event-based event segmentation method.**

*3.3.2 Image Upload Search.* A new way to search for images is by dragging (or selecting) an image file into a drop zone at the dashboard. The system will generate annotations for that image using some of the image annotation algorithms and then use those annotations on the retrieval page. This new feature allows searching for images similar to those on the system.

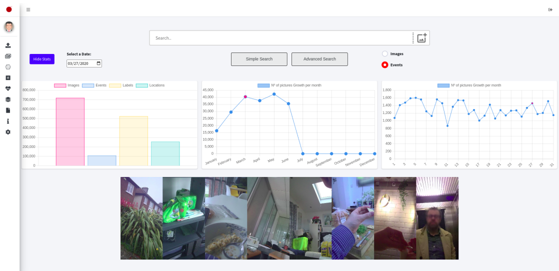


Figure 3: Dashboard Page

## 3.4 Interface

The interface is currently being improved to allow new ways to interact with the system and to provide an easier and more intuitive user experience. The old pages are still accessible from the sidebar, so any feature that is not present in the new pages is still accessible. Some of the new changes will be addressed in the following subsections.

*3.4.1 Dashboard.* The home page (dashboard) has been completely revamped to improve the user experience and make the search for images/events faster 3. The text field on the page offers a faster and easier way to search for some images by redirecting the user to the Image Retrieval Page 6 with the inputted query. When a user clicks on the text field, the previous searches will also be shown so that the user can make the same searches faster. A user can also search for images by uploading an image by clicking on the icon on the right of the text field. Finally, under the text field, there is a radio button to enable searching for images or events. By clicking the "Show Stats" button on the left of the page, three different graphs will be displayed on the screen alongside a date picker, with today being the default date, and a visualization of the second layer of events (Part of day), related to the selected date in the date picker. The first graph allows for a quick rundown of user statistics, such as the number of pictures and events. The second (year graph), allows the user to see the number of monthly images of the selected year. And finally, the third graph (month graph) similar to the previous one, allows the user to see the number of daily pictures of the selected month. By clicking on a dot on the year graph, the date picker's month will be changed which, in turn, will update the month graph and the events view. By clicking on a dot on the month graph, the date picker's day is updated and the events view will also be updated to present the events of the selected date. Upon clicking an event, the user will be directed to the Event Navigation page 4, where they can delve deeper into the events of that specific date.

*3.4.2 Event Navigation.* A new, more straightforward, intuitive way to navigate and search for events was created to enable for a faster and more intuitive way to navigate through the events of a user. Previously, the user needed to go through several pages to navigate through events, and now they can do that in only one in a more accessible way. This can be viewed in the figure 4. The page contains, on the top left, a date picker and a button to hide the two graphs on the top right. The graphs are similar to the dashboard ones. The first contains the number of events per month of the selected year and the second contains the number of events of the days of the selected month. By clicking on a dot on the year graph, the date picker's month will be changed which, in turn, will update the month graph and the events navigation section. This section, sectioned on the bottom half of the page, allows the user to navigate through all the event layers. By clicking on an event of a layer, the next layer will pop up with the next layer-related events. When clicking on the last layer, the images of that event will be shown and if clicked, will redirect the user to the page with all the image's information. This page allows for back-and-forward event navigation. If users are currently viewing the events of a certain layer, they can click on a previous layer and continue searching from there.

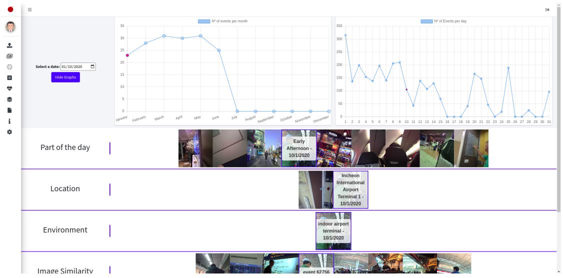



Figure 4: Event Navigation Page

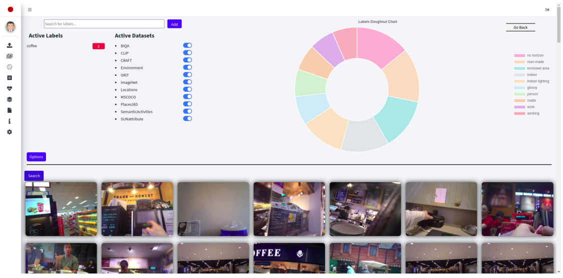


Figure 5: Labels Page

*3.4.3 Labels Page.* A new page displays all the system's labels created. This page offers a pie chart that shows the number of times a label was attributed to an image and its overall percentage. This chart is highly interactive, it can be clicked, its range and offset can be changed, and the presented labels that appear can be filtered by the model that created the chart. Finally, this page offers a different option to search for an image: after selecting some labels, this page can search for images that must have all the selected labels and also order them by the highest confidence value given by its models. This page can be seen in the figure 5.

