OSTBAYERISCHE TECHNISCHE HOCHSCHULE REGENSBURG



Applied Data Science with Python – Mr. Lengyel –WiSe23

Final Report

Applied Data Science with Python

Tourist Attractions Recommendation Software

Advisor: Istvan Lengyel

Students: Nguyen Huy Hoang - 3421125.

Contents

1	Introduction 2			
	1.1	Requirements Analysis		
	1.2	Benefits for Users		
2	Frontend 3			
	2.1	Templates		
	2.2	HTML Structures		
	2.3	Static		
	2.4	CSS Structures		
3	Bac	ekend 5		
	3.1	Model		
4	3.2	Routing		
	3.3	Rendering		
	3.4	Databases		
	3.5	Flowchart		
4	Deep Learning Models 8			
	4.1	Summary		
	4.2	Classification		
	4.3	Dataset Collecting		
	4.4	Training		
	4.5	Export Models		
5	Implementation 11			
	5.1	Home Page		
	5.2	Searching		
	5.3	Exploring		
	5.4	Image Searching		
6	Summary 14			
	6.1	Github		
	6.2	Conclusion		
	6.3	Sources		



1 Introduction

1.1 Requirements Analysis

Creating a software that recommends the famous and interesting tourist attrations in Paris, France.

The software should provide following functionality:

- 1. List of available attrations.
- 2. Search by Names to find the nearest places.
- 3. AI feature to automatically suggest the matching tourist places based on image that users upload.

1.2 Benefits for Users

This system can help users obtain attractions that match their preferences, save time, and improve their satisfaction with the platform. It is easier and more comfortable for users since they do not have to search for places on Google, check it on Maps, read the reviews on the other websites. Everything is integrated into one platform.

By using deep learning algorithms, this system can identify tourist attractions based on images, which can be more effective than text-based searches. Additionally, this system can provide personalized recommendations based on geographical location and interests. This can help users plan travel routes and itineraries more efficiently.

2 Frontend

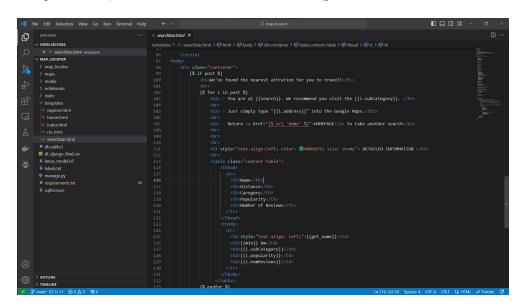
In Brief: During the time considering, my plan has a little change in the frontend from Reactjs that I decided to used in Task 1, to Django templates. Because I think it is more convenient for me to using it in only one framework for both frontend and backend.

2.1 Templates

In order to use Templates HTML in Django, I added ${\bf BASE_DIR}$ / templates directory to the system.

2.2 HTML Structures

For each specific task, I created one .html file for rendering interface.



2.3 Static

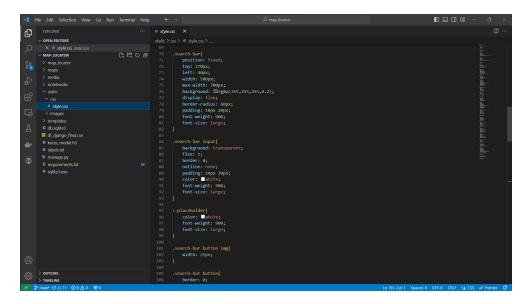
In order to use Static CSS in Django, I added BASE DIR / 'static' directory to the system

```
STATICFILES_DIRS = [
BASE_DIR / 'static'
]
```

2.4

CSS Structures

Style.css will handle all of the .html file in Templates HTML by $\{\% \text{ load static } \%\}$ command.





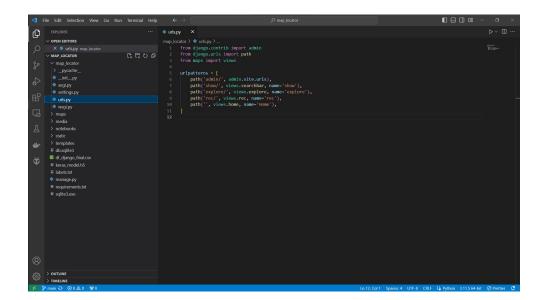
Backend 3

3.1 Model

In Brief: Used to store data from csv file and retrieve data to display on HTML page.

Routing

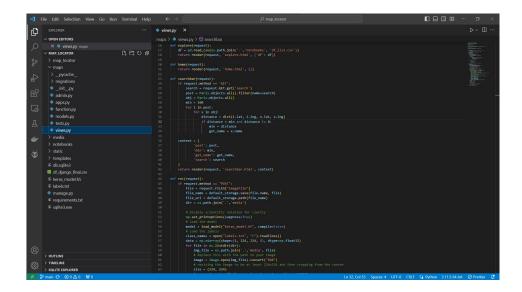
In Brief: Route the website to different urls making every specific function.





