A logo with a white ball and black text

AI-generated content may be incorrect.A black background with white text

AI-generated content may be incorrect.

**Helwan University**

**Faculty of Computing and Artificial Intelligence**

**Computer Science Department**

A black letter on a white background

AI-generated content may be incorrect.

A graduation project dissertation by:

Abdalla Mohamed Said [20210556]

Marwa Omar Muhammad Mahmoud [20210900]

Toqa Mohamed Ahmed El-Saady [20210242]

Rana Essam El-Deen Essa [20210335]

Nada Mohamed Fekry Ahmed [20211002]

Yousef Mostafa Ghanem Abdelaziz [20211111]

Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computing & Artificial Intelligence at the Computer Science Department,

the Faculty of Computers & Artificial Intelligence, Helwan University

Supervised by:

Dr. Hala Abdel-Galil

June 2025

**جـامـعـة حـلـوان**

**كـليـة الحـاسـبـات والـذكـاء الاصطناعي**

**قـسـم علوم الحاسب**

A black letter on a white background

AI-generated content may be incorrect.

رسالة مشروع تخرج مقدمة من:

عبدالله محمد سعيد [20210556]

مروه عمر محمد [20210900]

تقى محمد أحمد السعدي [20210242]

رنا عصام الدين عيسى [20210335]

ندى محمد فكري [20211002]

يوسف مصطفى غانم عبد العزيز [20211111]

رسالة مقدمة ضمن متطلبات الحصول على درجة البكالوريوس في الحاسبات والذكاء الاصطناعي، بقسم علوم الحاسب،

كلية الحاسبات والذكاء الاصطناعي، جامعة حلوان

تحت إشراف

أ.د/ هالة عبدالجليل

يونيو/حزيران 2025

# **Abstract**

This project introduces an Augmented Reality (AR) tour guide that uses Visual Place Recognition (VPR) to recognize locations and display relevant information. The system works by analyzing images from the camera and matching them with a database of known places. We use deep learning models like ResNet-50, Vision Transformer (ViT), and TransVPR to extract features and compare scenes.

Our model was trained and evaluated using more than 18 different datasets, which were carefully cleaned, filtered, and augmented to improve performance and generalization. The final dataset consists of 77 classes, with each class containing an average of 3,000 images.

The system also includes an interactive chatbot powered by **Llama 3**, a **transformer-based Large Language Model (LLM)** developed by Meta AI. The chatbot uses a **Retrieval-Augmented Generation (RAG)** approach, enhanced with **Hugging Face embeddings** (via the all-MiniLM-L6-v2 model) to convert input and context into semantic vectors. These embeddings are stored and efficiently searched using **FAISS (Facebook AI Similarity Search)**, which is optimized for fast similarity retrieval over large vector databases. This enables the chatbot to provide fast, context-aware responses based on a curated knowledge base of 13+ trusted English references.  
The chatbot can operate independently to answer user questions, or it can work alongside the classifier: when the system detects an artifact and the user taps on it, the chatbot provides detailed and relevant information seamlessly.

The goal is to create a smooth and intelligent AR experience that helps users explore and learn about their surroundings by simply pointing their camera and interacting naturally.

**Keywords:**

1. Augmented Reality (AR)
2. Tour Guide System
3. Visual Place Recognition (VPR)
4. Large Language Model Meta AI, version 3 (Llama 3)
5. Retrieval-Augmented Generation (RAG)
6. Hugging Face Embeddings
7. Facebook AI Similarity Search (FAISS)
8. LangChain
9. Graph-Reduction-On-Quanta (Groq)
10. Interactive Chatbot
11. Deep Learning
12. Convolutional Neural Networks (CNN)
13. ResNet-50
14. Vision Transformer (ViT)
15. Transformer Architecture
16. TransVPR
17. Feature Extraction
18. Image Retrieval
19. Real-time Inference

Acknowledgment

In order to complete this project successfully, we were never alone; we are deeply grateful to every individual who contributed to making our goal a reality, even if it was simply through spiritual support.

Our heartfelt appreciation goes to our professor, Dr. Hala Abdel-Galil, whose feedback, guidance, and immense patience have been invaluable throughout the entire year. Without her, this journey and the achievement of our objective would not have been possible. Her generous sharing of knowledge was a cornerstone of our progress.

Finally, we cannot forget to express our sincere thanks to our families, especially our parents. Their unwavering belief in us has been a constant source of strength, lifting our spirits and keeping our motivation alive during every step of this journey.

Table of Contents

[**Abstract** 1](#_Toc200608984)

[Acknowledgment 3](#_Toc200608985)

[**Chapter 1: Introduction** 6](#_Toc200608986)

[1.1 Problem Statement 6](#_Toc200608987)

[1.2 Scope and Objectives 7](#_Toc200608988)

[**Chapter 2: Literature review and overview** 9](#_Toc200608989)

[2.1 Augmented Reality (AR) 9](#_Toc200608990)

[2.2 Computer vision 11](#_Toc200608991)

[2.3 Natural language processing 15](#_Toc200608992)

[2.4 Chatbot 15](#_Toc200608993)

[2.4.1 Llama 3 Overview and architecture 16](#_Toc200608994)

[2.4.1.1 A Decoder‑only Transformer Architecture 16](#_Toc200608995)

[2.4.2 Retrieval-Augmented Generation (RAG) 20](#_Toc200608996)

[2.4.3 Embedding and Document Processing 20](#_Toc200608997)

[2.4.4 Integration of Groq Hardware Accelerator 20](#_Toc200608998)

[2.4.5 Chatbot Flow and Session Management 21](#_Toc200608999)

[2.5 previous work 22](#_Toc200609000)

[2.5.1 Hassan et al. (2023) 22](#_Toc200609001)

[2.5.2 Chowanda and Sutoyo (2019) 22](#_Toc200609002)

[2.5.3. Teixeira et al. (2021) 23](#_Toc200609003)

[2.6 Overview 23](#_Toc200609004)

[**Chapter 3: System analysis and design** 24](#_Toc200609005)

[3.1 User Requirements 24](#_Toc200609006)

[3.2 Functional Requirements 25](#_Toc200609007)

[3.3 Non-functional Requirements 28](#_Toc200609008)

[**Class diagram** 29](#_Toc200609009)

[**ERD diagram** 30](#_Toc200609010)

[use case diagram 31](#_Toc200609011)

[activity diagram 32](#_Toc200609012)

[**Sequence diagram** 36](#_Toc200609013)

[**Chapter 4: implementation** 42](#_Toc200609014)

[4.1 Data preprocessing 42](#_Toc200609015)

[4.1.1 Image Data 42](#_Toc200609016)

[4.1.2 Text Data: 43](#_Toc200609017)

[4.2 Implementation details for Augmented Reality Tour Guide 45](#_Toc200609018)

[4.2.1 Computer Vision 45](#_Toc200609019)

[4.2.2 Chatbot System 50](#_Toc200609020)

[**Chapter 5: results and testing** 53](#_Toc200609021)

[5.1 Evaluation metrics in Computer Vision models 53](#_Toc200609022)

[5.1.1 ResNet50 Standalone Evaluation 53](#_Toc200609023)

[5.2.1 TransVPR (ResNet50 + ViT) Evaluation 58](#_Toc200609024)

[5.2.2 Comparative Analysis 63](#_Toc200609025)

[5.3 Evaluation and Metrics in Chatbot model 64](#_Toc200609026)

[5.3.1 Evaluation Insights: 65](#_Toc200609027)

[**Chapter 6: Future Work** 66](#_Toc200609028)

[6.1 Multilingual Expansion 66](#_Toc200609029)

[6.2 Mobile Application Development 66](#_Toc200609030)

[6.3 Augmented Reality (AR) Enhancements 66](#_Toc200609031)

[6.4 Personalized Museum Tours Using AI 66](#_Toc200609032)

[**References** 67](#_Toc200609033)

# **Chapter 1: Introduction**

## 1.1 Problem Statement

The core problem addressed by the AR Tour Guide project lies in enabling machines to recognize real-world objects—such as monuments and artifacts—and provide meaningful, context-aware explanations about them through natural language. This requires the integration of visual understanding, information retrieval, interactive chatbot communication, and language generation in real time, all within an accessible, web-based AR environment.

This task presents several challenges, including the accurate recognition of diverse physical objects under various real-world conditions, retrieving the most relevant information from a knowledge base, and generating user-friendly responses that suit different audiences.

Here are some key aspects of the problem:

1. **Visual Recognition in Real-World Environments**  
   Teaching machines to correctly identify monuments, artifacts, or other physical objects using live camera input is a major challenge. The model must account for variability in lighting, angles, object sizes, wear and tear, partial views, and visual similarity between items.
2. **Knowledge Retrieval and Relevance**  
   Once an object is identified, retrieving the most relevant and accurate information from a pre-collected dataset or external source is essential. This involves matching the object label to the right knowledge entry while handling potential ambiguities or variations in naming.
3. **Documentation of Historical Information from Trusted Sources**  
   It is crucial to ensure that all historical information is documented from trusted and authoritative sources to guarantee accuracy, credibility, and reliability of the knowledge presented to users.
4. **Natural Language Response Generation**  
   Generating explanations that are coherent, informative, and suited to the user's context is another layer of complexity. The language model must transform retrieved facts into responses that are grammatically correct, easy to understand, and engaging for the user.
5. **Real-Time Integration in a Web-Based AR Interface**  
   All components must work efficiently in real time on a web platform, which adds technical challenges around performance, latency, and compatibility across devices and browsers.
6. **Generalization and Adaptability**  
   The system must be capable of handling a wide variety of objects across different domains—historical, artistic, scientific, etc.—and scale to accommodate new data without retraining from scratch. It also needs to support users with different needs, such as tourists, students, or visually impaired individuals.

## 1.2 Scope and Objectives

An AR Tour Guide project aims to bridge the gap between physical objects and digital knowledge by combining real-time computer vision with natural language generation. The core idea is to allow users to point a web-based AR camera at real-world artifacts—such as historical monuments or museum exhibits—and receive descriptive, context-aware explanations. The system uses visual recognition to classify the object and Retrieval-Augmented Generation (RAG) to generate natural language descriptions based on relevant data sources.

The objectives of the AR Tour Guide project are:

1. **Real-Time Object Recognition**  
   The primary objective is to enable the system to accurately recognize real-world physical objects (e.g., statues, monuments, or artifacts) through a live camera feed using computer vision models. This recognition must be robust under varying conditions such as lighting, angles, and partial occlusions.
2. **Relevant Knowledge Retrieval and Question Answering**  
   Once the user taps when an object is detected, the system should retrieve the most relevant and accurate information associated with that object’s label. In addition, users can interact with a standalone chatbot by typing questions directly. The chatbot will use Retrieval-Augmented Generation to search a knowledge base and provide intelligent responses, even without a visual input.
3. **Integration of Trusted Static Information in Knowledge Retrieval**

To insure the reliability of responses, the system must incorporate verified information from trusted and authoritative sources within the Retrieval-Augmented Generation (RAG) framework to ensure the accuracy, credibility, and reliability of the historical knowledge provided to users.

1. **Natural Language Explanation Generation**  
   The system aims to generate fluent, coherent, and user-friendly responses tailored to the needs of different users—such as tourists, students, or researchers. Using large language models (e.g., LLaMA 3) within the RAG framework, the retrieved knowledge is transformed into human-readable descriptions or chatbot answers that resemble natural conversation.
2. **Web-Based AR Integration**  
   A key objective is to develop the system as a web application that runs on standard browsers without requiring native mobile apps. This increases accessibility and usability across different platforms, ensuring a smooth AR experience through a camera-enabled web interface.
3. **Educational and Accessibility Enhancement**  
   The project also aims to enhance learning and accessibility by enabling users—especially those with visual impairments—to understand real-world artifacts through generated audio or text descriptions. The system should be usable in museums, classrooms, and heritage sites.
4. **Scalability and Adaptability**  
   The system is designed to support a wide range of objects and domains beyond historical artifacts, such as educational tools or commercial products. It should allow for easy integration of new data sources or expansion into new sectors without significant retraining
5. **Backend Development for Secure and Scalable Systems**

- Design and implement a secure backend using Spring Boot and Spring Security to manage user authentication, role-based access, and session handling.

- Support robust APIs for interacting with AR modules, data retrieval, and frontend interfaces.

- Integration of JWT-based authentication for stateless sessions and secure user interactions across roles (Admin, User).

# **Chapter 2: Literature review and overview**

## 2.1 Augmented Reality (AR)

AR Tour Guide technology combines two powerful tools: **Augmented Reality (AR)** and **Retrieval-Augmented Generation (RAG)**. It uses **computer vision** to recognize real-world objects—like monuments or artifacts—through a camera, and then uses a **smart language model**, supported by **semantic search and curated reference data**, to explain what the user is seeing. The result is an **interactive, informative experience** where users get real-time knowledge simply by viewing objects through a web-based AR interface.

This technology brings together visual understanding and intelligent language generation to create helpful, human-like explanations.