*3.4.4 New Image Retrieval Page.* The image retrieval page has been revamped to enable a more intuitive and pleasing user experience. The search options have been reduced to the most relevant ones and also simplified. When searching, all the results are divided by

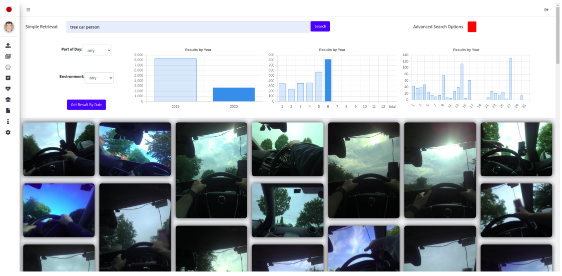


Figure 6: New Image Retrieval Page

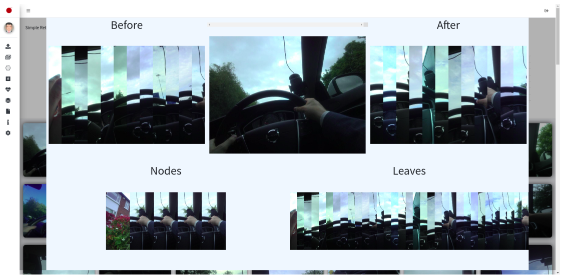



Figure 7: Retrieval Popup

year, month, and day so that a user can filter the results by selecting a desired year, month, or day(s) by clicking on the different graphs. Additionally, upon clicking an image, a popup will appear containing images preceding and succeeding the selected one, which are also clickable which makes them the new image at the center. This popup will also have the events that belong to that image as well as the images that are on the same event of that image. When an event on the popup is clicked, the user will be redirected to the events navigation page for further exploration. Lastly, if users click on an image in the leaves section, they will be redirected to the page containing the image information. We can see the page and popup, respectively, in figures 6 and 7.

# 4 ONGOING WORK

This section discusses ongoing Work on MEMORIA, highlighting its relevance to the possible participation at LSC'24, the lifelogging scenario and memory enhancement.

## 4.1 Large Language Models

Currently, work is being done to leverage LLM capabilities and integrate them into MEMORIA; we expect the natural language abilities of LLMs to be helpful in annotating images with meaningful descriptions derived from the various labels extracted from the different models. There are currently multiple objectives involving ways to integrate LLM capabilities in MEMORIA. MEMORIA will use Mistral 7B[6], a model from Mistral AI hosted locally as the model of choice. To ensure the model delivers the desired outcomes, we use prompt engineering to design a prompt that will allow better results, along with different techniques. Asking the model to return a JSON object or an XML tree is one of them, which makes it easy to extract the useful parts from the output and separate them from the additional text the model may return.

*4.1.1 Querying.* Using a Large Language Model is a great usability addition to the query experience as it can behave as an automatic translator to English (the language of extracted annotations), while also correcting spelling mistakes that may occur. Another possible application that can be investigated is using the LLM as a preprocessing tool to remove irrelevant words from the query, keeping only the keywords and allowing free text search.

*4.1.2 Description of events.* Another feature focused on usability is using a Large Language Model to generate a text description for an event using the annotations of the images that are part of the event. So far, if the model is given all annotations, it is very prone to hallucination as it is very hard to derive the context of images and events by using the objects that are on them. Thus, better results were produced by using only annotations that come from image caption models, such as ClipCap [7].

## 4.2 Event classification

A procedure to automatically identify and retrieve important moments for the user, without needing searches, is currently under development. Although the purpose of this feature differs from that of the challenge, it provides the user with genuinely important moments according to their preferences, an interesting feature in a lifelogging system.

The work involves acquiring everyday data, annotating the data, segmenting events, and suggesting relevant moments. MEMORIA is being customized to discover what is relevant to each person.

Event annotation was carried out to summarize the annotations of an event's images. For this, only the frequency of each concept in the images belonging to the event was saved.

Data collection is being prepared, in which participants will be invited to upload their images and classify events according to their importance. The events to be classified are those in the "Similar" layer, the most similar to real-life moments.

The annotations of the events and the relative relevance of all the concepts for each user will be gathered in this manner. Event annotations will then be linked to the ones the user deemed most pertinent. Eventually, using the data gathered, it will be possible to anticipate and retrieve potentially significant occasions for the user.

A reduced-weight version of MEMORIA has been developed specifically for data collection, to facilitate installation on participants' devices. Image processing will be carried out on a separate server, these images will be sent to the server via an API when the user uploads the images. The results of the processing will be obtained by an endpoint of the developed API. During this process, the images will not be saved on the server to preserve user privacy. Event segmentation is present without any changes. A new feature will allow users to classify semantic events, according to how important each one is to them. In addition, the interface allows users to view the most important moments planned along with the events previously classified, as Figure 8 shows.


**Figure 8: Visualization interface of the most important moments planned with previously classified events.**

Following data collection, an algorithm utilizing Natural Language Processing (NLP) and Content-based Filtering (CBF) approaches will be put into practice to suggest significant events (Figure 9). This algorithm will then compare the annotations of pertinent events to all of the user's event annotations, which will then output the events that most closely match.

