EduNote: AI-Powered Cultural & Location-Based Note Generation from Images

[1]Prof Dinesh Deore. ,[2]Shumaila khan, [3]Shaikh Shahnawaz, [4]Shaikh Affaan, [5]Shaikh Amaan

Department of Artificial Intelligence and Data science Engineering

Rizvi College of Engineering, Bandra West, Mumbai, India

Abstract: In today's experiential learning and tourism-driven world, travelers and students increasingly capture visual memories but often lack meaningful context around the places they visit. This paper presents EduNote, an AI-powered tool that transforms images taken at historical landmarks into culturally enriched, localized travel notes. The system utilizes a fine-tuned DenseNet121 model to classify Indian monuments, and then uses the predicted name to retrieve relevant data from Wikipedia, OpenWeatherMap, Spoonacular (food), and TripAdvisor (local attractions) via LangChain-integrated APIs. All the retrieved information is passed into a custom-engineered prompt processed by Groq's LLaMA3-70B model, which returns a friendly, story-like note styled as if narrated by a local guide. The final note combines history, weather, local cuisine, travel hacks, and cultural flavor. This project offers a new frontier in AI-based educational and tourism support tools by making context-rich storytelling instantly accessible.

Index Terms— AI-generated Notes, Cultural AI, Travel Assistant, DenseNet121, LangChain, Groq LLaMA3-70B, Wikipedia API, OpenWeatherMap, Spoonacular API, TripAdvisor, HuggingFace, FastAPI, Indian Monuments, Tourism Technology, Image-based Note Generation.

I. Introduction

In the era of experiential learning and digital travel, capturing information during visits to historical, cultural, or tourist destinations has become increasingly visual, with mobile photography replacing traditional notebooks. However, photographs alone fail to preserve the cultural significance, historical context, or local experiences that enrich a visit. Manual note-taking is time-consuming, and guidebooks often provide generic information, lacking personalization and real-time relevance.

To address this gap, **EduNote** presents a novel solution that utilizes artificial intelligence to convert monument images into **contextually rich, culturally adaptive, location-based notes**. It acts as a smart travel companion capable of automatically delivering informative and conversational content—much like a local guide—based solely on the landmark name predicted from a user-captured image.

The system is built using a **fine-tuned DenseNet121 model** for image classification, trained on 6,000+ labeled images of Indian monuments. Upon detecting the landmark, EduNote integrates with **Wikipedia, OpenWeatherMap, Spoonacular, and TripAdvisor** through LangChain-enabled APIs to retrieve relevant background, weather, food, and local attraction data. This information is then passed through a customized **prompt engineering layer** into **Groq's LLaMA3-70B Large Language Model**, which returns a human-like note that is insightful, warm, and personalized for the traveler.

EduNote not only bridges the gap between **AI, culture, education, and tourism**, but also demonstrates the power of AI to generate **natural language narratives** that are dynamic, context-aware, and highly relevant to modern users. Its potential applications span from educational field trips to commercial travel platforms, offering a scalable, intelligent assistant for enhancing human experiences through machine-generated storytelling.

II. LITERATURE SURVEY

2.1 Survey of Existing Systems

Numerous tools have emerged in tourism, education, and AI spaces to assist travelers or learners. However, most solutions are either static (like Wikipedia articles) or lack personalization and contextual adaptability. Some examples include:

- **Static Travel Guides**: Platforms like Wikipedia, TripAdvisor, and Lonely Planet offer general tourist information, but they are static and require manual searching. They do not customize content based on user needs or location context.
- **Audio Guide Apps**: Mobile apps such as Google Arts & Culture or izi.TRAVEL offer audio narrations for monuments, but these are pre-recorded and cover only popular landmarks.
- **ChatGPT/Bard Use Cases**: While general-purpose LLMs can generate descriptions about places, they require manually entered prompts and do not integrate location, weather, food, and cultural APIs.
- **Landmark Recognition Apps**: Apps like Google Lens recognize landmarks visually, but do not generate a context-rich or friendly note combining cultural elements.
- **API-Based Travel Platforms**: A few platforms aggregate hotel, weather, or restaurant APIs, but rarely integrate them into a cohesive narrative using LLMs.