It can be applied in many fields:

* 1. **Tourism and Museums**  
     Visitors can point their camera at landmarks or exhibits and get instant information, making tours more interactive and self-guided without needing a human guide.\

1. **Education**  
   Students can use it during museum trips or classroom activities to learn more about historical items or scientific tools by just scanning them.
2. **Cultural Preservation**  
   Institutions can document and share stories about heritage objects in an engaging way, making cultural content more accessible to the public.
3. **Customer Support in Retail**  
   Stores can use the system to help customers get product information just by scanning an item on the shelf.
4. **Healthcare Training**  
   Medical students could use AR to identify tools or body models and receive detailed explanations pulled from verified sources.
5. **Manufacturing and Maintenance**  
   Technicians can scan machinery parts and receive on-the-spot guidance or documentation related to the component being viewed.
6. **Accessibility**  
   For users with disabilities, the system can describe objects visually and verbally, making real-world content easier to understand.

The importance of an AR tour guide lies in its ability to seamlessly connect real-world environments with enriched digital information, making cultural and historical sites more accessible and engaging to a diverse audience. It enhances user experience by:

* providing interactive, context-aware content
* improving navigation and learning
* supporting personalized tours
* fostering inclusivity by catering to different user needs, including those with disabilities.

Additionally, the AR tour guide advances technology by integrating computer vision, augmented reality, and natural language processing to deliver immersive and informative experiences that bring history and culture to life in a way that resonates with users.

So, AR Tour Guide technology plays a vital role in connecting real-world visuals with intelligent, accessible explanations. By combining augmented reality with retrieval-based AI, it enhances learning, accessibility, and engagement across multiple domains. Its advancement supports more natural human-computer interaction, making knowledge easier to access, understand, and explore in both educational and practical settings.

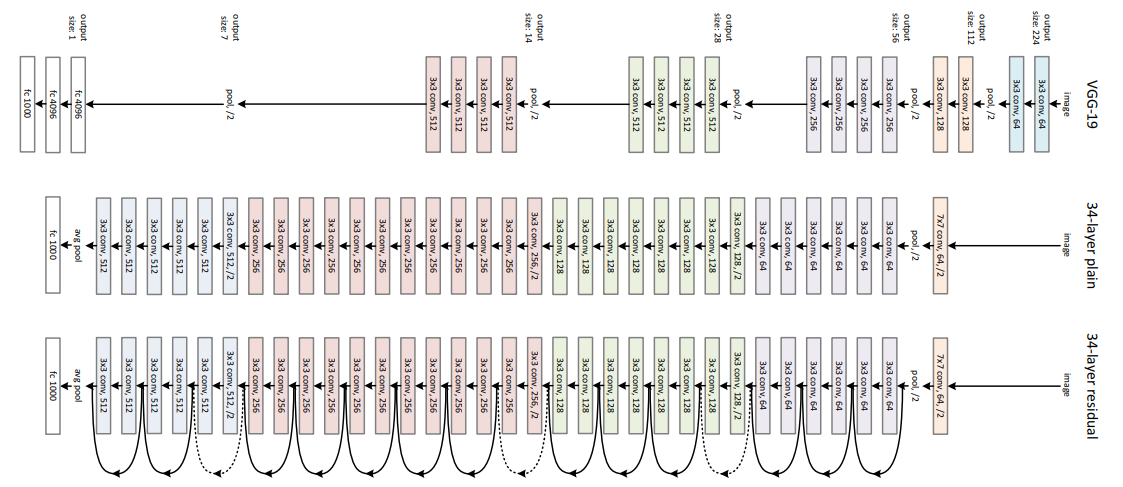
## 2.2 Computer vision

Computer Vision plays a pivotal role in the AR Tour Guide by enabling the system to automatically detect and recognize landmarks, monuments, and historical structures in real time. Using advanced image recognition techniques, the application identifies what the user is viewing through the device's camera and dynamically overlays informative content related to that object.

This approach provides a smooth and intuitive experience, allowing users to interact with their surroundings without the need for manual input. By accurately linking physical locations to digital content, Computer Vision enhances engagement and makes the exploration of historical sites more informative and immersive.

#### 2.2.1 Models overview

##### 2.2.1.1 **ResNet50 – Based on “Deep Residual Learning for Image Recognition” paper**

ResNet50 is a 50-layer deep convolutional neural network introduced in the **ResNet** paper by He et al. (2015). The key innovation of ResNet is the introduction of **residual connections**, or **skip connections**, which allow the model to learn **residual functions** instead of direct mappings. 

**Fig1: Residual-Network Architecture:** This network uses a 34-layer plain network architecture inspired by VGG-19 in which then the shortcut connection is added. These shortcut connections then convert the architecture into a residual network.

#### LightboxKey Concepts:

1. **Residual Block**: Each block learns a residual function F(x)=H(x)−xF(x) = H(x) - xF(x)=H(x)−x, where H(x)H(x)H(x) is the desired mapping and xxx is the input. The block outputs F(x)+xF(x) + xF(x)+x, allowing gradients to flow through the identity shortcut.

Fig2: F(x) = H(x) - x is the "residual", which is usually smaller and easier to learn The network combines this residual F(x) with the input x using a **shortcut** or **skip connection**:   
which gives H(x) = F(x) + x.

1. **Bottleneck Architecture**: ResNet50 uses a bottleneck design — each block has 3 layers: 1x1 (for reducing dimensions), 3x3 (processing), and 1x1 (expanding back).
2. **Depth and Layers**: ResNet50 includes 49 convolutional layers followed by a fully connected (FC) layer, totaling 50 layers.

A group of blue rectangular objects with black text

AI-generated content may be incorrect.

Fig3: ResNet50 model architecture

1. **Vanishing Gradient Solution**: By using skip connections, ResNet enables training much deeper networks without suffering from vanishing gradients.
2. **Feature Extraction**

In the initial phase, ResNet50 was fine-tuned on our custom Augmented Egyptian Monuments dataset. During this stage, a dropout layer with a dropout rate of 20% was applied before the final fully connected layer to mitigate overfitting. After training, the model achieved strong performance as a standalone classifier.

Subsequently, for integration into the full TransVPR architecture, ResNet50 was repurposed as a pure feature extractor. The classification head—including the dropout and fully connected layers—was removed and replaced with an identity layer (nn.Identity()). This modification enabled the model to output high-dimensional feature maps, which were then passed directly to the Vision Transformer (ViT) for further processing and final classification.

##### **2.2.1.2 Vision Transformer (ViT) – Based on “An Image is Worth 16×16 Words” Paper**

The Vision Transformer (ViT) is a novel architecture that applies a pure transformer model to image classification tasks by representing images as sequences of fixed-size patches, analogous to token sequences in natural language processing. This allows ViT to exploit global context from the input image, unlike CNNs which focus on local features. A diagram of a transformer

AI-generated content may be incorrect.

**Vision Transformer (ViT)** architecture

* + - 1. **Patch Embedding**

Instead of using convolutional filters, ViT first splits the input image into fixed-size patches (e.g., 16x16), then flattens and linearly projects each patch into a lower-dimensional embedding space. These patch embeddings form a sequence similar to word embeddings in NLP.

In our implementation, since the output from ResNet50 is a 7×7×2048 feature map, we treat each 1×1 region (patch size = 1) in the 7×7 grid as a token, resulting in 49 tokens. These are then linearly projected into a fixed hidden size of 768.

* + - 1. **Positional Encoding**

Transformers are permutation-invariant, so positional encodings are added to patch embeddings to retain spatial information. These encodings help the model understand the relative positions of patches, enabling it to reason about structure and layout in the image.

* + - 1. **Transformer Encoder Blocks**

ViT uses a stack of transformer encoder layers, each consisting of:

* **Multi-head self-attention**: Enables the model to attend to all patches simultaneously and model long-range dependencies across the image.
* **Feedforward network**: A fully connected network applied to each token independently.
* **Layer normalization and residual connections**: Help stabilize training and preserve information flow.

In our setup, we used 12 encoder layers, each with 12 attention heads, a hidden size of 768, and an intermediate feedforward size of 3072.

* + - 1. **CLS Token for Classification**

A special learnable [CLS] token is prepended to the input sequence. After passing through all transformer layers, the final representation of the CLS token is used as the aggregated representation of the image. This token is passed to a classification head (linear layer) to predict the final class label.

##### 2.2.1.3 Vision-Based Recognition Model: TransVPR

The computer vision component of our AR Tour Guide system is handled by a custom hybrid model named **TransVPR** (Transformer-based Vision for Egyptian Landmark Recognition). This model combines the spatial sensitivity of convolutional neural networks with the global context modeling capabilities of transformers. It is specifically designed for robust classification of Egyptian monuments from real-world images.

* + - * 1. **Architecture Overview**

TransVPR abstract architecture

CLS token

7\*7 feature maps

224\*224

Tensor

Class probabilities

**TransVPR model**

**Softmax**

**fc layer**

**Vision Transformer**

**(ViT)**

**ResNet50**

(Without fc layer and classifier)

*TransVPR consists of three main stages:*

1. **Feature Extraction with ResNet50**  
   We use a ResNet50 model pretrained on our Augmented Egyptian Monuments dataset. Its classification head is removed and replaced with an identity layer to retain only the deep spatial features. The model was trained for 10 epochs and its weights were then frozen to serve as a static feature extractor in the final pipeline.
2. **Vision Transformer (ViT)**  
   The output features from ResNet50 (of shape [batch\_size, 2048]) are reshaped and passed into a custom-configured Vision Transformer. The ViT is not pretrained but initialized from scratch using ViTConfig. It processes the image features using 12 transformer blocks, each equipped with multi-head self-attention and feedforward layers.
3. **Classification Layer**  
   The [CLS] token representation from the ViT’s output is passed to a linear classification head, which maps it to one of 77 landmark classes.

## 2.3 Natural language processing

**Natural Language Processing (NLP) is a branch of Artificial Intelligence that helps computers understand and respond to human language. It allows our system to read, interpret, and generate answers in a way that feels natural to the user.**

**Why we use it:**

NLP allows the chatbot to:

* Understand the user’s questions
* Find the right answers about the statue
* Respond in a natural and easy-to-read way

Without NLP, the chatbot would not be able to **understand what the user is saying** or provide helpful answers.

## 2.4 Chatbot

A chatbot is a computer program that can simulate conversation with users. It uses **Natural Language Processing (NLP)** to understand questions and give smart answers—just like talking to a human tour guide. In our project, the chatbot is powered by the **Llama3 language model**, which helps it generate accurate, natural, and informative responses.

**Why we use it:**

The chatbot acts like a **virtual tour guide**. After a statue is identified using computer vision (CV), the chatbot gives a **brief description** and allows users to ask **more questions** about it, creating an interactive and educational experience.

Additionally, the chatbot can also be used **independently** so users can **freely ask about any monument or statue** they are curious about. This makes the experience flexible, informative, and engaging for all types of users.

### 2.4.1 Llama 3 Overview and architecture

**Llama 3: Large Language Model Meta AI 3**

**Llama** 3 is a state-of-the-art large language model (LLM) developed by Meta AI. It is designed to understand and generate human-like text with remarkable fluency and coherence, making it highly effective for tasks such as question answering, summarization, conversational AI, and more. Below is a deeper look into its core characteristics:

* **Llama 3's Training Methodology:** Llama 3 has been trained on a massive multilingual and multi-domain corpus. This enables it to learn statistical relationships between words, sentences, and concepts. As a result, it can generate fluent, coherent, and contextually relevant, and human-like responses across diverse subject areas.
* **Transformer Architecture:** Llama 3 is built upon a decoder-only Transformer architecture — an autoregressive model that processes tokens from left to right. Unlike RNNs, which handle tokens sequentially, transformers use self-attention to analyze all tokens in parallel, greatly enhancing long-range dependency modeling and training efficiency. because it’s decoder-only, Llama 3 applies a causal (triangular) attention mask, ensuring each token only attends to earlier tokens during both training and generation.

**Key Capabilities of Llama 3 for Chatbot Applications**

* **Contextual Understanding:** Llama 3 maintains context over long conversations, allowing it to generate responses that are accurate and coherent across multiple dialogue turns.
* **Natural Language Generation:** The model produces high-quality, fluent, and grammatically accurate responses, enhances the natural feel of chatbot interactions.
* **Adaptability:** Llama 3 integrates seamlessly with Retrieval-Augmented Generation (RAG) frameworks. This enables the chatbot to generate grounded answers based on external documents or knowledge bases, improving factual correctness.
* **Multilingual Proficiency:** Due to its diverse training data, Llama 3 is capable of understanding and generating content in multiple languages, which expands the usability of the chatbot for international or multicultural users.

By leveraging these capabilities, Llama 3 becomes an ideal foundation for intelligent chatbot systems that require rich language understanding, fluent generation, and integration with domain-specific knowledge.

### 2.4.1.1 A Decoder‑only Transformer Architecture

As Llama 3‑8B‑8192 is rooted in the decoder‑only Transformer, we will explain it now.