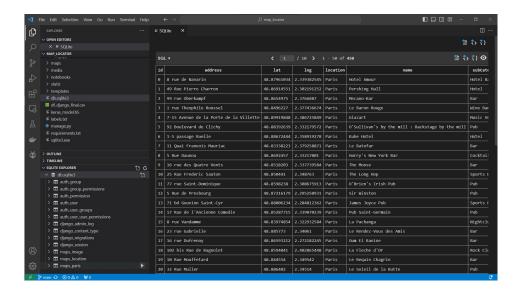
3.3 Rendering

In Brief: Use Django framework to process users' API call by **post** and **get** with views.py. Additionally, it also help users implement Image Classification models within the software.



3.4 Databases

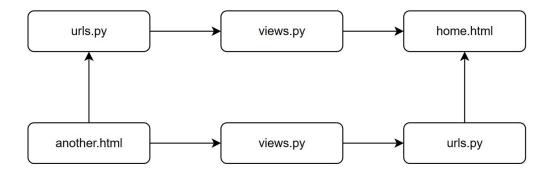
In Brief: Show the SQLite integrated into VS Code to manage and import information to Django database.





3.5 **Flowchart**

In Brief: Supposing, we are in the localhost: http://127.0.0.1:8000/homepage. Django framework will find the views.home correspond to 'homepage/'. Next, inside the views.home, there is a function that renders home.html. Ideally, on that page, we click on a button which will redirect to 'aboutus/'. Now, the urls.py will be in charge of 'aboutus/', then continue the process again.





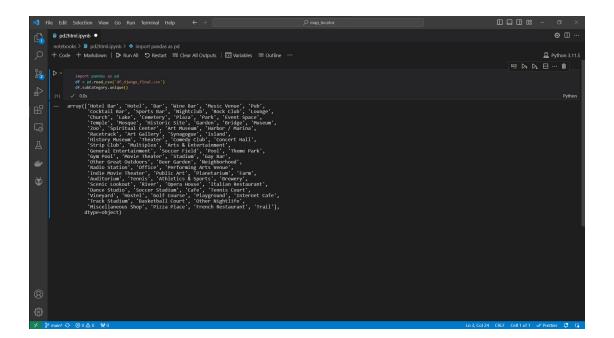
Deep Learning Models

4.1 Summary

In Brief: Firstly, I planned to use the DeepImageSearch library. With an input image, it will find the similar image from the datasets that I provide. After integrated it into the software, the input and output is not supported for Django and Web Development. Therefore, I decided to build my own deep learning model to find the comparable image.

4.2 Classification

In Brief: To find similar pictures, I need a large database and put images into that. Django framework and local host is not for that. Therefore, I came up with an classification model which will get an input image, then return a label. With that label, I can retrieve from the Category of database, the website ends up with a list of recommended places that matches users' interests.





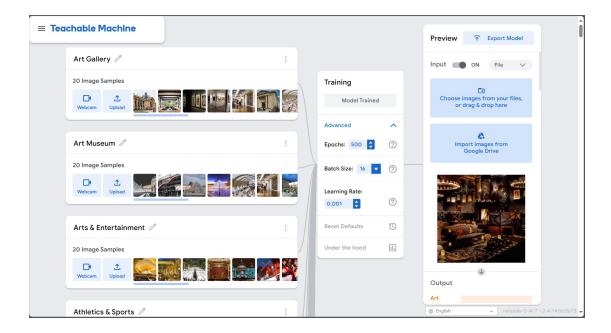
4.3 **Dataset Collecting**

In Brief: With the list from dataset, I will be downloading 80-class images using the code below.

```
import pandas as pd
from bing_image_downloader import downloader
3 import os
4 df = pd.read_csv('df_django_final.csv')
5 df.subCategory.unique()
6 for each in df.subCategory.unique():
     paris = ' in Paris'
     name = each + paris
     downloader.download(name, limit=20, output_dir=os.path.join('.', 'paris'))
```

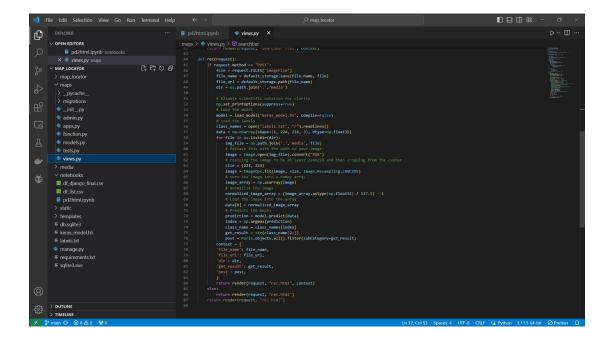
4.4 Training

In Brief: I am going to use the dataset above to train an Image Classification Model by using Teachable Machine. I tried 500 Epochs, Batch Size of 16 and Learning Rate of 0.001.



4.5 Export Models

In Brief: After training, I export the keras_model.h5 and labels.txt. Next, by using it for the code below in views.py, Django is able handle the request from users and return the classification label. Finally, I will retrieve the same Category places with that label, and showing all of that on the website.