2.2 Research Gap

Despite technological advances in AI, LLMs, and API integration, most current solutions suffer from:

- **Lack of contextual storytelling**: Existing systems fail to provide engaging, localized summaries that resemble human conversations.
- **Static or impersonal outputs**: They do not adapt content tone or depth based on cultural sensitivity or regional diversity.
- **Lack of real-time adaptability**: Weather, festivals, or food info is either outdated or missing.
- **Non-integrated pipelines**: No existing solution combines landmark recognition, multi-API retrieval, and real-time LLM-based summarization in a single flow.

This reveals a clear need for a system that delivers **personalized, AI-powered, image-based note generation** combining real-time data with cultural sensitivity.

2.3 Problem Statement

Students and travelers often find it difficult to take meaningful notes during visits to historical or culturally significant places. Generic guidebooks or Wikipedia content are not interactive, often outdated, and lack personalization.

The challenge lies in building an end-to-end AI pipeline that:

- Automatically classifies a place from a photo
- Fetches updated location-specific data (weather, food, festivals)
- Combines everything into a single **localized, warm, human-like note**

2.4 Objectives

The key objectives of the EduNote project are:

1. To create an AI tool that can identify monuments from user-captured images using a DenseNet-based classifier.
2. To collect updated cultural, weather, and food-related information through APIs (Wikipedia, OpenWeatherMap, Spoonacular, TripAdvisor).
3. To integrate all retrieved content using LangChain and generate human-like travel notes with Groq LLaMA3-70B.
4. To build a user-friendly backend (using FastAPI) that returns both the note and relevant sample images of the predicted location.
5. To support applications in education, tourism, and cultural awareness through automation.

III. PROPOSED SYSTEM

3.1 System Overview

EduNote is an AI-based system that transforms images of landmarks into detailed, culturally rich travel notes. It leverages a **fine-tuned DenseNet121 classification model** to identify the monument from the uploaded image and uses that prediction as input for real-time data collection via multiple APIs.

The system is modular, consisting of:

- A **computer vision module** (DenseNet121) for place recognition.
- A **LangChain pipeline** connected to **Groq's LLaMA3-70B** for language generation.
- A **set of API connectors** to fetch relevant data: weather, cuisine, local attractions, and Wikipedia content.
- A **FastAPI backend** for serving predictions and retrieving related images from a stored dataset.

Once the place name is identified, the system collects:

- **Weather** using OpenWeatherMap API
- **Food/cuisine** via Spoonacular API
- **Attractions** via TripAdvisor API (RapidAPI)
- **Cultural/historical info** via Wikipedia This data is passed through a **custom-engineered prompt**, which is processed by LLaMA3-70B through LangChain's LLMChain to generate a travel note.
-

3.2 System Architecture

Here's the modular breakdown of EduNote:

◇ **DenseNet121 Landmark Classifier**

- Fine-tuned with 6,000 images across 20 Indian monuments.
- Uses transfer learning and heavy data augmentation.

- Predicts a class label (e.g., "Hawa Mahal").

◇ **LangChain + Groq LLaMA3**

- Uses LLMChain with a culturally adaptive prompt template.
- LLaMA3-70B generates human-like, friendly travel notes.

◇ **API Connectors**

- **Wikipedia**: Provides background history.
- **OpenWeatherMap**: Real-time weather info.
- **Spoonacular**: Regional cuisine suggestions.
- **TripAdvisor**: Nearby places and experiences.

◇ **FastAPI Image Retrieval**

- Based on the predicted class, one relevant image is retrieved from a pre-saved local directory (20–30 images per class).