#### 2.4.1.1.1 Overview

A decoder‑only Transformer architecture for autoregressive modeling, with left‑to‑right causal masking, supporting 8,192‑token context via Rotary Positional Encodings and optimized inference via Grouped‑Query Attention (GQA) mechanism.

#### 2.4.1.1.2 Model Components

**1. Tokenizer & Embeddings**

* **Tokenizer**: Uses a **128 K subword vocabulary** for rich language representation and efficient tokenization across diverse languages.
* **Token Embeddings**: Learns embedding vectors of size *d* = 4,096. The embedding matrix is tied with the output projection matrix, a technique to reduce parameters and improve generalization (as also used in GPT and BERT)

**2. Positional Encoding**

* **Rotary Positional Encoding (RoPE)** with **θ = 500,000**, applied to Q/K projections across all layers, enabling strong extrapolation.
* Implements rotation-in-the-complex-plane on embeddings, preserving relative relationships and improving extrapolation beyond training lengths.

**3. Attention Mechanism**

**3.1 Scaled Dot‑Product Attention**

* Computes self-attention per:

Attention(Q,K,V)=softmax(​)V

using multi-head configuration for nuanced representations.

**3.2 Multi-Head Attention**

* Splits into **32 attention heads** (*d\_model* / *num\_heads* = 128-dim per head) .
* Uses residual pathways around each attention sub-layer followed by **RMSNorm** for efficiency and training stability

**3.3 Grouped Query Attention (GQA)**

* Optimizes inference by grouping query heads into **8 shared KV groups**, reducing memory/cache needs without sacrificing representational quality.
* Keys/values cached once per group for reuse across tokens during generation.

**3.4 Context Masking**

* Applies a **document-boundary-aware mask** during long-context pretraining to prevent tokens from attending across document segments .

**4. Feed-Forward Network (FFN)**

* Implements per-position fully connected layers of structure:

SwiGLU(W1x+b1)W2+b2

* Hidden dimension set to **6,144** (≈ 1.5 × 4,096) γ times wider for richer intermediate representations.
* **SwiGLU activation** chosen for performance gains over GeLU.

**5. Normalization & Residual Connections**

* Utilizes RMSNorm (Root Mean Square Layer Normalization); more parameter-efficient and stable than LayerNorm.
* Employs pre-norm architecture: normalization precedes residual addition for training smoothness.

**6. Decoder-Only Generator**

* Causal (left- to-right) self‑attention mask is applied to the attention weights to prevent access to future tokens during generation.
* 32 stacked transformer layers, each containing:

Masked GQA self-attention

Residual + RMSNorm

SwiGLU FFN

Residual + RMSNorm

* Final transformation outputs logits via a shared linear layer before softmax.

**7.KV Caching (Inference)**

• Per‑layer KV groups are cached once per group—no recomputation across tokens—leading to faster sequential generation.

**8. Hyperparameters**

| Component | Specification |
| --- | --- |
| Layers | 32 |
| Embedding / Model (§) | 4,096 |
| FFN hidden dim | 6,144 |
| Attention heads | 32 (8 KV groups via GQA) |
| Context length | 8,192 tokens |
| Vocabulary size | 128,000 |
| Positional encoding | RoPE, θ = 500k |
| Activation | SwiGLU |
| Normalization | RMSNorm |
| Masking | Document-boundary aware |

#### 2.4.1.1.3 Key Innovations

1. **Grouped‑Query Attention (GQA)**
   * Groups KV heads to 8 for memory‑efficient cache and 4× fewer KV tensors per layer during inference..
2. **RoPE with Extended θ**
   * Facilitates handling up to 8K tokens and supports better monotonicity and extrapolation.
3. **Document-boundary Masking**
   * Prevents tokens from attending across distinct documents during training to preserve contextual purity.
4. **SwiGLU + RMSNorm**
   * Improves performance and stability compared to standard GeLU + LayerNorm choices.

#### 2.4.1.1.4 Training

* **Pretraining Dataset**: Over **15 trillion publicly sourced tokens**, approximately seven times the size of Llama 2’s dataset. It also includes four times more code and >5% multilingual data across 30 + languages.
* **Objective**: Autoregressive next‑token prediction (causal language modeling), minimizing cross‑entropy between predicted and actual next tokens using teacher forcing.
* **Preprocessing**: Involves heuristic filters (e.g., NSFW), semantic deduplication, and quality classifiers to ensure dataset cleanliness.
* **Infrastructure**: Trained across two clusters of **~24,000 GPUs**, handling 8,192‑token sequence lengths and document-boundary masking for long-context efficiency.
* **Fine-Tuning & Alignment**: Instruction-tuned models integrate supervised fine-tuning (SFT), rejection sampling, proximal policy optimization (PPO), and direct policy optimization (DPO), using millions of human-curated samples to boost dialogue capability and safety.

### 2.4.2 Retrieval-Augmented Generation (RAG)

The chatbot is built on a **RAG architecture**, which combines:

* **Retriever**: Uses FAISS to find the top relevant chunks from a vectorized document database of Egyptian monument texts.
* **Generator**: Uses Llama 3 to synthesize answers based on retrieved content and user queries.

This allows the chatbot to return accurate, well-grounded responses to specific questions about monuments.

### 2.4.3 Embedding and Document Processing

The chatbot's knowledge base is built from curated documents about Egyptian artifacts. Steps included:

* Extracting and cleaning text from historical PDFs using PyMuPDF.
* Splitting the content into overlapping text chunks for better retrieval accuracy.
* Embedding the chunks with HuggingFace models.
* Storing them in a FAISS index for low-latency retrieval.

### 2.4.4 Integration of Groq Hardware Accelerator

To optimize inference latency and throughput, the Llama 3 model runs on Groq’s AI acceleration hardware. Groq's architecture offers:

* Ultra-low latency: Real-time response generation essential for smooth chatbot interactions.
* High throughput: Efficient handling of multiple concurrent queries.
* Energy efficiency: Reducing operational costs during inference.

Groq accelerates the transformer model’s forward pass, enabling the chatbot to maintain conversational fluency and context even under heavy usage.

### 2.4.5 Chatbot Flow and Session Management

The chatbot interaction follows a carefully designed flow to maintain context and provide coherent responses:

1. User Query Submission: The user inputs a question or request related to Egyptian monuments.
2. Session Tracking: Each user interaction is linked to a session ID that preserves conversational context across multiple turns. This allows the chatbot to remember prior exchanges and maintain coherence.
3. Document Retrieval: The retriever module queries the FAISS index to fetch the top relevant document chunks corresponding to the user’s question.
4. Response Generation: The Groq-accelerated Llama 3 model processes the retrieved context and user input to generate a precise, contextually aware response.
5. Response Delivery: The generated response is sent back to the user, maintaining the conversational flow.
6. Context Update: The chatbot updates the session state to include the latest user query and response, supporting continuous, multi-turn dialogue.

This flow ensures that the chatbot not only answers isolated queries but also engages in dynamic, context-rich conversations, enhancing user experience.

## 2.5 previous work

Numerous studies have investigated the integration of **Computer Vision** and **Augmented Reality (AR)** technologies to enrich **cultural heritage experiences**, particularly within the domains of **monuments and landmarks**.

### 2.5.1 Hassan et al. (2023)

presented a **comprehensive survey** on **monument recognition** using **artificial intelligence** techniques. The study addressed key challenges such as **scale variation**, **occlusion**, **rotation**, and **viewpoint shifts**. Several **Convolutional Neural Network (CNN)** architectures were evaluated, including **ResNet50 (96.57% accuracy)**, **InceptionV3 (94.28%)**, **MobileNet (92.85%)**, and **DenseNet (95.71%)**. Among these, **ResNet50** and **DenseNet** emerged as the **most reliable** across datasets featuring **Indian, Turkish, and Egyptian landmarks**. The research also underscored the critical role of **data augmentation** and **preprocessing** in improving **model generalization** for **monument classification**.

### 2.5.2 Chowanda and Sutoyo (2019)

developed a **deep learning framework** for the **classification of Indonesian landmarks** using CNNs. Their study introduced a **dataset of over 16,000 augmented images** sourced from the internet, covering six major Indonesian islands: **Bali, Java, Kalimantan, Papua, Sulawesi**, and **Sumatra**. They evaluated three CNN architectures:

* **VGG-16** (average accuracy: **68.6%**)
* **VGG-19** (**77.5%**)
* **GoogleNet** (**79.2%**)

While **GoogleNet** demonstrated the **most consistent performance** across all classes, **VGG-19** achieved the **highest individual class accuracy** (**92% on Kalimantan**). The study highlighted challenges in **distinguishing similar architectural features**, particularly those of **traditional houses** and **cultural landmarks** across different islands.

### 2.5.3. Teixeira et al. (2021)

introduced the ***Ecomuseu Virtual Guide***, a **mobile AR application** designed to enhance **museum visits** through **gamified interaction** and **digital overlays**. Developed using **Unity** and **EasyAR**, the app featured **3D models**, **interactive missions**, and a **marker-based AR system**. Visitors engaged with **digital mascots** and received **contextual historical information** via **text, images, and audio**. Despite being tested virtually due to **pandemic restrictions**, feedback from 15 participants indicated strong **educational value** and **user engagement**. The application also showed promise in **promoting repeat visits** and **deepening interest** in **museum exhibits**.

Collectively, these studies underscore the potential of combining **deep learning** for **object recognition** with **AR interfaces** to create **immersive**, **context-aware**, and **educational cultural heritage experiences**.

## 2.6 Overview

The **AR Tour Guide system** is built using a modern **web technology stack** that includes **Angular** on the **frontend** and **Spring Boot** on the **backend**. These frameworks are supported by **AI components** such as **AR.js** for **augmented reality**, **OpenCV** for **computer vision**, and **LLaMA** for **natural language understanding**.

**Frontend: Angular**

**Angular** is a **TypeScript-based framework** developed by **Google** for building **web applications**. It provides a **structured** and **scalable architecture** for creating **dynamic user interfaces**. In this project, **Angular** is responsible for **rendering the AR interface**, **managing user interactions**, and **connecting with backend APIs**. Its support for **component-based development** and **responsive design** makes it suitable for **cross-device compatibility**.

**Backend: Spring Boot**

**Spring Boot** is a **Java-based framework** used for building **standalone**, **production-grade backend services**. It simplifies the development of **RESTful APIs** and provides integrated **security features** through **Spring Security**. In the **AR Tour Guide system**, **Spring Boot** handles **user authentication**, **reservation management**, **content delivery**, and **real-time API responses**. **JWT (JSON Web Tokens)** ensures **secure**, **stateless user sessions**, while **Spring Data JPA** manages **database operations**.

The **backend** also integrates with **AR** and **NLP modules**, coordinating **object recognition** and **information retrieval tasks**. This **modular architecture** supports **scalability** and allows the system to evolve with additional features like **multilingual support** or **expanded domain coverage**.

# **Chapter 3: System analysis and design**

## 3.1 User Requirements

The following requirements reflect what users expect from the AR Tour Guide system.

**1. General/User Account Management**

* Users must be able to create accounts, log in, and manage their profiles.
* Different roles should exist:
  + **Admin**: Full access to manage content, users, and system configurations.
  + **Regular Users**: Access to explore AR features, make bookings, interact with data, and ask questions.

**2. Augmented Reality (AR) Features**

* Users must be able to display real-world objects (e.g., monuments, exhibits) through the device camera and receive relevant and interactive explanations.
* The AR camera must provide real-time feedback for recognized objects.
* Users should be able to scan objects from multiple angles, even with varying lighting conditions.

**3. Knowledge Retrieval and Question Answering**

* Users should get accurate and engaging information about recognized objects using **Retrieval-Augmented Generation (RAG)**.
* Users must be able to ask questions and receive intelligent, human-like answers via a chatbot, even without visual inputs.

**4. Accessibility and Educational Tools**

* Visually impaired users should have the option for audio descriptions of objects detected by the system.
* The system must highlight features beneficial in educational settings, such as step-by-step learning for scanned objects.

**5. Web-Based Accessibility**

* The application should work seamlessly on standard web browsers and not require native apps to run AR functionalities.
* Users should have access across devices like desktops, tablets, and smartphones.

**6. Reservation and Booking Features**

* Users must be able to reserve a time slot for museum tours or visit specific landmarks.
* Booking details should be emailed or stored in their user account for easy reference.
* Reservations should be modifiable or cancelable by the user.

**8. Notifications**

* Users must receive email notifications for account activity (e.g., login), reservations, or upcoming tours.

**9. Scalability**

* Users should be able to explore objects from different domains (e.g., tourism, retail, education) as more data becomes available.