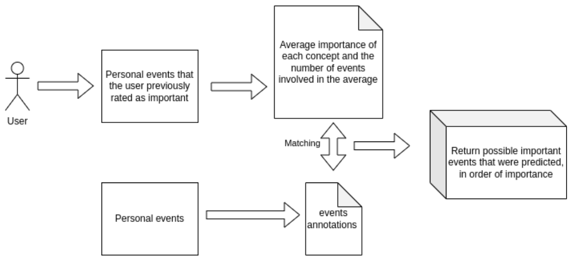



**Figure 9: Recommendation algorithm for important events.**

## 4.3 Album feature

Another feature in development is the album feature, in which a user can upload a folder to the system, and an album with all the uploaded images should be created. Additionally, a page to view said albums should also be developed. This feature could help the user to more easily store and search for a big event in their life that spans multiple days, such as a vacation.

## 4.4 Event Search

A new way to search for events via a text query is being developed. A user should be able to type a text query and see all the events relevant to that query. This search will retrieve events from the lowest event layer (image similarity). When a retrieved event is clicked, a new tab (or popup) will open, with the event navigation page to navigate through the closest events.

# 5 CONCLUSION

This paper presented the current state of the lifelog retrieval system MEMORIA (Memory Enhancement and MOment RetrIeval Application) with the intent of participating in the LSC24 event. One of the main goals of its participation is the evaluation of its performance and accuracy when compared with other similar systems. We hope to gain new insights into how the system can be further improved which, in turn, will advance the research in the field of lifelog retrieval.

# ACKNOWLEDGMENTS

This work was supported in part by National Funds through the FCT - Foundation for Science and Technology, in the context of the projects UIDB/00127/2020, CEECIND/01914/2017, and UID/04810/2020.

# REFERENCES

[1] Naushad Alam, Yvette Graham, and Cathal Gurrin. 2023. Memento 3.0: An Enhanced Lifelog Search Engine for LSC'23. In *Proceedings of the 6th Annual ACM Lifelog Search Challenge* (Thessaloniki, Greece) *(LSC '23)*. Association for Computing Machinery, New York, NY, USA, 41–46. https://doi.org/10.1145/3592573.3593103
[2] Ahmed Alateeq, Mark Roantree, and Cathal Gurrin. 2022. Voxento 3.0: A Prototype Voice-Controlled Interactive Search Engine for Lifelog. In *Proceedings of the 5th Annual on Lifelog Search Challenge* (Newark, NJ, USA) *(LSC '22)*. Association for Computing Machinery, New York, NY, USA, 43–47. https://doi.org/10.1145/3512729.3533009
[3] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. 2019. Character Region Awareness for Text Detection. https://doi.org/10.48550/ARXIV.1904.01941
[4] Cathal Gurrin, Graham Healy, Liting Zhou, Björn Þór Jónsson, Duc Tien Dang Nguyen, Jakub Lokoc, Luca Rossetto, Minh-Triet Tran, Wolfgang Hürst, Werner Bailer, and Klaus Schoeffmann. 2024. Introduction to the Seventh Annual Lifelog Search Challenge, LSC'24. In *Proc. International Conference on Multimedia Retrieval (ICMR'24)* (Phuket, Thailand) *(ICMR '24)*. New York, NY, USA.
[5] Olivier Jeunehomme1 and Arnaud D'Argembeau. 2020. Event segmentation and the temporal compression of experience in episodic memory. *Psychological Research* 84 (2020), 481–490. https://doi.org/10.1007/s00426-018-1047-y
[6] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. [n. d.]. Mistral 7B. ([n. d.]).
[7] Ron Mokady, Amir Hertz, and Amit H. Bermano. 2021. ClipCap: CLIP Prefix for Image Captioning. https://doi.org/10.48550/ARXIV.2111.09734
[8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. https://doi.org/10.48550/ARXIV.2103.00020
[9] Ricardo Ribeiro, Luísa Amaral, Wei Ye, Alina Trifan, António J. R. Neves, and Pedro Iglésias. 2023. MEMORIA: A Memory Enhancement and MOment RetrIeval Application for LSC 2023. In *Proceedings of the 6th Annual ACM Lifelog Search Challenge* (Thessaloniki, Greece) *(LSC '23)*. Association for Computing Machinery, New York, NY, USA, 18–23. https://doi.org/10.1145/3592573.3593099
[10] Ricardo Ribeiro, Alina Trifan, and António J R Neves. 2022. Lifelog Retrieval From Daily Digital Data: Narrative Review. *JMIR Mhealth Uhealth* 10, 5 (2 May 2022), e30517. https://doi.org/10.2196/30517
[11] Klaus Schoeffmann. 2023. LifeXplore at the Lifelog Search Challenge 2023. In *Proceedings of the 6th Annual ACM Lifelog Search Challenge* (Thessaloniki, Greece) *(LSC '23)*. Association for Computing Machinery, New York, NY, USA, 53–58. https://doi.org/10.1145/3592573.3593105
[12] Ly Duyen Tran, Binh Nguyen, Liting Zhou, and Cathal Gurrin. 2023. MyEachtra: Event-Based Interactive Lifelog Retrieval System for LSC'23. In *Proceedings of the 6th Annual ACM Lifelog Search Challenge* (Thessaloniki, Greece) *(LSC '23)*. Association for Computing Machinery, New York, NY, USA, 24–29. https://doi.org/10.1145/3592573.3593100
[13] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. https://doi.org/10.48550/ARXIV.2207.02696
[14] Jialian Wu, Jianfeng Wang, Zhengyuan Yang, Zhe Gan, Zicheng Liu, Junsong Yuan, and Lijuan Wang. 2022. GRiT: A Generative Region-to-text Transformer for Object Understanding. *arXiv preprint arXiv:2212.00280* (2022).
[15] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. 2022. Zero-Shot Video Question Answering via Frozen Bidirectional Language Models. arXiv:2206.08155 [cs.CV]