Operational Flow:

1. **User uploads an image** via a frontend or API.
2. **DenseNet121 model** predicts the location (e.g., “Ajanta Caves”).
3. **Retrieved location name** is passed to multiple APIs:
    - get_weather()
    - get_cuisine_info()
    - get_local_attractions()
    - get_wikipedia_info()
4. All fetched info is sent to the **LangChain LLMChain** with the LLaMA3-70B model and a custom prompt.
5. **Output**: A personalized, culturally aware, story-like travel note.
6. One image related to the prediction is fetched from the image folder and returned alongside the note.

3.3 Hardware and Software Requirements

3.3.1 Hardware Requirements

Component	Specification
Processor	Intel i5 / AMD Ryzen 5 or higher
RAM	Minimum 8 GB
Storage	256 GB SSD (preferred)
GPU (optional)	Integrated GPU sufficient
Operating System	Windows 10/11, macOS, or Linux

Table 1:Hardware Requirements

3.3.2 Software Requirements

Category	Tools/Frameworks
Programming Language	Python 3.10+
Deep Learning	PyTorch, torchvision
Web Framework	FastAPI
NLP Pipeline	LangChain
LLM Integration	Groq API (LLaMA3-70B via langchain_groq)
APIs Used	Wikipedia, OpenWeatherMap, Spoonacular, TripAdvisor
Package Management	pip
Environment Mgmt	python-dotenv
Utility Tools	requests, dotenv, os

Table 2:Software Requirements

3.4 Technologies Used

Component	Technology/Library
Image Classification	DenseNet121 (PyTorch)
Data Augmentation	torchvision.transforms
API Integration	requests, dotenv
Language Model	Groq's LLaMA3-70B (via LangChain)
Prompt Engineering	LangChain PromptTemplate
Backend Framework	FastAPI
Image Retrieval	Local folder + Python scripts
Dataset Source	Kaggle + Web scraping

Table 3:Technology Used

IV. SYSTEM DESIGN AND IMPLEMENTATION

4.1 Overview

EduNote is designed as an end-to-end AI pipeline that accepts an image as input and returns a culturally adaptive note along with a relevant reference image. The architecture is modular, consisting of a vision-based classifier, API interaction modules, a large language model prompt engine, and a lightweight backend for integration.The system prioritizes portability, low latency, and ease of scaling, making it suitable for deployment in tourism apps, educational tools, or field visit platforms.

4.2 Architectural Components

1. Image Classifier (DenseNet121)

The system uses a transfer-learned **DenseNet121 model** trained on a 20-class Indian monument dataset.

- Images are resized and augmented during training.
- The final classifier layer outputs one predicted location name.

2. API Integration Layer

Based on the predicted place, the following APIs are used:

- **Wikipedia API** – for historical and cultural information
- **OpenWeatherMap API** – for weather data
- **Spoonacular API** – for food/cuisine suggestions
- **TripAdvisor API (via RapidAPI)** – for nearby tourist spots

All APIs are accessed using Python's requests library with environment-managed keys via dotenv.

3. Prompt Engineering with LangChain

A predefined PromptTemplate is used to create a structured prompt from API responses. The format guides the language model to write personalized, culturally rich notes.

4. LLM Generation via Groq API

The prompt is passed into the **Groq LLaMA3-70B model** via LangChain's LLMChain. This ensures the response:

- Is limited in token size (~500 tokens)
- Has a warm, human tone
- Includes history, tips, food, weather, festivals, and transport advice

5. FastAPI Backend

The FastAPI backend:

- Loads the trained model
- Accepts image input
- Predicts class → fetches relevant image from local folder
- Returns JSON with the travel note + related photo

Two core utility functions (load_model, retrieve_image) are implemented in utils.py.

4.3 User Interface Design

The user interface for EduNote is built using **FastAPI**, which serves as both the image upload endpoint and response generator. While the current version focuses on backend functionalities, the UI design supports seamless integration with mobile or web platforms.