## 3.2 Functional Requirements

**User Management**

|  |  |
| --- | --- |
| Functional Requirement | User Management |
| Action | Manage users (add, view, update, delete). |
| Input | - Add/Update: (username, password, email). - Delete/Retrieve: `id` or none (for all users). `AuthRequest` |
| Pre-condition | - Admin is authenticated and authorized. - Users exist for retrieval or modification. |
| Post-condition | - Users are added, updated, deleted, or retrieved in the database. |
| Output | - `UserEntity` for add, update, retrieve. - Empty response (`Void`) for deletions. |

**Tourism Package Management**

|  |  |
| --- | --- |
| Functional Requirement | Tourism Package Management |
| Action | Manage museums, moments, or other tourism packages (e.g., add, update, view, delete). |
| Input | - Add/Update: Package details (name, description, location, ticket price, multimedia, etc.). - Delete/View: `id` or none (for all). |
| Pre-condition | - Admin is authenticated and authorized. - Valid package data for add/update. - Package exists for delete/view. |
| Post-condition | - Packages are added, updated, deleted, or retrieved in the database. |
| Output | - Package details as JSON for retrieval. - Empty response (`Void`) for deletions. |

**Booking System**

|  |  |
| --- | --- |
| Functional Requirement | Booking System |
| Action | Make, view, or cancel reservations. |
| Input | - Make: User `id`, package `id`, number of tickets. - Cancel/View: Reservation `id`. |
| Pre-condition | - User is authenticated. - Package availability is verified for requested date and tickets. - Reservation exists to modify. |
| Post-condition | - Reservations are created, updated (on cancel), or retrieved. - Ticket inventory is adjusted. |
| Output | - Reservation details (confirmation or cancellation). |

**Database Persistence**

|  |  |
| --- | --- |
| Functional Requirement | Database Persistence |
| Action | Store and retrieve data (e.g., users, reservations, packages). |
| Input | Data objects (e.g., `UserEntity`, `Reservation`, `Museum`) |
| Pre-condition | - Database connection is active. - Proper JPA mapping for entities. |
| Post-condition | - Data is saved or retrieved from the database. |
| Output | - JSON data or success acknowledgment (`200 OK`). |

**Mail System**

|  |  |
| --- | --- |
| Functional Requirement | Mail System |
| Action | Send email notifications (e.g., confirmations, reminders). |
| Input | - Recipient email address. - Email subject and body (e.g., reservation details). |
| Pre-condition | - SMTP configuration is properly set up. - Recipient email exists and is valid. |
| Post-condition | - Email is delivered (or logged if failed). |
| Output | - Confirmation logs of success or errors. |

**Template Rendering with Thymeleaf**

|  |  |
| --- | --- |
| Functional Requirement | Template Rendering with Thymeleaf |
| Action | Render web pages for the user and admin interface. |
| Input | Thymeleaf templates with dynamic context data (e.g., logged-in user, reservations). |
| Pre-condition | - Thymeleaf is configured properly. - Templates exist and are valid. |
| Post-condition | - HTML pages are dynamically rendered with context data. |
| Output | - Responsive HTML pages displayed to users. |

**Error Logging and Debugging**

|  |  |
| --- | --- |
| Functional Requirement | Error Logging and Debugging |
| Action | Log application behavior and errors. |
| Input | Error or operational context (captured automatically). |
| Pre-condition | - Logging levels for relevant packages are configured (e.g., `Spring`, `Thymeleaf`, `Mail`). |
| Post-condition | - Logs are created with error details or success status. |
| Output | - Log files or console logs. |

## 3.3 Non-functional Requirements

Non-functional requirements outline the operational attributes, focusing on how functionality is delivered:

1. Performance:

- Optimize response times for search and booking functionalities to under 3 seconds under normal load.

- Ensure database queries are efficient by leveraging Hibernate’s MySQL8Dialect.

3. Security:

- Encrypt sensitive data managed by the system, such as email credentials and database passwords.

- Implement input validation to protect against SQL Injection and Cross-Site Scripting (XSS).

4. Reliability:

- Ensure application reliability by enabling automatic recovery and logging when errors occur.

5. Maintainability:

- Use a modular codebase with clearly defined responsibilities for each feature (e.g., a separate module for mailing).

6. Usability:

- Ensure the web interface is intuitive and mobile-friendly.

### **Class diagram**

A diagram of a computer code

AI-generated content may be incorrect.

**Figure 2: Class diagram**

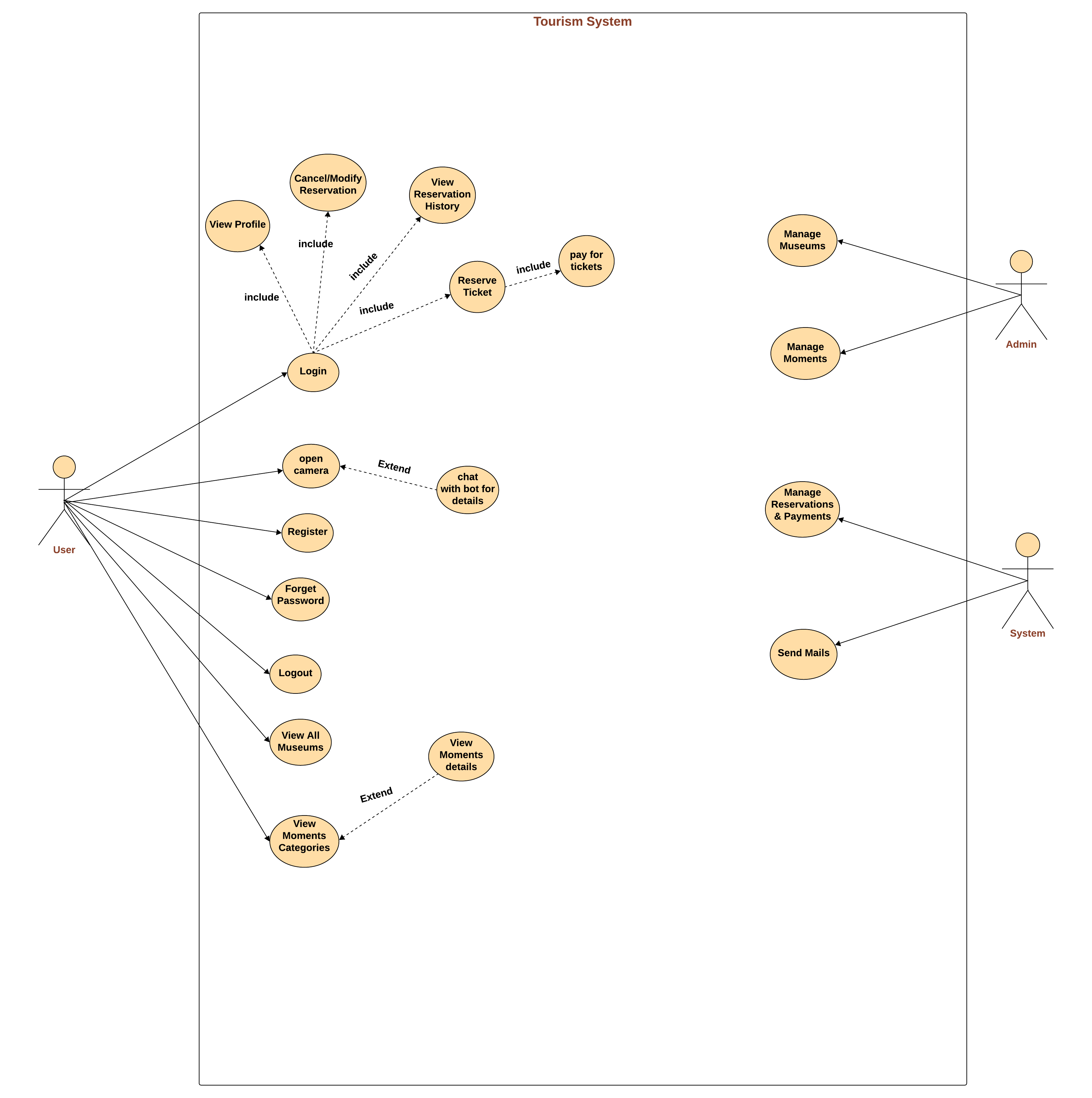
### **ERD diagram**

A computer screen shot of a computer

AI-generated content may be incorrect.

**Figure 3: ERD diagram**

### use case diagram



**Figure 4: use case diagram**

### activity diagram

**login activity diagram**

A diagram of a computer program

AI-generated content may be incorrect.

**Figure 4: Login activity diagram**

**Register activity diagram**

A diagram of a computer program

AI-generated content may be incorrect.

**Figure 5: Register activity diagram**

**Reservation activity diagram**

A diagram of a process

AI-generated content may be incorrect.

**Figure 6: Reservation activity diagram**

**Select museum activity diagram**

A diagram of a museum

AI-generated content may be incorrect.

**Figure 7: Select museum activity diagram**

### **Sequence diagram**

**Admin sequence diagram**

A diagram of a process flow

AI-generated content may be incorrect.

**Figure 8: Admin sequence diagram**

**Login sequence diagram**

A diagram of a login

AI-generated content may be incorrect.

**Figure 9: Login sequence diagram**

**Register sequence diagram**

A diagram of a workflow

AI-generated content may be incorrect.

**Figure 10: Register sequence diagram**

**Reservation sequence diagram**

A screenshot of a computer screen

AI-generated content may be incorrect.

**Figure 11: Reservation sequence diagram**

**Select museum sequence diagram**

A diagram of a structure

AI-generated content may be incorrect.

**Figure 12: Select museum sequence diagram**

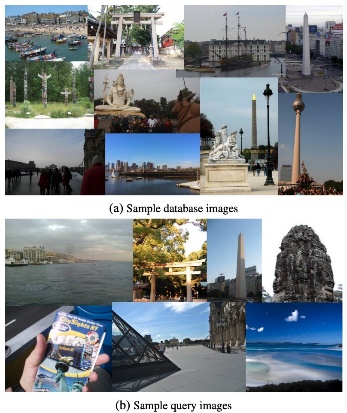
# **Chapter 4: implementation**

## 4.1 Data preprocessing

### 4.1.1 Image Data

#### 4.1.1.1 Datasets Used

To train the object recognition component of the AR Tour Guide system, we utilized a combination of large-scale public datasets, curated collections, and manually scraped data:

* **Google Landmarks Dataset (v2)**  
  A comprehensive dataset with over 5 million images representing thousands of landmarks worldwide. It served as a foundational dataset for recognizing a wide variety of architectural and cultural sites, offering diverse, real-world image data for large-scale classification tasks.
* A logo with a person in a circle

  AI-generated content may be incorrect.A blue text on a black background

  AI-generated content may be incorrect.**Kaggle and Roboflow Datasets**  
  We incorporated 18 smaller datasets from Kaggle and Roboflow platforms, each focused on Egyptian landmarks and heritage structures. These datasets were essential for fine-tuning the model to recognize Egypt-specific cultural monuments with greater accuracy.
* **Web Scraping from Wikimedia and Search Engines**  
  To supplement underrepresented classes, we implemented web scraping techniques to collect additional images from sources like Wikimedia Commons and browser-based search engines (e.g., Google Images). The collected images were manually filtered and verified to ensure relevance and accuracy before being added to the training data.

#### 4.1.1.2 Challenges in Data Preparation

Constructing a reliable and effective dataset involved overcoming several key challenges:

1. **Misclassification and Label Noise**  
   Many images, especially from community-contributed datasets, contained inaccurate or ambiguous labels. We addressed this by performing manual filtering and verification, significantly improving label quality and consistency.
2. **Duplicates and Redundancy**  
   Merging multiple datasets led to a high volume of duplicated content. A deduplication process was applied, along with consistent re-labeling, to maintain data integrity and avoid skewed training.
3. **Class Imbalance**  
   After cleaning, the dataset exhibited severe class imbalance—some monuments were overrepresented, while others had few samples. To correct this, we applied targeted data augmentation strategies to increase minority class sizes and promote model fairness.
4. **Missing Data for Specific Monuments**  
   Certain monuments had little to no representation in existing datasets. We compensated by scraping additional image data from Wikimedia and browser sources, ensuring even the rare classes were represented in the final training set.
5. **Data Augmentation Techniques**  
   To enhance generalization and prepare the model for real-world conditions, we applied a series of augmentation methods:
   * **Brightness adjustments**
   * **Perspective transformations**
   * **Blurring**
   * **Noise injection**

These techniques improved robustness to different lighting, angles, and camera qualities encountered in live AR use cases.

1. **Final Dataset: Augmented Egyptian Monuments**  
   The outcome of this process is our custom dataset—[**Augmented Egyptian Monuments**](https://www.kaggle.com/datasets/marwa0mar/augmented-egyptian-monuments)—comprising high-quality, domain-specific images. On average, each class contains approximately **3,270 images**, making it a balanced, scalable dataset tailored for training the AR Tour Guide system.

### 4.1.2 Text Data:

The chatbot component of the AR Tour Guide system is responsible for providing users with historical, cultural, and contextual information about Egyptian monuments once a monument has been detected and identified. To ensure the chatbot delivers accurate and detailed responses, we implemented a multi-step data preparation pipeline focusing on textual data from verified historical sources.