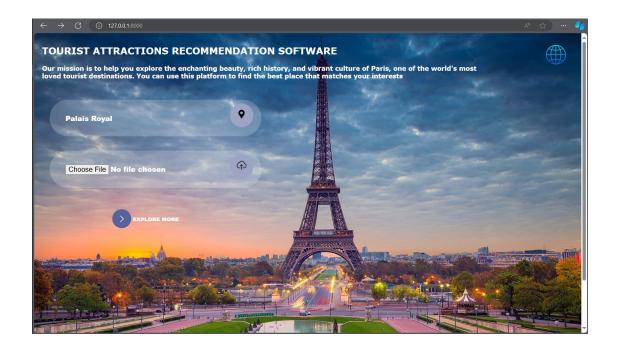




Implementation

5.1 Home Page

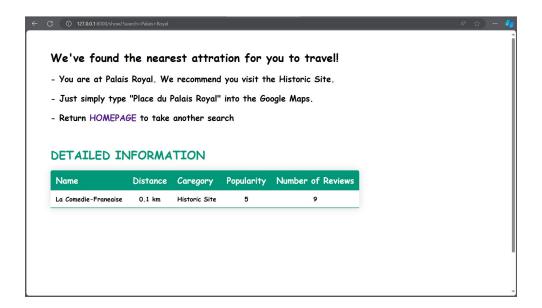
In Brief: The home page with three functions: Search by text, search by image and explore top 50 places in Paris.





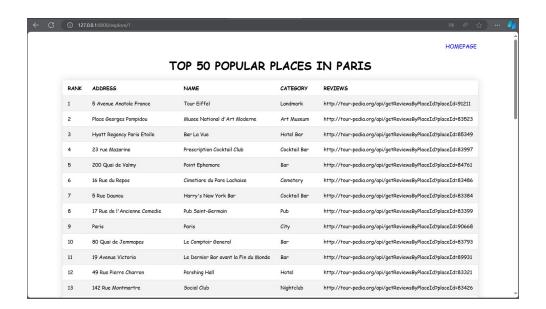
5.2Searching

In Brief: The result for Palais Royal search.



Exploring 5.3

In Brief: Users can read and user the name to find the nearest attrations.

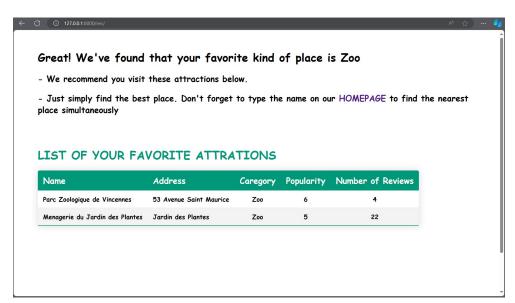




5.4 Image Searching

In Brief: With an input picture, the software will return the same caregory that matches users' interests.





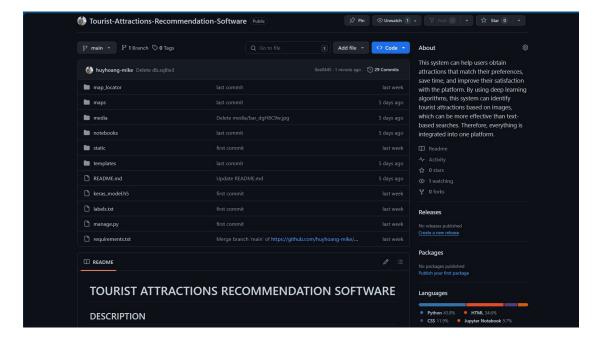


6 Summary

6.1 Github

I created a Github respository where you can see all the line of codes easily. There is also a Readme.md file and a detailed video how it works.

https://github.com/huyhoang-mike/Tourist-Attractions-Recommendation-Software



6.2 Conclusion

In conclusion, the project undertaken for the Applied Data Science with Python course at Ostbayerische Technische Hochschule Regensburg has been successful. I has demonstrated a comprehensive understanding of both frontend and backend development using Django, HTML, CSS, and SQLite. The deep learning model developed for image classification using Teachable Machine has been effectively integrated into the software, enabling it to recommend tourist attractions in Paris based on image inputs.

The software's features, including text and image search, exploration of top 50 places in Paris, and image-based recommendations, have been implemented and demonstrated effectively. This project showcases the technical skills and the ability to apply data science concepts to real-world applications, which I achieved by the Applied Data Science with Python course. Future work could involve expanding the software's capabilities to include other cities and improving the accuracy of the image classification model.



6.3 Sources

- \bullet https://docs.djangoproject.com/en/5.0/
- $\bullet \ \, \rm https://www.tensorflow.org/tutorials/images/classification$
- $\bullet \ \, \rm https://pypi.org/project/bing-image-downloader/$
- http://tour-pedia.org/api/index.html
- https://www.w3schools.com/Css/
- $\bullet \ \, {\rm https://www.sqlite.org/docs.html}$