APPENDIX B

# Experiment Protocol

*This chapter presents the experiment protocol developed and provided to the participants in the experiment carried out for this dissertation. The purpose of the protocol is to guide participants through the experiment.*

# 1. Introduction

As part of the dissertation *“Memory retrieval through lifelogging using the MEMORIA application”*, this document has been developed which depicts the experiment protocol, a guide for the participants of the experiment that will be developed for this dissertation. Step-by-step instructions on how participants can complete the experiment are presented in this protocol.

The experience will unfold in the application Memory Enhancement and MOment RetrIeval Application (MEMORIA), system for retrieving and analyzing egocentric images using multimodal datasets (Figure 1). The MEMORIA system allows users to submit personal data in a variety of digital formats, such as images, videos, audio, physiological signs, etc. Especially focused on images, this system processes the uploaded data, providing various functionalities for detailed data exploration and analysis [1] .

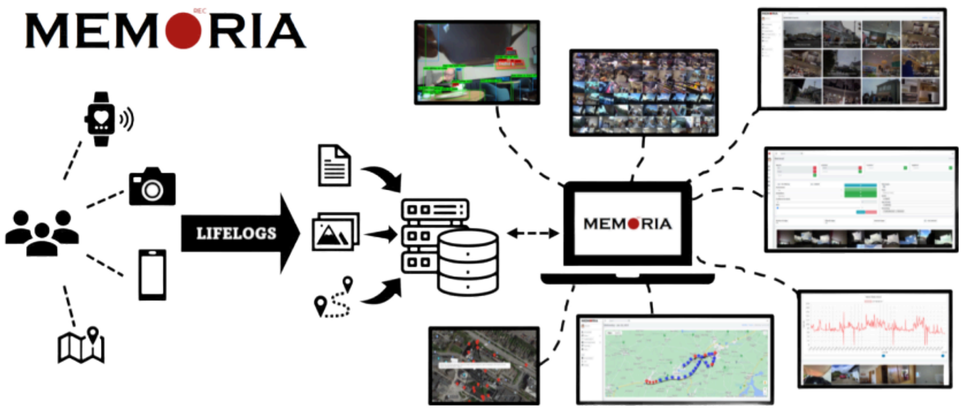



Figure 1 - An overview of the MEMORIA system.

The aim of this dissertation is to develop and implement algorithms capable of extracting and organizing information obtained from images/videos that are useful for the analysis, identification and automatic retrieval of relevant memories in the user’s daily lives, without the need for research. This new functionality is expected to improve users' memory capacity, providing truly meaningful moments by adapting to individual preferences.

In order to help implement the algorithm, data collection is being organized through an experiment with some participants. In this data collection, participants are invited to use MEMORIA. First to upload some of their personal images, followed by the classification of moments/events generated by the system, according to how important they consider the event to them. Each moment/event created by MEMORIA is a set of images, from the personal images uploaded, that represent an occurrence in a given place, time with a beginning and end. This process allows statistical collection that allows events to be associated with user evaluation.

Through the use of the MEMORIA application, the experiment attempts to investigate memory retrieval with an emphasis on how technology can enhance our recollection of certain memories. The impact of capturing and organizing daily photos on memory recovery and emotional well-being will be explored.

# 2. Participants

Participants should have some knowledge of the basic sciences of Computer Engineering, specifically Software Engineering and Information Systems. It is also necessary to know and be able to use Docker technology and a basic understanding of command-line interface operations and permissions to execute commands successfully.

# 3. Materials/Equipment

To carry out the experiment, participants must have a computer with Docker and Docker Compose installed [2]. They also need access to the internet, as well as some personal images of the participant.

# 4. Procedure

## 4.1. Setup Instructions

To participate in the experiment, follow these detailed setup instructions to ensure everything works:

1. **Docker Installation**
    a. If you don't have Docker installed, visit the official Docker website [2] and download the Docker Desktop application compatible with your operating system (Windows, macOS or Linux).
    b. Follow the installation guide provided by Docker to install Docker Desktop on your computer.
    c. Once installed, open Docker to ensure it is running correctly on your system.
2. **Docker Compose Installation**
    a. Often included with Docker Desktop installations, make sure it is installed and configured correctly according to the requirements of your operating system.
3. **Downloading the MEMORIA**
    a. Access the MEMORIA code in a web browser using the link [3] . Transfer the code folder to your computer, and extract the ZIP file into the desired directory.
    b. In the transferred folder you will find a support file “README.md”, with the steps for installing and running MEMORA. These steps will be described below in this protocol, so just follow the instructions presented here.
4. **Creating an environment file**
    a. In the application's root directory, create an environment file, a file that stores all the configuration settings for the application.
    b. Set the file name to ".env", and copy the following content into the file (Figure 2):

```
POSTGRES_DB=memoria_db2
POSTGRES_USER=admin
POSTGRES_PASSWORD=admin
DJANGO_SUPERUSER_USERNAME=admin
DJANGO_SUPERUSER_PASSWORD=admin
DJANGO_SUPERUSER_EMAIL=admin@ua.pt
API_URL=https://193.136.175.76:443
```

Figure 2 - Contents of the environment file.

    c. If this file already exists with exactly the content of Figure 2,, you do not need to create it again.