#### **4.1.2.1 Document Collection and Curation**

We collected high-quality textual descriptions, historical references, and scholarly summaries of Egyptian statues and monuments from reliable sources, including:

* Digitized museum archives
* Government cultural websites
* Academic publications in PDF format
* Official guides published by the Ministry of Tourism and Antiquities

Each document was manually verified to ensure historical accuracy and relevance to the monuments covered by our AR recognition model.

#### **4.1.2.2 PDF Parsing and Chunking**

The collected documents were parsed using LangChain's PyPDFLoader, which enables efficient loading and segmentation of multi-page PDFs. To preserve semantic structure and improve retrieval accuracy, we applied the following chunking strategy:

* **Flattening Nested Documents:** Pages from multiple documents were flattened into a continuous sequence of text blocks.
* **Chunk Size:** Text was divided into segments of 512 characters with an overlap of 64 characters to maintain contextual flow across chunks.
* **Metadata Attachment:** Each chunk retained metadata (document title, source, and page number) to ensure traceability and contextual anchoring during response generation.

#### **4.1.2.3 Embedding and Vector Store Creation**

To support fast and semantically relevant retrieval, we employed **HuggingFace Embeddings** to convert textual chunks into dense vector representations. The embedding model used was:

* all-MiniLM-L6-v2 (a sentence transformer from Hugging Face), chosen for its balance between performance and computational efficiency.

These vectors were stored using **FAISS (Facebook AI Similarity Search)**, an optimized vector store for efficient nearest-neighbor searches during retrieval-augmented generation (RAG).

## 4.2 Implementation details for Augmented Reality Tour Guide

### 4.2.1 Computer Vision

#### A logo of a company AI-generated content may be incorrect.A logo with orange and grey circles AI-generated content may be incorrect.4.2.1.1 Used tools and python packages

##### 4.2.1.1.1 Used Tools

* A blue text on a black background

  AI-generated content may be incorrect.**Google Colab & Kaggle Notebooks**:

Used for model training and experimentation.

* **Jupyter Notebooks**:

Local development and testing.

##### 4.2.1.1.2 Python packages

* **PyTorch:** Core framework for building, training, and evaluating deep learning models.
* **Transformers:** Vision Transformer (ViT) model used for visual recognition; sourced from Google’s official implementation, offering strong performance on landmark classification tasks.
* **OpenCV & PIL (Pillow):** Used extensively for image manipulation, preprocessing, augmentation, and format conversion.
* **Torchvision:** For pre-trained CNN models, image transforms, and dataset loaders.
* **scikit-learn:** Utilized for data preprocessing, model evaluation, and metrics computation.
  + **sklearn.metrics**: Specifically used to evaluate model performance (e.g., accuracy, precision, recall, F1-score) for classification tasks.
* **NumPy & Python:** For handling structured data and performing numerical computations and CSV data resulted from web scraping reading.
* **Matplotlib & Seaborn:** Used for visualizing dataset distributions, model training progress, and performance metrics.
* **os & time:** For file handling and tracking training time.
* **python-docx**: Used to generate and manipulate Microsoft Word documents for result reporting and documentation.
* **json**: Used to load and parse annotation files (e.g., bounding boxes, labels) in dataset preparation.
* **Roboflow API**: Used for importing and extending datasets from Roboflow during preprocessing.
* **Kagglehub:** Used for importing and extending datasets from Kaggle during preprocessing.
* **re**: Used to match, search, and replace patterns during dataset cleaning, particularly for renaming inconsistent class labels and filenames.
* **Shutil:** for folder and file manipulation through the current environment.
* **Hashlib:** for efficient duplicated images detection.
* **Flask:** Used as the backend framework to create APIs and manage communication between the frontend interface and the system components.

**4.2.1.2 Training Details**

**4.2.1.2.1 *ResNet50 Training Configuration***

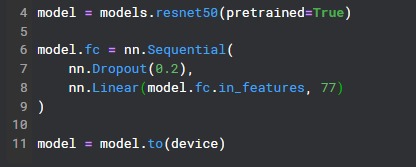
* **Input Preprocessing**:  
  During inference and training, input images are resized to a fixed dimension, normalized, and converted to tensors.

A computer screen shot of a program

AI-generated content may be incorrect.

During the initial training of ResNet50 on the Augmented Egyptian Monuments dataset, the following configuration was used:

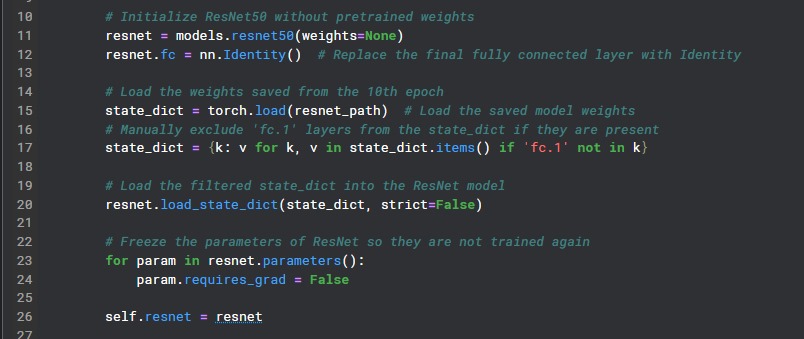
* **Epochs**: 10
* **Optimizer**: Adam optimizer with a learning rate of **0.001**
* **Loss Function**: CrossEntropyLoss.

A screenshot of a computer program

AI-generated content may be incorrect.

**4.2.1.2.2 *TransVPR Training Configuration***

* Only the Vision Transformer and final classifier layers were trained; ResNet50 was kept frozen.

****

* The ViT is configured to take 7×7 feature maps from ResNet-50 (with 2048 channels) using a patch size of 1. It uses 12 transformer layers, 12 attention heads, a hidden size of 768, and outputs predictions for the target number of classes.

**A computer screen with text and numbers

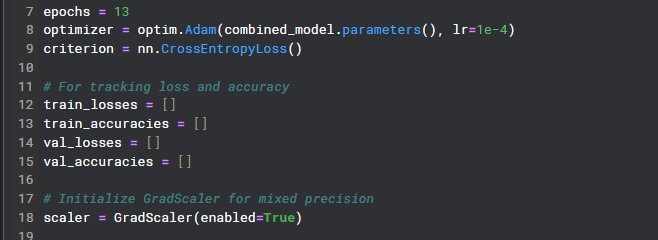
AI-generated content may be incorrect.**

* The features get reshaped to match the input size expected by ViT (7x7 patches)
* The final classification is done based on the [CLS] token from the transformer output.

A screen shot of a computer

AI-generated content may be incorrect.

* **Epochs:** 13
* **Optimizer:** Adam optimizer with a learning rate of **1e-5**.
* **Loss function:** Cross-entropy loss.
* **Mixed Precision Scaling**: A GradScaler was initialized with enabled=True to allow automatic mixed precision training on supported GPUs, enabling faster computation and reduced memory usage.

****

* **Target:** Accurate classification of 77 Egyptian landmarks.

### 4.2.2 Chatbot System

#### 4.2.2.1 Used Tools and Python Packages

##### 4.2.2.1.1 Used Tools

* **Visual Studio Code (VS Code)**: Used exclusively for writing, debugging, and testing the entire chatbot system, including document embedding, RAG pipeline integration, and the user interface with Streamlit.

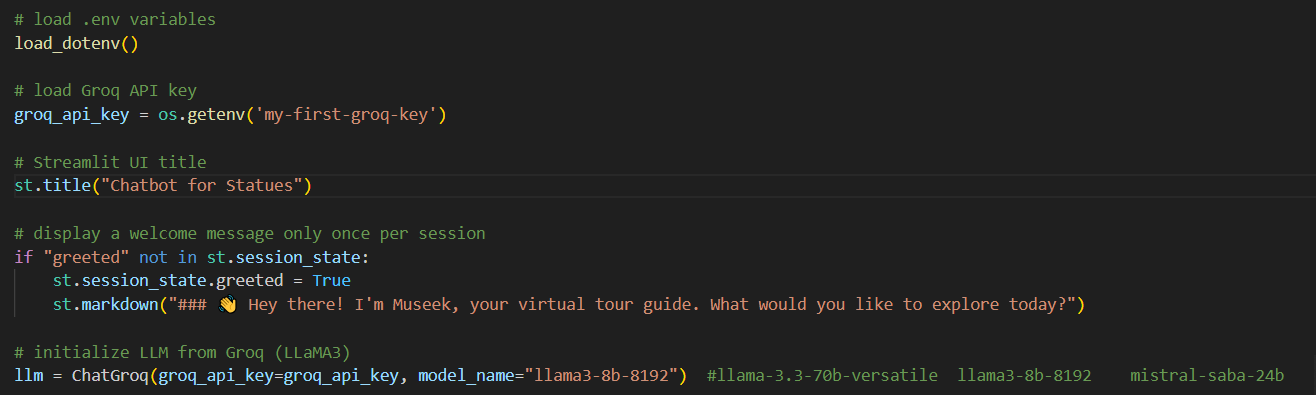
##### 4.2.2.1.2 Python Packages

* **Streamlit**: Used to build an interactive web interface for the chatbot. It handles user input, displays responses, and renders a clickable microphone icon for text-to-speech playback.
* **langchain** and **langchain\_community/langchain\_core**: Used to build the Retrieval-Augmented Generation (RAG) pipeline. LangChain manages prompt templates, document chains, retrieval mechanisms, and integration with the Llama 3 model.
* **langchain\_groq**: Connects to the Llama 3 model hosted via the Groq API, enabling high-speed inference and accurate response generation.
* **HuggingFace Embeddings**: The sentence-transformers/all-MiniLM-L6-v2 model is used to embed textual data (from PDFs) into vector representations for similarity search.
* **FAISS (Facebook AI Similarity Search)**: Used as the vector database to store and retrieve relevant document embeddings efficiently.
* **PyPDFDirectoryLoader** (LangChain community): Loads and parses multiple PDF documents from the Books directory for use as the chatbot's knowledge base.
* **RecursiveCharacterTextSplitter**: Splits long text documents into overlapping chunks (2,000 characters with 100-character overlap) to enhance retrieval accuracy.
* **dotenv**: Used to securely manage and load the Groq API key from an .env file.
* **gTTS (Google Text-to-Speech)**: Converts the chatbot's text responses into speech, which is then playable using a custom audio player in the UI.
* **base64, uuid, io, pathlib**: Used to create and render a custom audio playback button using a local microphone icon image.

**4.2.2.2 Implementations details**

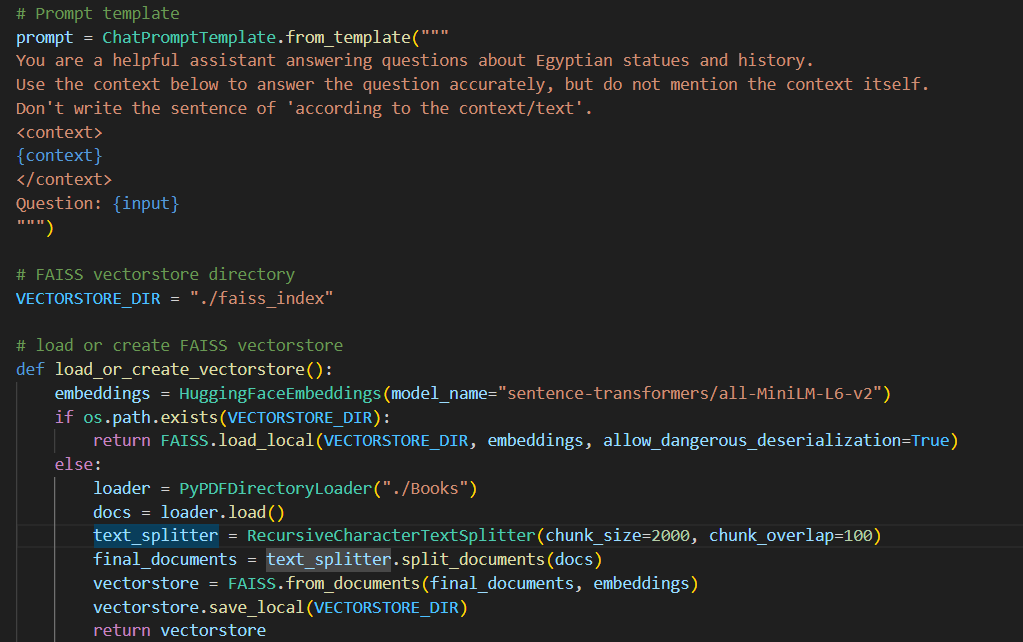
**Purpose:** Initializes the chatbot interface and loads required settings.

* **Environment Setup:** Loads API keys and environment variables securely using .env files.
* **Title Display:** Sets the Streamlit app title as **"Chatbot for Statues"**.
* **User Greeting:** Displays a friendly welcome message like “Hey there! I’m Museek...” — but only once per user session.
* **LLM Initialization:** Connects to Groq’s **LLaMA3 language model** to power the chatbot’s responses.