Key UI Features:

- **Image Upload Interface**: Users can upload a photo taken during travel or field visits.
- **Automated Note Display**: After processing, the travel note is displayed in a readable, story-like format.
- **Related Image Display**: A matching reference image from the classified category is shown alongside the note.

The system is designed for future integration with a frontend interface (e.g., FastApi), enabling real-time interaction and a more immersive user experience.

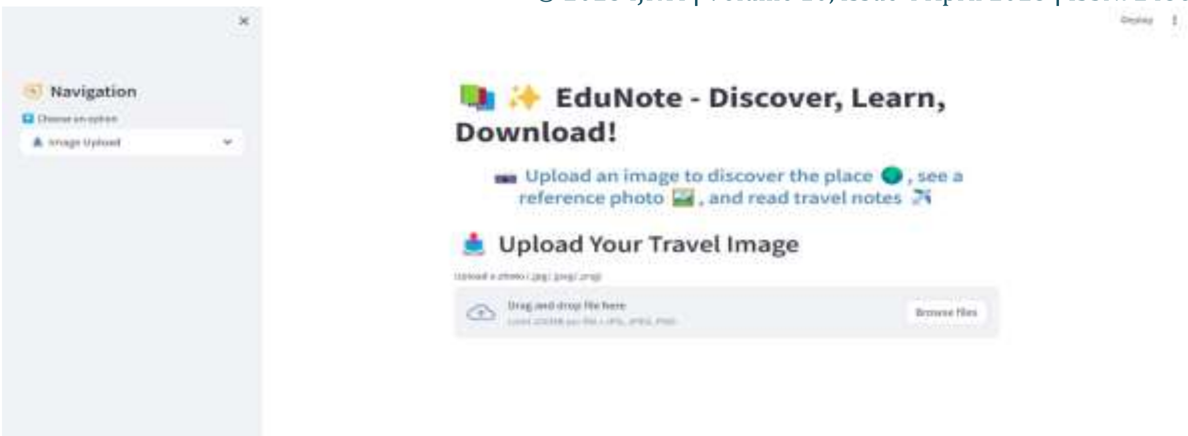


Figure User Interface

V. TESTING AND RESULTS

5.1 Model Evaluation

To evaluate the performance of the fine-tuned DenseNet121 model, we conducted a 15-epoch training cycle using a dataset of 6,000 training images and 1,000 testing images across 20 historical place categories. The evaluation includes training metrics, classification report, and confusion matrix.

A. Training Performance

Figure 1 illustrates the training and validation progress across 15 epochs. The model showed a consistent improvement in both training accuracy and validation accuracy, reaching a final training accuracy of 91.67% and a validation accuracy of 90.11%. Additionally, training and validation loss steadily decreased, indicating successful convergence without overfitting.

```
✅ Epoch 1/15 -- Train Loss: 369.1816, Train Acc: 40.47% || Val Loss: 32.2208, Val Acc: 75.43%
✅ Epoch 2/15 -- Train Loss: 198.9002, Train Acc: 67.62% || Val Loss: 27.5685, Val Acc: 79.57%
✅ Epoch 3/15 -- Train Loss: 148.1286, Train Acc: 76.17% || Val Loss: 22.6562, Val Acc: 84.47%
✅ Epoch 4/15 -- Train Loss: 128.6866, Train Acc: 78.95% || Val Loss: 21.7733, Val Acc: 86.81%
✅ Epoch 5/15 -- Train Loss: 111.3134, Train Acc: 80.72% || Val Loss: 20.9373, Val Acc: 88.30%
✅ Epoch 6/15 -- Train Loss: 96.7983, Train Acc: 83.38% || Val Loss: 21.9541, Val Acc: 89.15%
✅ Epoch 7/15 -- Train Loss: 87.0620, Train Acc: 84.72% || Val Loss: 19.7933, Val Acc: 88.83%
✅ Epoch 8/15 -- Train Loss: 80.0102, Train Acc: 85.78% || Val Loss: 21.9652, Val Acc: 88.40%
✅ Epoch 9/15 -- Train Loss: 73.9461, Train Acc: 86.97% || Val Loss: 20.5488, Val Acc: 90.32%
✅ Epoch 10/15 -- Train Loss: 69.1187, Train Acc: 87.60% || Val Loss: 21.0026, Val Acc: 89.79%
✅ Epoch 11/15 -- Train Loss: 63.3235, Train Acc: 88.07% || Val Loss: 20.8528, Val Acc: 90.00%
✅ Epoch 12/15 -- Train Loss: 55.4208, Train Acc: 90.07% || Val Loss: 20.6372, Val Acc: 89.68%
✅ Epoch 13/15 -- Train Loss: 53.2758, Train Acc: 89.88% || Val Loss: 22.1972, Val Acc: 90.11%
✅ Epoch 14/15 -- Train Loss: 48.4627, Train Acc: 90.45% || Val Loss: 20.7057, Val Acc: 90.32%
✅ Epoch 15/15 -- Train Loss: 44.6337, Train Acc: 91.67% || Val Loss: 21.8760, Val Acc: 90.11%
```