## 4.2. Experiment Steps

After the setup instructions, you can proceed to the experimental steps, the tasks you have to perform on the MEMORIA system. During this process, make sure that you are not connected to a VPN (Virtual Private Network).

1. **Running the MEMORIA Application**
    a. In the terminal, in the root directory of the application, execute the following command to start the MEMORIA using Docker (Figure 3):

```
docker-compose up --build
```

Figure 3 - Command to start the Docker services and rebuild images if necessary.

    b. For the first run of the application, use the command specified above (Figure 3). For subsequent runs of the application, you can simply use the command shown in Figure 4:

```
docker-compose up
```

Figure 4 - Command to start the Docker services without rebuilding images.

    c. Before running the command in Figure 4, you may have to stop Redis or PostgreSQL services, if they are in use (Figure 5).

```
sudo service redis-server stop
sudo service postgresql stop
```

Figure 5 - Commands to stop Redis and PostgreSQL services.

2. **Accessing the MEMORIA Application**
    a. Open a web browser and navigate to http://0.0.0.0:8000/ to access the MEMORIA application.
3. **Account Creation and Login**
    a. On the MEMORIA application's landing page, select the option to create a new account or log in if you already have one.
    b. Follow the on-screen instructions to complete the account creation or login process.
4. **Uploading Images**

a. Images need to have one of the following extensions: '.png', '.jpg', '.jpeg', '.JPG', '.JPEG'. If, by chance, you upload an image with a different extension, don't worry, the system will simply ignore it and not save it. Each image must have a unique name, ignoring non-alphanumeric characters to differentiate them. For example, an image with the name “img1.jpg” is uploaded, and then another whose name is “img_1.jpg”, the system considers that they both have the same name, so the second image will not be uploaded to the MEMORIA. In order for it to be uploaded successfully, you have to change the name of the second one following the guidelines mentioned in this point.
b. Before proceeding with image submission, it is important to try to follow these guidelines. The desired images should not be individually selected for the system, as this could compromise the integrity of the study, since the participant will only be selecting the images considered most relevant. Instead, it is recommended that the participant uploads a complete set of images from one day to MEMORIA, without prior selection or removal. The ideal is to submit images from days with a large volume of available photographs.
c. **It's important to note that the images sent by the participant to the system are never stored on any device other than the participant's own during the experiment**. Only the image data in byte form is transferred to a server, but neither the images themselves nor their corresponding bytes are stored on the server at any point in the experience process, not even temporarily. This practice aims to fully respect the participant's privacy, thus allowing the images to be sent without prior selection or removal, without worrying about the possibility of third viewing them.
d. Navigate to the image upload page by clicking on "File Uploader" in the sidebar menu.
e. Upload some of your personal images to the system, note that you can upload several images simultaneously, but the system only saves them one by one. **So stay on the page until all the images in the dropzone have been confirmed with a check mark (Figure 6)**. Also, remember to follow the specific criteria for images mentioned above.

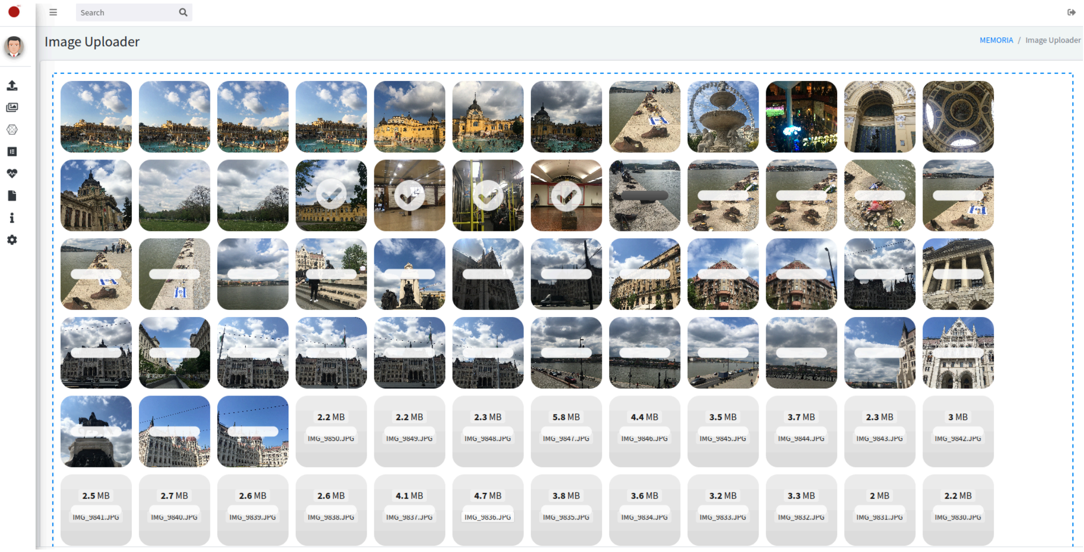



Figure 6 - Image upload page saving images.

f. As the images are uploaded to the system, they are sent for processing on a server. This can be a lengthy process, given the complexity involved in image analysis.