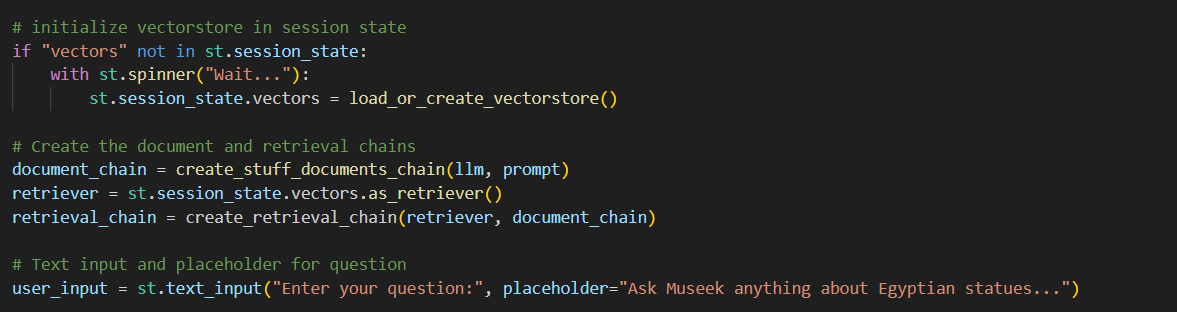
**Purpose:** Defines how the chatbot understands questions and finds relevant information.

* **Prompt Template:** Provides instructions to the chatbot, telling it how to answer questions clearly.
* **FAISS Directory:** Sets up a path to store or retrieve a **vector database** used for finding relevant documents.
* **Vectorstore Loader:** Loads existing data or processes PDF books by:
  + **Splitting documents** into chunks.
  + **Embedding** them into vectors.
  + **Storing** them using **FAISS** for fast retrieval later.



**Purpose:** Prepares document search and enables user input.

* **Session Check:** Ensures vectorstore is loaded only once per session with a loading spinner.
* **Retrieval Chain Setup:** Connects the document search logic with the chatbot’s answering logic.



# **Chapter 5: results and testing**

## 5.1 Evaluation metrics in Computer Vision models

### ResNet50 Standalone Evaluation

#### 5.1.1.1 Training and Validation Curves

**Model Performance Summary (10 Epochs)**

The learning curves indicate a stable and effective training process across 10 epochs.

**Training Loss:**  
The model’s training loss steadily decreased from **1.100 to 0.070**, reflecting successful learning and a consistent reduction in error on the training set.

**Validation Loss:**  
Validation loss declined from **0.650 to 0.120**. Although there was a slight increase between epochs 2 and (rising from **0.330 to 0.350**), the overall trend was downward. This suggests good generalization with no evidence of overfitting.

**Training Accuracy:**  
Training accuracy improved progressively from **69.0% to 98.5%**, indicating that the model effectively learned from the training data throughout the epochs.

**Validation Accuracy:**  
Validation accuracy increased from **80.5% to 96.5%**, closely tracking the training accuracy. The small gap between training and validation performance confirms that the model generalizes well to unseen data.

**Overall Assessment:**  
The training process converges smoothly, with consistent performance gains up to epoch 10. While the rate of improvement slows in later epochs, no overfitting is observed. The model continues to improve across both training and validation metrics, reflecting a successful and effective training process.

Figure : Accuracy-Loss curve with ResNet50 model

A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

AI-generated content may be incorrect.

#### 5.1.1.2 Confusion Matrix

* This **2×2 averaged confusion matrix** summarizes the overall classification behavior of our model across all 77 classes by computing the average **True Positives (TP)**, **True Negatives (TN)**, **False Positives (FP)**, and **False Negatives (FN)**.
* The values represent the **mean contribution of each class** to the total confusion matrix, providing a more interpretable high-level view of model performance in multi-class classification.
* The matrix indicates that, on average:
  + **TP (correctly predicted target class)** = **0.41**
  + **TN (correctly predicted non-target class)** = **25476.66**
  + **FP (incorrectly predicted as target class)** = **12.56**
  + **FN (missed target class)** = **326.38**

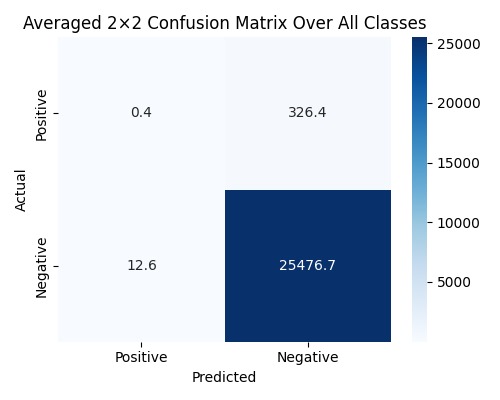


Figure 14: Confusion matrix with ResNet50

#### 5.1.1.2 Classification Report

**Individual Class Performance**

**Test accuracy:** 96.0%

**Test loss:** 0.1154

**Total test samples:** 25,159

**Total number of classes:** 77

**Average Precision: 96.44%**

**Average Recall: 96.57%**

**Average F1-score:** 97%

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | Precision | Recall | F1-Score | Support |
| Abu Simbel Temples | 0.95 | 0.97 | 0.96 | 326 |
| Ahmose I | 0.99 | 1.0 | 0.99 | 299 |
| Akhenaten | 0.98 | 0.98 | 0.98 | 333 |
| Al-Azhar Mosque | 0.99 | 0.84 | 0.91 | 330 |
| Al-Deir al-Bahary Temple of Queen Hatshepsut | 0.91 | 0.93 | 0.92 | 341 |
| Alexandria Library | 0.88 | 0.99 | 0.93 | 339 |
| Amenhotep III and Tiye | 0.98 | 0.98 | 0.98 | 318 |
| Amr Ibn al-Aas Mosque | 1.0 | 0.99 | 1.0 | 323 |
| Bab Zuwayla | 1.0 | 1.0 | 1.0 | 313 |
| Bab al-Nasr | 0.98 | 0.98 | 0.98 | 308 |
| Babylon Fortress | 0.99 | 1.0 | 1.0 | 331 |
| Bagawat | 1.0 | 0.98 | 0.99 | 345 |
| Baron Empain Palace | 0.98 | 1.0 | 0.99 | 324 |
| Bayt al-Suhaymi | 0.99 | 1.0 | 1.0 | 363 |
| Ben Ezra Synagogue | 0.98 | 0.99 | 0.99 | 316 |
| Bent Pyramid for Senefru | 1.0 | 0.96 | 0.98 | 317 |
| Cairo Citadel | 0.96 | 0.96 | 0.96 | 352 |
| Cairo Tower | 0.99 | 0.98 | 0.98 | 304 |
| Cavern Church Abu Serga | 0.99 | 1.0 | 0.99 | 335 |
| Cleopatra VII | 0.99 | 0.99 | 0.99 | 335 |
| Colossoi of Memnon | 0.98 | 0.91 | 0.94 | 371 |
| Deir el-Medina | 0.98 | 1.0 | 0.99 | 330 |
| Dendera Temple Complex | 0.89 | 0.87 | 0.88 | 306 |
| Djoser | 0.83 | 0.89 | 0.86 | 324 |
| Edfu Temple | 0.85 | 0.85 | 0.85 | 348 |
| Egyptian Museum, Cairo | 0.99 | 0.99 | 0.99 | 337 |
| Fatimid Cemetery in Aswan | 1.0 | 0.98 | 0.99 | 352 |
| Gayer Anderson Museum | 0.97 | 0.99 | 0.98 | 336 |
| Gebel el-Silsila | 0.99 | 1.0 | 1.0 | 309 |
| Giza Pyramid Complex | 0.97 | 0.96 | 0.96 | 342 |
| Goddess Isis with Her Child | 1.0 | 0.99 | 1.0 | 331 |
| Golden Mask of Tutankhamun | 0.97 | 0.99 | 0.98 | 313 |
| Golden Throne of Tutankhamun | 1.0 | 1.0 | 1.0 | 329 |
| Great Hypostyle Hall of Karnak | 0.98 | 0.99 | 0.99 | 298 |
| Green Head | 0.97 | 0.98 | 0.98 | 325 |
| Hanging Church (St. Virgin Mary Coptic Orthodox Church) | 0.96 | 0.98 | 0.97 | 355 |
| Horemheb | 0.99 | 0.95 | 0.97 | 336 |
| Ibn Tulun Mosque | 0.93 | 1.0 | 0.96 | 311 |
| Karnak Temple | 0.94 | 0.92 | 0.93 | 323 |
| Khafre | 0.99 | 0.93 | 0.96 | 335 |
| Khufu Statue | 0.98 | 0.99 | 0.98 | 324 |
| King Thutmose III | 0.99 | 0.99 | 0.99 | 324 |
| Kiosk of Trajan in Philae | 0.98 | 0.98 | 0.98 | 335 |
| Luxor Temple | 0.98 | 0.98 | 0.98 | 293 |
| Mausoleum of Aga Khan | 0.99 | 1.0 | 1.0 | 314 |
| Mortuary Temple of Amenhotep III | 0.9 | 0.99 | 0.94 | 329 |
| Mummy of Ramsis II | 1.0 | 0.99 | 0.99 | 337 |
| Narmer (Menes) | 0.99 | 1.0 | 0.99 | 346 |
| Narmer Palette | 0.98 | 1.0 | 0.99 | 321 |
| Nefertiti | 0.99 | 0.94 | 0.97 | 317 |
| Pompeys Pillar Alexandria | 1.0 | 1.0 | 1.0 | 344 |
| Ptolemaic Temple of Hathor in Deir el-Medina | 0.98 | 1.0 | 0.99 | 333 |
| Pyramid of Djoser | 1.0 | 0.98 | 0.99 | 315 |
| Pyramid of Unas | 0.97 | 0.99 | 0.98 | 335 |
| Qaitbay Citadel | 0.98 | 0.99 | 0.98 | 336 |
| Queen Hatshepsut | 0.94 | 0.89 | 0.92 | 307 |
| Ramesseum | 0.95 | 0.86 | 0.9 | 333 |
| Ramsis II | 0.95 | 0.88 | 0.92 | 321 |
| Ramsis II Red Granite Statue | 0.84 | 1.0 | 0.91 | 308 |
| Red Pyramid | 0.95 | 1.0 | 0.97 | 342 |
| Serapeum of Saqqara | 0.98 | 0.99 | 0.99 | 334 |
| Sesostris III | 0.95 | 0.99 | 0.97 | 345 |
| Sobekneferu | 0.99 | 0.96 | 0.98 | 289 |
| Sphinx | 0.97 | 0.96 | 0.96 | 314 |
| St. Catherine Monastery Mount Sinai | 0.96 | 0.96 | 0.96 | 353 |
| St. George Church in Coptic Cairo | 0.99 | 1.0 | 1.0 | 310 |
| Statue of Tutankhamun with Ankhesenamun | 0.99 | 0.99 | 0.99 | 344 |
| Temple of Habu | 0.98 | 0.96 | 0.97 | 344 |
| Temple of Hibis | 1.0 | 0.99 | 1.0 | 333 |
| Temple of Horus at Edfu | 0.81 | 0.81 | 0.81 | 314 |
| Temple of Isis in Philae | 0.96 | 0.85 | 0.9 | 330 |
| Temple of Khonsu in Karnak | 1.0 | 0.98 | 0.99 | 336 |
| Temple of Kom Ombo | 0.89 | 0.98 | 0.94 | 311 |
| Temple of Seti I at Abydos | 0.98 | 0.95 | 0.96 | 330 |
| Temple of the Oracle of Amun at Siwa | 0.99 | 0.97 | 0.98 | 312 |
| The Solar Boat of Khufu | 1.0 | 1.0 | 1.0 | 300 |
| The Statue of Sekhmet | 0.99 | 0.98 | 0.98 | 283 |

### 

### 5.2.1 TransVPR (ResNet50 + ViT) Evaluation

#### 5.2.1.1 Training and Validation Curves

**Model Performance Summary (13 Epochs)**

The learning curves indicate a stable and effective training process across 13 epochs.

* **Training Loss:**  
  The model’s training loss consistently decreased from **0.155 to 0.038**, demonstrating that it effectively minimized error on the training set over time.
* **Validation Loss:**  
  Validation loss showed a steady decline from **0.093 to 0.030**. Although a slight fluctuation occurred around epoch 8 (rising from **0.045 to 0.056**), the overall trend remained downward. This pattern reflects good generalization and no signs of overfitting.
* **Training Accuracy:**  
  Accuracy on the training set improved steadily from **95.5% to 98.7%**, indicating consistent learning progress.
* **Validation Accuracy:**  
  Validation accuracy increased from **96.9% to 98.9%**, closely mirroring the training accuracy. The narrow gap between the two confirms the model’s strong generalization capabilities.
* **Overall Assessment:**  
  The convergence is smooth, and performance gains continue through epoch 13, albeit with diminishing returns. While early stopping could be considered for efficiency, there is no indication of overfitting, and both loss and accuracy metrics reflect a highly effective training process.