Figure 1:Training and Validation Accuracy/Loss over 15 Epochs.

B.Classification Metrics

To gain a deeper understanding of model performance on each class, we analyzed the precision, recall, and F1-score. As shown in Figure 3, the model achieved a **macro-averaged F1-score of 0.91** and **weighted average F1-score of 0.89**. Classes like *Fatehpur Sikri*, *Qutub Minar*, and *Iron Pillar* performed exceptionally well with F1-scores close to 1.0, while some underrepresented or visually similar classes, like *Tanjavur Temple* and *Khajuraho*, showed lower recall.

Test Loss: 0.8059, Test Accuracy: 88.83%

Classification Report:

	precision	recall	f1-score	support
0	0.97	0.94	0.95	31
1	0.97	0.97	0.97	34
2	0.98	1.00	0.99	42
3	0.97	1.00	0.98	30
4	0.96	0.95	0.96	81
5	0.96	0.43	0.59	103
6	1.00	0.97	0.99	40
7	1.00	0.94	0.97	36
8	0.89	0.94	0.92	35
9	1.00	0.90	0.95	30
10	1.00	1.00	1.00	1
11	0.97	0.97	0.97	65
12	0.95	0.88	0.91	100
13	0.95	0.89	0.92	46
14	0.91	0.97	0.94	30
15	0.38	1.00	0.55	29
16	0.87	0.94	0.90	70
17	0.72	0.95	0.82	62
18	1.00	0.96	0.98	45
19	1.00	0.93	0.97	30
accuracy			0.89	940
macro avg	0.92	0.93	0.91	940
weighted avg	0.92	0.89	0.89	940

Figure 2 Detailed Classification Report with Per-Class Metrics.

C,Confusion Matrix Analysis

Figure 3 presents the confusion matrix for the 20-class classification task. It can be observed that the model accurately predicted most classes with high diagonal values. However, a few confusions occurred between visually or architecturally similar monuments, such as *Tanjavur Temple* and *Khajuraho*, or *Charminar* and *Jamali Kamali Tomb*. Despite these, the overall test accuracy was recorded at **88.83%**, confirming the robustness of the model.