5. **Getting image processing results**

 a. Navigate to the daily image display page by selecting the "Gallery" option and then the "Day" option in the sidebar menu. On this page you are encouraged to explore the uploaded images, organized according to the date each one was taken (Figure 7).

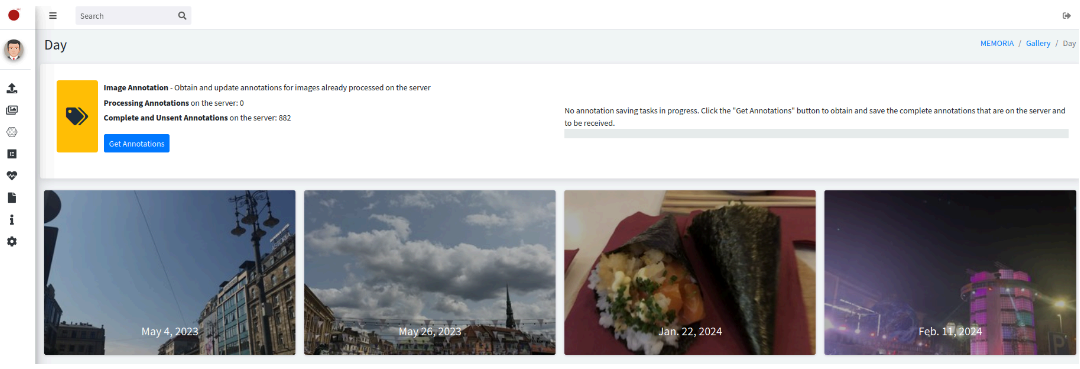



Figure 7 - Daily image display page.

b. On this page you can view two important values for the next step, “Processing Annotations” and “Complete and Unset Annotations”, as depicted in Figure 8. The “Processing Annotations” value represents the number of annotations that are still being processed on the server, that is, results that have not yet been finalized or defined. The “Complete and Unset Annotations” value indicates the number of annotations that have been completely processed and that already have a result, but that the user (participant, in this case) has not yet received in the application. When “Complete and Unset Annotations” is different from 0, you can click on the “Get Annotations” button to obtain the corresponding annotations that are already complete. When you click on this button, on the side, you will see a progress bar indicating the progress in storing the results obtained (Figure 9). **It is advisable to click the button again only when progress reaches 100%, or when a message on the bar indicates “No annotation saving tasks in progress.”.**

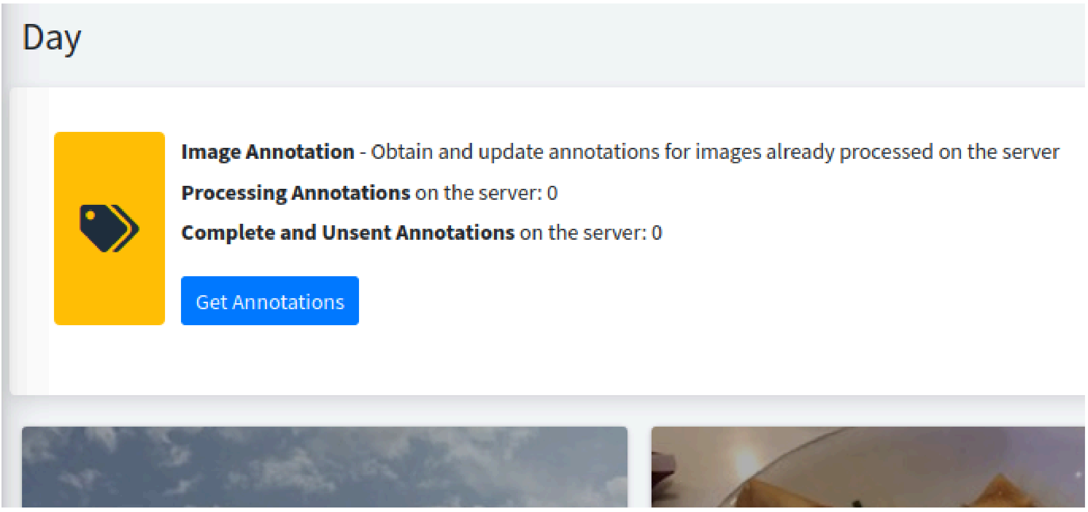



Figure 8 - Information about the annotations on the server, on daily image display page.