**A graph of different colored lines

AI-generated content may be incorrect.**

Figure 15: Accuracy-Loss curve with TransVPR model

#### 5.2.1.2 Confusion Matrix

With awareness of global features, TransVPR shows improvement in discrimination in the True Negative value

The matrix indicates that, on average:

* + **TP (correctly predicted target class)** = **0.027**
  + **TN (correctly predicted non-target class)** = **25773.178**
  + **FP (incorrectly predicted as target class)** = **21.398**
  + **FN (missed target class)** = **21.398**

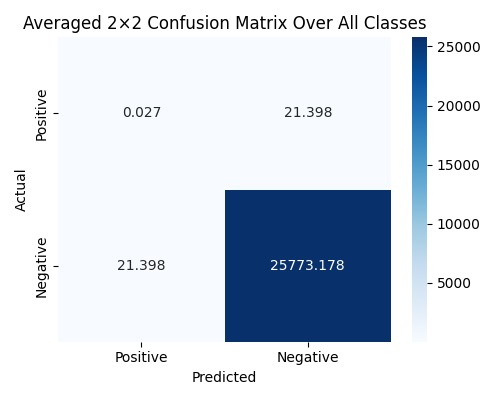


Figure 16: Confusion matrix with TransVPR

#### 5.2.1.3 Classification Report

**Test Loss:** 0.0316

**Test Accuracy:** 98.904%

**Total test samples:** 25,159

**Total number of classes:** 77

**Average Precision:** 98.55%

**Average Recall:** 98.66%

**Average F1-score:** 99%

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | Precision | Recall | F1-Score | Support |
| Abu Simbel Temples | 0.988 | 0.985 | 0.987 | 341 |
| Ahmose I | 1.000 | 1.000 | 1.000 | 332 |
| Akhenaten | 0.994 | 0.997 | 0.995 | 319 |
| Al-Azhar Mosque | 1.000 | 0.974 | 0.987 | 308 |
| Al-Deir al-Bahary Temple of Queen Hatshepsut | 0.964 | 0.969 | 0.967 | 359 |
| Alexandria Library | 1.000 | 1.000 | 1.000 | 327 |
| Amenhotep III and Tiye | 1.000 | 0.997 | 0.998 | 327 |
| Amr Ibn al-Aas Mosque | 0.991 | 1.000 | 0.995 | 320 |
| Bab Zuwayla | 0.997 | 1.000 | 0.999 | 347 |
| Bab al-Nasr | 0.991 | 1.000 | 0.995 | 331 |
| Babylon Fortress | 0.997 | 1.000 | 0.998 | 313 |
| Bagawat | 1.000 | 1.000 | 1.000 | 330 |
| Baron Empain Palace | 1.000 | 1.000 | 1.000 | 299 |
| Bayt al-Suhaymi | 1.000 | 0.997 | 0.998 | 293 |
| Ben Ezra Synagogue | 1.000 | 1.000 | 1.000 | 354 |
| Bent Pyramid for Senefru | 0.991 | 1.000 | 0.995 | 317 |
| Cairo Citadel | 0.990 | 0.997 | 0.994 | 312 |
| Cairo Tower | 1.000 | 1.000 | 1.000 | 310 |
| Cavern Church Abu Serga | 1.000 | 0.997 | 0.999 | 335 |
| Cleopatra VII | 1.000 | 1.000 | 1.000 | 341 |
| Colossoi of Memnon | 0.994 | 0.982 | 0.988 | 341 |
| Deir el-Medina | 0.994 | 1.000 | 0.997 | 311 |
| Dendera Temple Complex | 0.950 | 0.956 | 0.953 | 340 |
| Djoser | 0.959 | 0.908 | 0.933 | 338 |
| Edfu Temple | 0.967 | 0.821 | 0.888 | 319 |
| Egyptian Museum, Cairo | 1.000 | 1.000 | 1.000 | 339 |
| Fatimid Cemetery in Aswan | 1.000 | 1.000 | 1.000 | 312 |
| Gayer Anderson Museum | 0.997 | 1.000 | 0.999 | 343 |
| Gebel el-Silsila | 1.000 | 1.000 | 1.000 | 314 |
| Giza Pyramid Complex | 1.000 | 0.979 | 0.989 | 326 |
| Goddess Isis with Her Child | 1.000 | 0.997 | 0.999 | 339 |
| Golden Mask of Tutankhamun | 0.997 | 0.997 | 0.997 | 342 |
| Golden Throne of Tutankhamun | 1.000 | 1.000 | 1.000 | 310 |
| Great Hypostyle Hall of Karnak | 0.986 | 1.000 | 0.993 | 343 |
| Green Head | 1.000 | 1.000 | 1.000 | 339 |
| Hanging Church (St. Virgin Mary Coptic Orthodox Church) | 0.997 | 1.000 | 0.999 | 338 |
| Horemheb | 0.997 | 1.000 | 0.999 | 338 |
| Ibn Tulun Mosque | 1.000 | 1.000 | 1.000 | 341 |
| Karnak Temple | 0.970 | 0.988 | 0.979 | 329 |
| Khafre | 1.000 | 0.981 | 0.990 | 317 |
| Khufu Statue | 0.994 | 0.994 | 0.994 | 341 |
| King Thutmose III | 1.000 | 1.000 | 1.000 | 301 |
| Kiosk of Trajan in Philae | 1.000 | 1.000 | 1.000 | 334 |
| Luxor Temple | 0.997 | 0.997 | 0.997 | 341 |
| Mausoleum of Aga Khan | 1.000 | 1.000 | 1.000 | 331 |
| Mortuary Temple of Amenhotep III | 0.991 | 0.997 | 0.994 | 347 |
| Mummy of Ramsis II | 1.000 | 1.000 | 1.000 | 353 |
| Narmer (Menes) | 1.000 | 1.000 | 1.000 | 306 |
| Narmer Palette | 1.000 | 1.000 | 1.000 | 304 |
| Nefertiti | 1.000 | 0.997 | 0.998 | 301 |
| Pompeys Pillar Alexandria | 1.000 | 1.000 | 1.000 | 302 |
| Ptolemaic Temple of Hathor in Deir el-Medina | 1.000 | 0.997 | 0.999 | 346 |
| Pyramid of Djoser | 1.000 | 0.985 | 0.992 | 328 |
| Pyramid of Unas | 1.000 | 1.000 | 1.000 | 316 |
| Qaitbay Citadel | 1.000 | 0.993 | 0.997 | 301 |
| Queen Hatshepsut | 0.967 | 0.959 | 0.963 | 338 |
| Ramesseum | 0.982 | 0.953 | 0.967 | 337 |
| Ramsis II | 0.969 | 0.972 | 0.971 | 325 |
| Ramsis II Red Granite Statue | 0.962 | 0.991 | 0.976 | 329 |
| Red Pyramid | 1.000 | 0.997 | 0.998 | 311 |
| Serapeum of Saqqara | 0.994 | 1.000 | 0.997 | 321 |
| Sesostris III | 0.994 | 1.000 | 0.997 | 329 |
| Sobekneferu | 0.997 | 1.000 | 0.998 | 311 |
| Sphinx | 0.987 | 1.000 | 0.994 | 307 |
| St. Catherine Monastery Mount Sinai | 0.994 | 1.000 | 0.997 | 354 |
| St. George Church in Coptic Cairo | 1.000 | 0.997 | 0.998 | 321 |
| Statue of Tutankhamun with Ankhesenamun | 0.997 | 1.000 | 0.998 | 312 |
| Temple of Habu | 0.985 | 0.988 | 0.987 | 339 |
| Temple of Hibis | 1.000 | 1.000 | 1.000 | 331 |
| Temple of Horus at Edfu | 0.850 | 0.963 | 0.903 | 352 |
| Temple of Isis in Philae | 0.989 | 0.983 | 0.986 | 287 |
| Temple of Khonsu in Karnak | 1.000 | 0.994 | 0.997 | 336 |
| Temple of Kom Ombo | 0.982 | 1.000 | 0.991 | 326 |
| Temple of Seti I at Abydos | 0.997 | 0.990 | 0.993 | 308 |
| Temple of the Oracle of Amun at Siwa | 1.000 | 0.997 | 0.998 | 308 |
| The Solar Boat of Khufu | 1.000 | 1.000 | 1.000 | 319 |
| The Statue of Sekhmet | 0.994 | 1.000 | 0.997 | 355 |

### 5.2.2 Comparative Analysis

#### 5.2.2.1 Quantitative Improvements

The TransVPR model demonstrated clear improvements over ResNet50 across key performance metrics:

* **Accuracy** increased from **96% (ResNet50)** to **98.9% (TransVPR)**.
* **Loss** decreased from **11.5%** **(ResNet50)** to **3.1% (TransVPR)**.
* **Average Precision** increased from **96.44% (ResNet50)** to **98.55% (TransVPR)**, representing a **2.11 percentage point improvement**.
* **Average Recall** increased from **96.57% (ResNet50)** to **98.66% (TransVPR)**, a **2.09 percentage point improvement**.
* The **F1-score** improved from **97% to 99%**, reflecting a **2 percentage point gain**, indicating a more balanced performance between precision and recall across all classes.

#### 5.2.2.2 Qualitative Insights

In addition to improved metrics, TransVPR showed noticeable advantages in handling visual confusion between similar landmarks or architectural styles:

* **Fewer misclassifications** between visually similar classes such as “Temple of Philae” vs. “Karnak Temple”.
* The incorporation of **transformer-based feature aggregation** allowed the model to capture **global contextual cues**, leading to better differentiation between historically or visually related classes.

## 5.3 Evaluation and Metrics in Chatbot model

A graph of blue bars

AI-generated content may be incorrect.

Figure 5: Chatbot Similarity Analysis based on Cosine Similarity

**Chatbot Answer Similarity Analysis (Model: llama3-8b-8192)**

The figure titled **"Chatbot Answer Similarity with Context using llama3-8b-8192"** presents an evaluation of how accurately the chatbot responds to user questions based on context retrieved from a vector database. The evaluation is based on **cosine similarity** between the chatbot's generated answers and the original context chunks.

Each bar in the chart represents a single question, and the height of the bar indicates the **cosine similarity score** (ranging from 0 to 1), where higher values reflect stronger alignment between the chatbot's response and the reference context. Most responses in this evaluation achieved a similarity score above **0.85**, indicating high semantic relevance and contextual fidelity.

Above each bar, two annotations are shown:

* **Accuracy label**: Each answer is categorized based on its similarity score. If the score is 0.80 or higher, the answer is labeled as **"Very Accurate"**. Scores between 0.70 and 0.79 are labeled **"Generally Accurate"**. Scores from 0.60 to 0.69 are considered **"Somewhat Relevant"**, and anything below 0.60 is flagged as **"Needs Review"**. In this evaluation, nearly all responses are rated as "Very Accurate", with only one falling into the "Generally Accurate" range.
* **Response time**: This shows the time (in seconds) that the model took to generate each answer, ranging from **0.72s to 1.63s**, which demonstrates the model's quick response time.

A red dashed horizontal line marks the **average similarity score (0.87)**, helping to visually compare individual scores to the overall performance. A second label shows the **average response time (1.19 seconds)**, indicating efficient inference for the model.

### 5.3.1 Evaluation Insights:

* The llama3-8b-8192 model demonstrates **high contextual relevance** across a diverse set of questions about Egyptian monuments and historical artifacts.
* The **majority of answers** meet the "Very Accurate" threshold, confirming the model’s effectiveness in context-based answering.
* The **low average generation time** supports the model's use in **real-time chatbot systems**.
* This similarity evaluation highlights the suitability of llama3-8b-8192 for **retrieval-augmented generation (RAG)** applications focused on cultural heritage and museum experiences.

# **Chapter 6: Future Work**

## 6.1 Multilingual Expansion

An important direction for future development is to enhance the entire monument system with multilingual support, ensuring accessibility and usability for users from diverse linguistic backgrounds. By incorporating multilingual models and localized content, the system can effectively serve tourists and researchers worldwide. This expansion will help promote Egyptian heritage on a global scale while improving user experience by offering services in languages such as Arabic, English, French, Spanish, and others.

## 6.2 Mobile Application Development

In the future, we plan to develop a mobile application for our monument project. Currently, the system works through a website, but creating a mobile app would make it easier and more convenient for users, especially tourists visiting museums or historical places. The app could include features like scanning monuments using the phone's camera, chatting with the assistant for information, booking museum tickets, and supporting multiple languages. This would help more people use the system easily wherever they are.

## 6.3 Augmented Reality (AR) Enhancements

In the future, we plan to improve our AR feature by adding interactive 3D models and animations that appear right on the monument when viewed through the phone camera. For example, when someone points their phone at a statue, the app could show a 3D image of how the statue looked long ago or display important events related to it. This will make learning more fun and interesting, helping visitors see history come alive and explore it in a more detailed and exciting way during their museum visit.