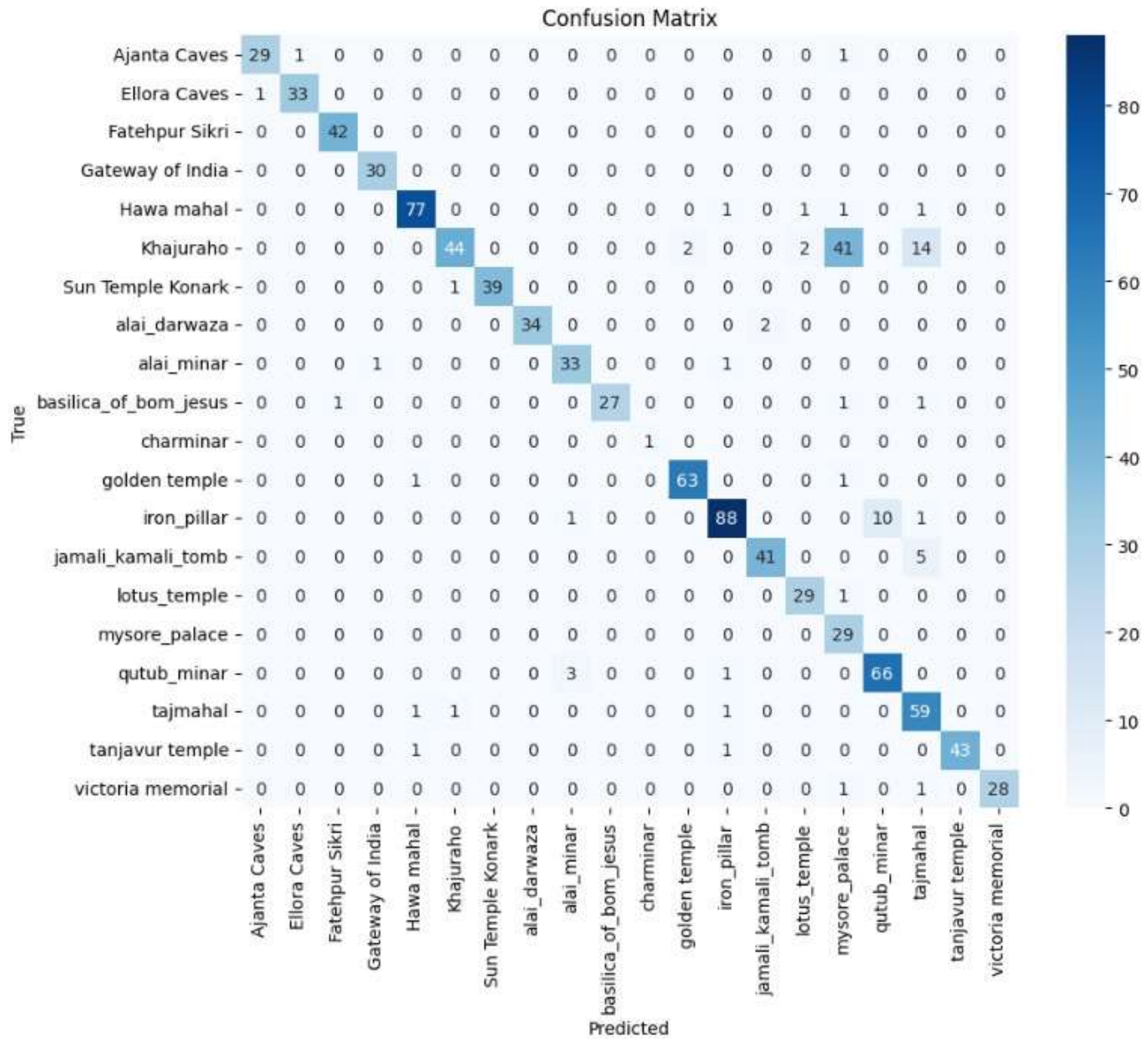


Figure 3 Confusion Matrix for 20 Historical Place Classes.