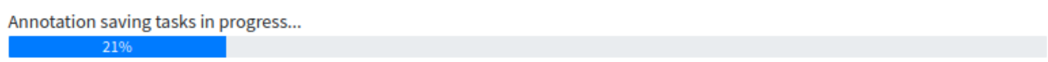



Figure 9 - Progress bar for saving annotations tasks, on the daily image display page.

c. When the number of “Complete and Unset Annotations” reaches a significant level, click the “Get Annotations” button to obtain the processing results of the loaded images. If the entire process is taking a long time and there is no results storage task in progress (you can check this in the progress bar on the left, Figure 9), you can let all “Processing Annotations” go to the “Complete and Unset Annotations” state. In other words, wait for “Processing Annotations” to reach zero value and then click on “Get Annotations”. During this waiting period, you can stop MEMORIA from running and even turn off the computer if you want, but **before stopping execution, make sure that there are no results storage tasks in progress by checking the progress bar, Figure 9.** To resume running MEMORIA after an interruption, simply run the command in Figure 4 again in the application's root directory. You will continue to have your account created and the images you uploaded.
d. When you have no more images to upload, the “Processing Annotations” and “Complete and Unset Annotations” values are both 0 (visualize their respective values as shown in Figure 8), and there isn't a results storage task running (see the progress bar, Figure 9), you can proceed to the next step. If you are curious, you can view the annotations and results obtained from the server on the page you are on, the “Day” page, by clicking on the desired day, and then on the desired image (Figure 7).

6. **Creating events**
    a. Before detailing the creation of events, it is crucial to define an event. Each moment/event created by MEMORIA is a set of images, from the personal images uploaded, that represent an occurrence in a given place, time with a beginning and end. MEMORIA visually organizes these events in a hierarchical structure of five levels: Days, Parts of Day, Locations, Environment and Image Similarity (Figure 10). The organization begins with the aggregation of images into “Day Events”, which are sets of images taken on the same day. Each “Day Event” is then subdivided into “Parts of Day”, such as morning or afternoon, reflecting when the images were captured. Within these, the images are grouped into “Location Events”, based on the specific location where they occurred. These events are further categorized by “Environment”, distinguishing, for example, indoor and outdoor spaces. Finally, the last layer is “Similarity”, where events are formed by similar images, possibly representing the same moment or theme (Figure 10).

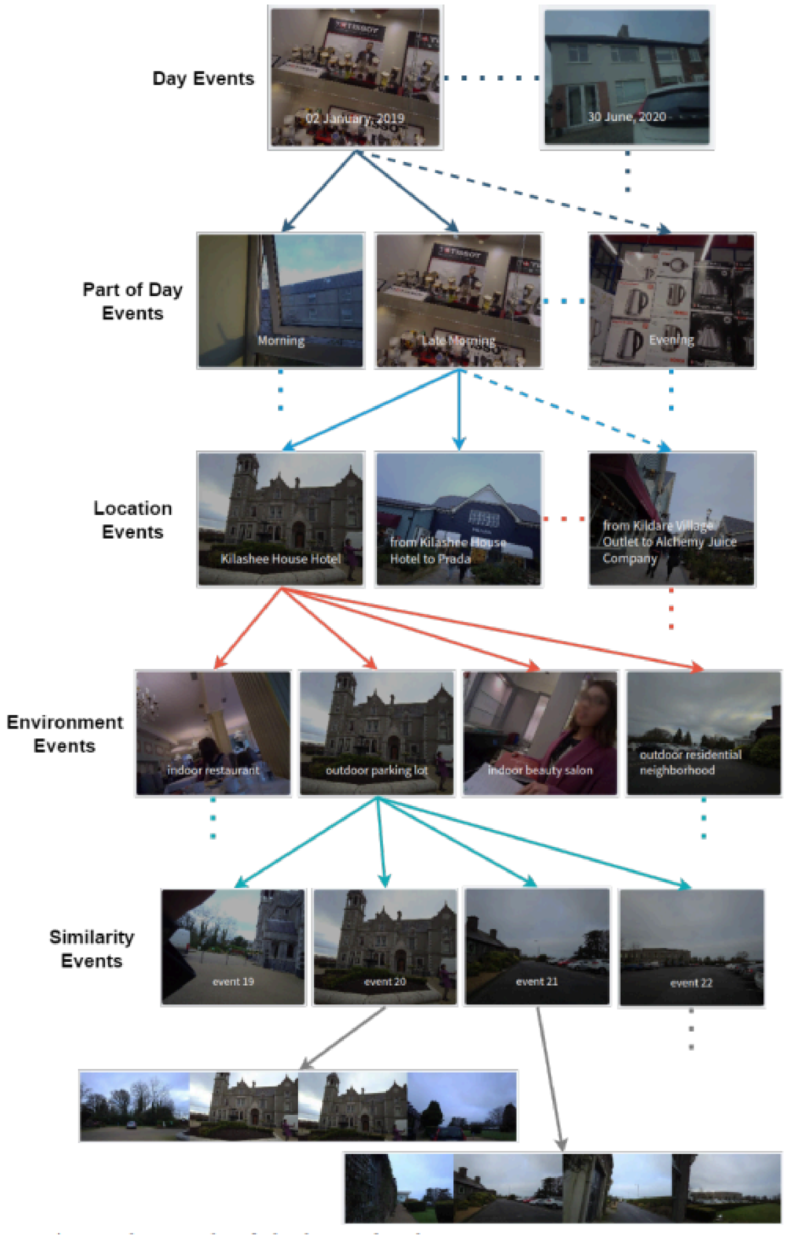



Figure 10 - Hierarchical organization of events in MEMORIA.

b. **Before proceeding with creating events, ensure that you no longer have any images to upload.** Event creation is a time-consuming process, and if you upload an image after creating events, you will have to perform event creation again as the image will not be in this organization.

c. From the sidebar menu, access the event creation page by clicking on "Event Segmentation" and then "Events" (Figure 11).