## 6.4 Personalized Museum Tours Using AI

We also want to add a smart tour guide that uses AI to create personalized museum tours based on what each visitor likes. Visitors can answer a few simple questions about their favorite time periods, art styles, or themes, and the AI will suggest a special path through the museum that fits their interests. For example, someone interested in Ancient Egypt could get a tour focused on pharaohs, pyramids, and hieroglyphs. This way, visitors will enjoy the exhibits that matter most to them and have a more memorable and meaningful experience.

# **References**

[1] [TransVPR Transformer with Multilevel Attention - 2022](https://openaccess.thecvf.com/content/CVPR2022/papers/Wang_TransVPR_Transformer-Based_Place_Recognition_With_Multi-Level_Attention_Aggregation_CVPR_2022_paper.pdf)

[2] [AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE](https://bibbase.org/service/mendeley/bfbbf840-4c42-3914-a463-19024f50b30c/file/264ac473-27b7-bd53-3963-f6a07df9b72e/Dosovitskiy_et_al___2021___An_Image_is_Worth_16x16_Words_Transformers_for_Im.pdf.pdf)

[3] Deep Residual Learning for Image Recognition

[4] [Monument Recognition using Artificial Intelligence - 2023](https://fcihib.journals.ekb.eg/article_306352_b38eb7151c318636a0c69b3099cab9e8.pdf?lang=en)

[5] [Attention Is All You Need - 2017](https://docalysis.com/files/hpgzy/download/attention%20all%20you%20need.pdf)

[6] [Facial Recognition Using Transfer Learning In Deep Learning - 2023](https://oarepo.org/index.php/oa/article/view/2264/2242)

[7] [Augmented Reality on Museums: the Ecomuseu Virtual Guide - 2021](https://dl.acm.org/doi/pdf/10.1145/3488162.3488219)

[8] [Augmented Egyptian Monuments dataset](file:///C:\Users\Marwa.DESKTOP-9D64D1T\AppData\Local\Microsoft\Windows\INetCache\IE\SXFZ6AIJ\•%09https:\www.kaggle.com\datasets\marwa0mar\augmented-egyptian-monuments)

[9] Llama 3 Prompt Engineering, A Comprehensive Guide

[10] Transformer (deep learning architecture)

[11] Llama 3 Guide: Everything You Need to Know About Meta's New Model and Its Data

[12] Transformer Implementation | meta-llama/llama | DeepWiki

[13] Build RAG Chatbot with LangChain, Milvus, Groq llama3-70b-8192, and voyage-3-large

[14] sentence-transformers/all-MiniLM-L6-v

[15] <https://universe.roboflow.com/landmark-object-detection/kandmarks-xwttp/dataset/3>

[16] <https://universe.roboflow.com/landmark-object-detection/detection-jsxva/dataset/6>

[17] <https://universe.roboflow.com/gp-yhogh/egyptian-artifacts/dataset/1>

[18] <https://universe.roboflow.com/gp-yhogh/egypt-attractions/dataset/1>

[19] <https://universe.roboflow.com/eg-q7nxw/eg-xi4vg/dataset/1>

[20] <https://universe.roboflow.com/fatma-saeed-th9nx/egypt-landmarks/dataset/5>

[21] <https://universe.roboflow.com/landmark-object-detection/landmarks-for-fasterrcnn-mask/dataset/1>

[22] <https://universe.roboflow.com/etlas/monuments-detection/dataset/3>

[23] <https://www.kaggle.com/datasets/mohamedelgabry22/egyption-ancient-statues>

[24] <https://www.kaggle.com/datasets/mohamedyunis/tourism>

[25] <https://www.kaggle.com/datasets/daliayasser/egyptian-monuments>

[26] <https://drive.google.com/drive/folders/1hBzNW2XOp41v58GPTDITJvNxrug1U8b?usp=sharing>

[27] <https://universe.roboflow.com/statues/ramsis>

[28] <https://universe.roboflow.com/conqueror-13-inlcc/rams2>

[29] [Google Landmarks Dataset v2 (GLMDv2)](https://github.com/cvdfoundation/google-landmark)

[30] [**Sekhmet\_The\_Great\_Huntress\_of\_the\_Night**](https://www.academia.edu/25380529/Sekhmet_The_Great_Huntress_of_the_Night)

[31] [**Sekhmet Statues from the reign of Amenhotep iii in the britsh museum**](https://www.academia.edu/75693217/SEKHMET_STATUES_FROM_THE_REIGN_OF_AMENHOTEP_iii_IN_THE_BRITISH_MUSEUM_and_A_FORMERLY_UNCATALOGUED_HEAD_FRAGMENT_A_REASSESSMENT)

[32] [**Travellers from an Antique Land Statues**](https://www.academia.edu/12818150/_Travellers_from_an_Antique_Land_Statues_of_the_Goddess_Sekhmet_in_the_British_Museum)

[33] [**Sekhmet & Bastet- The Feline Powers of Egypt**](https://ia903101.us.archive.org/30/items/sekhmetbastetthefelinepowersofegyptpdfdrive.com_201911/Sekhmet%20%26%20Bastet-%20The%20Feline%20Powers%20of%20Egypt%20%28%20PDFDrive.com%20%29.pdf)

[34] [**The statue of Sekhmet**](https://ia803401.us.archive.org/12/items/jstor-3253827/3253827.pdf)

[35] [**Tutankhamun and his treasures**](https://www.academia.edu/83255197/TUTANKHAMUN_AND_HIS_TREASURES)

[36] [**The Complete Tutankhamun by Nicholas Reeves**](https://archive.org/details/TheCompleteTutankhamunByNicholasReeves)

[37] [**Restoration of the Historic Church of ST**](https://www.academia.edu/70100998/Restoration_of_the_Historic_Church_of_ST_George_in_Cairo_and_the_Roman_Tower_Underneath)

[38] [**Strengthening Restoration and Rehabilita**](https://www.academia.edu/14285074/Strengthening_Restoration_and_Rehabilitation_of_the_Greek_Orthodox_Church_of_Saint_George_Old_Cairo)

[39] [**The Monastery of Mount Sinai and the Cult of Saint Catherine**](https://www.academia.edu/2137444/The_Monastery_of_Mount_Sinai_and_the_Cult_of_Saint_Catherine)

[40] [**The solar boat of Khufu**](https://www.brown.edu/Departments/Joukowsky_Institute/courses/maritimearchaeology11/files/18445329.pdf)

[41] [**Sobekneferu and Her Legacy of Female Mas**](https://www.academia.edu/120391380/Sobekneferu_and_Her_Legacy_of_Female_Masculinity)

[42] [**Gender Deities and the Public Image of Sobekneferu**](https://www.academia.edu/98906849/Gender_Deities_and_the_Public_Image_of_Sobekneferu)

[43] [**The First Female Pharaoh (Andrew Collins)**](https://z-library.sk/book/25253121/b5aba1/the-first-female-pharaoh-sobekneferu-goddess-of-the-seven-stars.html)

[44] [**Restoration of the Mosaic of the Transfiguration in St. Catherine Monastery on Mount-Sinai**](https://ccaroma.org/wp-content/uploads/2018/03/Nardi_2011_Restoration-of-the-Mosaic-of-the-Transfiguration-in-St.-Catherine-Monastery-on-Mount-Sinai.pdf)

[45] [**Sobekneferu Egypt's First Female Pharaoh**](https://www.academia.edu/105497940/Sobekneferu_Egypts_First_Female_Pharaoh_and_Her_Rebirth_as_a_Modern_Pop_Icon)

[46] [**Sesostris The Image**](https://www.academia.edu/312369/Sesostris_Sesonchosis_Sesoosis_The_Image_of_the_Great_King_of_the_Past_and_Its_Connotations_of_the_Lybian_Time_in_Egypt)

[47] [**Senusret III A Pharaoh A Collector A Leg**](https://www.academia.edu/43804373/Senusret_III_A_Pharaoh_A_Collector_A_Legacy_in_the_Sand)

[48] [**Ancient Records of Egypt Vol-3**](https://archive.org/details/ancient-records-of-egypt-vol-3_202011)

[49] [**Serapeum**](https://www.diva-portal.org/smash/get/diva2%3A1589765/FULLTEXT01.pdf)

[50] [**The Serapeum of Saqqara Alternative theory for the granite coffers**](https://www.academia.edu/37092601/The_Serapeum_of_Saqqara_Alternative_theory_for_the_granite_coffers)

[51] [**The Red Pyramid**](https://www.researchgate.net/publication/320131285_The_Red_Pyramid_A_layman's_guide)

[52] [**Ramesses, Loved by Ptah The History of a Colossal Royal Statue (Susanna Thomas)**](https://1lib.sk/book/28519139/72c6c5/ramesses-loved-by-ptah-the-history-of-a-colossal-royal-statue.html)

[53] [**The Complete Temples of Ancient Egypt**](https://www.mediafire.com/file/vg8ff7buikp4xq4/the+completeTemples+of+Ancient+Egypt.pdf?fbclid=IwAR2YX7l_xSN2Zij752TwOHtxCugvTsPhS1JrVOPQ4BMyWk1wYhHz89C7CAo)

[54] [**Ancient Egypt - A Very Short Introduction**](https://www.mediafire.com/file/tmaxtsaa50ep4ys/Ancient+Egypt+-+a+Very+Short+Introduction.pdf?fbclid=IwAR2gbxWUh-qt7jSCznRYfnYUzNjmEvWau37_MVcVUOSWo6fGjbDdL_-Ypcs)

[55] [**Ramses III: The Life and Times of Egypt’s Last Hero**](https://www.mediafire.com/file/tmaxtsaa50ep4ys/Ancient+Egypt+-+a+Very+Short+Introduction.pdf?fbclid=IwAR2gbxWUh-qt7jSCznRYfnYUzNjmEvWau37_MVcVUOSWo6fGjbDdL_-Ypcs)

[56] [**Complete Pyramids: Mark Lehner**](https://drive.google.com/file/d/1Mqr1I_J_1STP_nIS_EDDol01vqHHLzWa/view?usp=drive_link)

[57] [**History of Egypt - Wikipedia paper**](https://drive.google.com/file/d/1SBKwgwl0pyadUWzU6GvvnIFer2cpX-0V/view?usp=drive_link)

[58] [**Encyclopedia of Ancient Egypt**](https://library.qu.edu.iq/wp-content/uploads/2020/09/2.pdf)

[59] [**Early Dynastic Egypt: Toby A.H.Wilkinson**](https://api.pageplace.de/preview/DT0400.9781134664207_A24596440/preview-9781134664207_A24596440.pdf)

[60] [**Akhenaten and The Religion of Light**](https://toaz.info/doc-view-3)

[61] [**Egyptian Art and Architectur**](https://www.researchgate.net/publication/384051485_Egyptian_art_and_architecture_Reading)

[62] [**A BRIEF HISTORY OF EGYPT: ARTHUR GOLDSCHMIDT JR.**](https://library.diplomatic.ac/wp-content/uploads/2020/01/epdf.pub_a-brief-history-of-egypt-brief-history.pdf)

[63] [**The Illustrated Guide to the Egyptian Museum**](https://archive.org/details/illustratedguide0000unse_s5e9)

[64] [**A history of Egypt in the Middle Ages - Stanley Lane-Poole**](https://www.noor-book.com/en/ebook-A-History-of-Egypt-in-the-Middle-Ages-pdf)

[65] [**The Arab conquest of Egypt and the last thirty years of the Roman dominion**](https://upload.wikimedia.org/wikipedia/commons/5/5d/The_Arab_conquest_of_Egypt_and_the_last_thirty_years_of_the_Roman_dominion_%28IA_arabconquestofeg00butl%29.pdf)

[66] [**The Cambridge History of Egypt, Vol. 1: Islamic Egypt, 640-1517**](https://www.zlibrary.to/dl/the-cambridge-history-of-egypt-vol-1-islamic-egypt-640-1517)

[67] [**Aristotle’s Journey to Europe: A Synthetic History of the Role Played by the Islamic Empire in the Transmission of Western Educational Philosophy Sources from the Fall of Rome through the Medieval Period**](https://core.ac.uk/download/pdf/213385291.pdf)

[68] [**A Comopolitan City (Muslims, Christian, and Jews in Old Cairo)**](https://isac.uchicago.edu/sites/default/files/uploads/shared/docs/oimp38.pdf)

[69] [**The Cambridge History of Egypt Volume 2**](https://archive.org/details/the-cambridge-history-of-egypt-volume-2)

[70] [**Brief report of the project of the second boat of king Khufu**](https://journals.librarypublishing.arizona.edu/jaei/article/id/1046/)

[71] [**Modern Egypt, from 1517 to the end of the twentieth century**](https://www.pdfdrive.com/the-cambridge-history-of-egypt-vol-2-modern-egypt-from-1517-to-the-end-of-the-20th-century-e161088845.html)