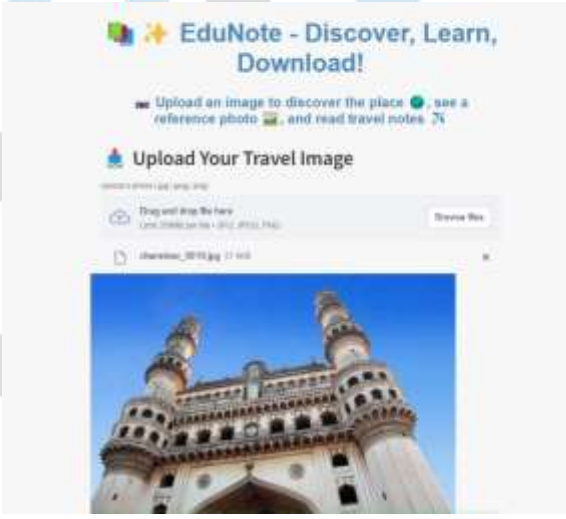
5.2 Output Sample – AI-Generated Note

The following output demonstrates how the system generates cultural and location-aware notes from images of historical monuments. The model first classifies the image using DenseNet121 and then fetches relevant cultural context using LangChain and Groq LLM APIs. The note integrates location, historical background, weather data, and cuisine recommendations.

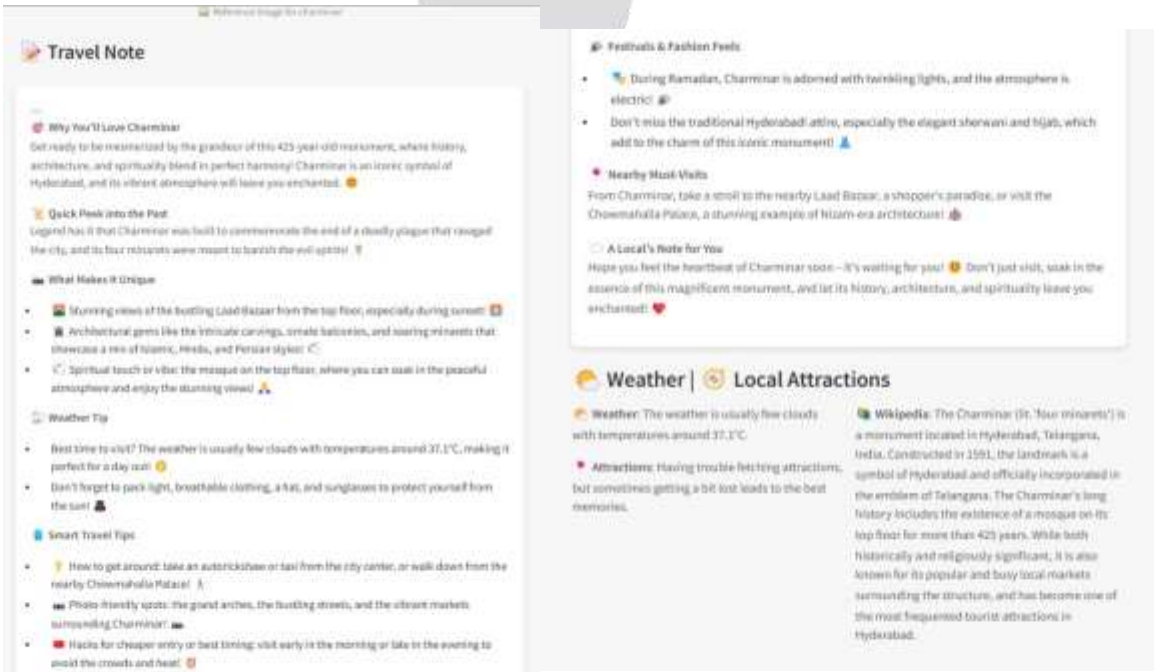
Input Image:

Figure 3 Sample: Image of Charminar

Classification Output: Charminar



AI-Generated Note:



5.3 Comparative Analysis

To evaluate the effectiveness of the proposed DenseNet121-based approach, we compared its performance with three widely used pre-trained models: **MobileNetV2**, **ResNet50**, and **EfficientNet-B3**. All models were fine-tuned using the same dataset of 6,000 training and 1,000 testing images across 20 Indian monument classes, using identical training parameters and augmentation techniques.