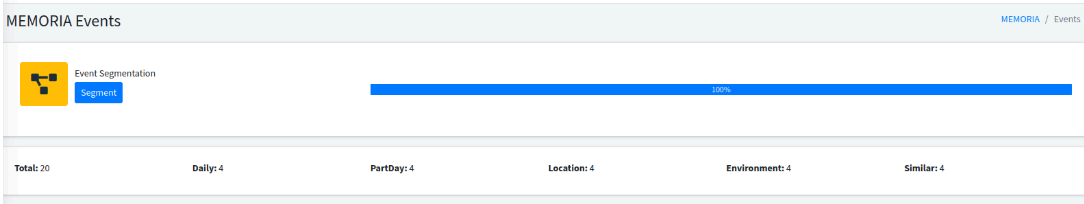



Figure 11 - Event creation page.

d. Press the "Segment" button for MEMORIA to start segmenting events, i.e. creating events (Figure 11). This process can take a while, and next to the button there will be a progress bar showing the progress of the event segmentation. When the progress is 100% it means that all the events have been created and you can move on to the next step.

7. **Classifying the events**

   a. Access the events created via the sidebar menu by selecting "Event Segmentation" and then "All" (Figure 12).

   b. Explore the events, navigating through the different layers (Days, Parts of Day, Locations, Environment and Similarity), until you understand the organization provided.

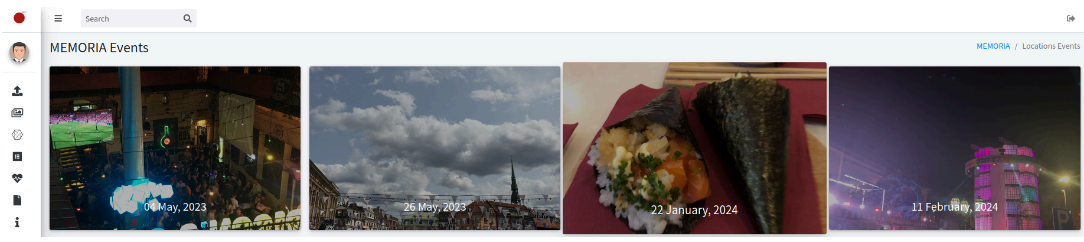



Figure 12 - Events page.

   c. Classify the events you want from the Similar layer (last layer) according to how important you consider the event to you. In each similar event, there is a classification bar available for this purpose (Figure 13). You can edit or remove a previously assigned rating. To remove a rating, simply click again on the star corresponding to the event's current rating (Figure 13).


Figure 13 - Example of a similar event with the rating bar.

d. You can view the classified events on the important moments page by clicking on "Event Segmentation" and then "Important Events" in the sidebar menu (Figure 14). On this page you can also edit and delete classifications.


Figure 14 - Important events page.

e. When you finish the event classification task, and there are no more events you want to classify, on the important moments page, click on the “Export” button located in the upper right corner (Figure 14). When you press this button, a csv (Comma-separated values) file will be transferred to your computer. This file will contain the annotations of the classified events, and the respective importances calculated during the classification process (Figure 15). After this, please conclude the experience by sending the downloaded file to the email address: evabartolomeu@ua.pt . Thank you for your participation.

```
Annotation,Importance
person,3.57416267942587
wig,5.0
dry,3.47945205479452
person in the snow after a snowstorm.,5.0
snow covered tree branch,5.0
aerial view of snow - covered trees in a park.,5.0
a large patch of white snow,5.0
the tree is green,3.6388888888889
a large waterfall of white snow,5.0
aerial view of the snow - covered forest.,5.0
snow covered trees visible in the background.,5.0
the snow covered ground,5.0
```

Figure 15 - Example of an exported file, with event annotations classified and the degree of importance of each annotation.

## 5. Ethical considerations

At no point in this experiment will the images uploaded by the participants be stored on any device other than the participant's own. The only information that will be collected during this experiment is the concepts/words/expressions attributed to each image by the application, as well as the degree of importance calculated to these concepts through the evaluations made by the participant throughout the experiment. This experiment seeks to take ethical considerations into account and preserve users' privacy.

## 6. Contact Information

In case of questions or problems, participants can contact via email: evabartolomeu@ua.pt .

# 7. References

[1] R. Ribeiro, L. Amaral, W. Ye, A. Trifan, A. J. R. Neves, and P. Iglésias, «Memoria: A memory enhancement and moment retrieval application for lsc 2023», in Proceedings of the 6th Annual ACM Lifelog Search Challenge, ser. LSC ’23, Thessaloniki, Greece: Association for Computing Machinery, 2023, pp. 18–23, isbn: 9798400701887. doi: 10.1145/3592573.3593099. [Online]. Available: https://doi.org/10.1145/3592573.3593099.

[2] Docker, Get Docker, Accessed: 2024-03-06. [Online]. Available: https://docs.docker.com/get-docker/.

[3] MEMORIA code, Created: 2024-04-17, Accessed: 2024-04-17. [Online]. Available: https://drive.google.com/drive/folders/1dh1K0zfbh9HFZaz90P27bmrfnsLt14Ak?usp=drive_link .