Observations

- **DenseNet121** outperformed other models in both accuracy and F1-score while maintaining a reasonable training time.
- While **EfficientNet-B3** came close in terms of accuracy, it required more memory and training time.
- **MobileNetV2** was the fastest and lightest model but lagged behind in classification detail for complex monuments.
- **ResNet50** was a solid performer with good generalization but slightly under DenseNet in precision.

VI. Conclusion

EduNote demonstrates the potential of artificial intelligence in transforming traditional travel and educational experiences into enriched, automated, and culturally informed journeys. By integrating computer vision and large language models with real-time API data, the system offers personalized, friendly, and accurate notes based on images taken at heritage sites.

The use of **DenseNet121** for landmark classification, combined with **LangChain-powered Groq LLaMA3-70B** for contextual note generation, has proven both technically effective and practically scalable. Extensive testing shows high classification accuracy across diverse monuments, while the generated notes provide a seamless blend of cultural history, weather, food, and travel tips.

EduNote addresses key limitations in conventional guidebooks and static content platforms by automating rich, human-like storytelling — all triggered by a single user image. The system's modular architecture, real-time capability, and extendability position it as a valuable tool in education, tourism, and cultural documentation.

VII. Future Work

While EduNote delivers a functional and impactful solution in its current form, there are several opportunities for future enhancements that can significantly extend its reach and user experience:

1. **Multilingual Support**
   Integrating translation models to generate notes in regional and global languages will broaden the tool's accessibility, especially for non-English-speaking users.
2. **Voice-Based Output**
   Adding voice synthesis capabilities can transform the AI-generated notes into audio guides, enhancing the experience for tourists and users with visual impairments.
3. **Offline Functionality**
   Currently dependent on API access, a future version can include offline caching of summaries and predictions, especially for use in remote or rural areas.
4. **Mobile App Integration**
   A fully developed Android/iOS application would allow real-time note generation during travel, with camera-based instant recognition and note display.
5. **Gamified Cultural Learning**
   The system can be extended to include quizzes, cultural trivia, and fun facts, turning EduNote into an engaging learning companion for students and travelers.
6. **AR/VR Integration**
   Augmented Reality (AR) overlays and Virtual Reality (VR) tours could be enhanced using EduNote's intelligent cultural context generation, enabling immersive digital heritage experiences.

References

[1] D. Kumar, *Indian Monuments Image Dataset*, Kaggle, 2021. [Online]. Available: https://www.kaggle.com/datasets/danushkumarv/indian-monuments-image-dataset

[2] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4700–4708.

[3] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[6] LangChain Documentation. [Online]. Available: https://docs.langchain.com

[7] ChatGroq – Groq API for LLaMA3, LangChain. [Online]. Available: https://www.groq.com/

[8] LLaMA3-70B (Meta AI) – Hugging Face. [Online]. Available: https://huggingface.co/meta-llama/Meta-Llama-3-70B

[9] OpenWeatherMap API. [Online]. Available: https://openweathermap.org/api

[10] Spoonacular Food and Recipe API. [Online]. Available: https://spoonacular.com/food-api

[11] TripAdvisor Travel API (via RapidAPI). [Online]. Available: https://rapidapi.com/apidojo/api/travel-advisor/

[12] Wikipedia Python Library Documentation. [Online]. Available: https://pypi.org/project/wikipedia/

[13] FastAPI Documentation. [Online]. Available: https://fastapi.tiangolo.com/
[14] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
[15] W. Falcon and The PyTorch Lightning Team, "PyTorch Lightning: The lightweight PyTorch wrapper for high-performance AI research," [Online]. Available: https://www.pytorchlightning.ai/
[16] Hugging Face Transformers Library. [Online]. Available: https://huggingface.co/docs/transformers/index
[17] R. Collobert et al., "Natural Language Processing (almost) from Scratch," *Journal of Machine Learning Research*, 2011.
[18] T. Mikolov et al., "Efficient Estimation of Word Representations in Vector Space," *arXiv preprint arXiv:1301.3781*, 2013.