



Project *Proposal*

**Project title: Image meta datasheet
Using python.**

Year : 2022-2026

Month : October 2024

Guided by:

Dr V Jean Shilpa

Associate Professor

Department of ECE

Done by : Madhan
RRN. : 220051601051

Dep: Election and communication
engineering
ECE- 'A'

Abstract:

This Python script performs image metadata extraction and report generation by analyzing a directory of image files. The script utilizes the Python Imaging Library (PIL) to read metadata from JPEG images, including dimensions (width and height) and creation date. It processes each image file in the specified directory, collects relevant metadata, and compiles this information into a structured CSV report. The CSV file includes columns for filename, width, height, and creation date, providing a comprehensive summary of the image properties. This automated approach simplifies metadata analysis, aids in managing large image collections, and generates an accessible report for further analysis or documentation purposes. The script ensures efficient handling of image data and facilitates easy integration into image management workflow

Implementation Platform:



PyCharm

PYTHON CODE

```
from datetime import datetime
import os
import tkinter as tk
from tkinter import filedialog, messagebox, Toplevel
from PIL import Image
from PIL.ExifTags import TAGS

def select_image():
    # Open a file dialog to select an image
    file_path = filedialog.askopenfilename(
        title="Select an Image",
        filetypes=[("Image Files", "*.*")]
    )

    if file_path:
        show_metadata(file_path)

def show_metadata(file_path):
    # Create a new window to display metadata
    metadata_window = Toplevel(root)
    metadata_window.title("Image Metadata")

    try:
        image = Image.open(file_path)
        width, height = image.size
        exif_data = image._getexif() or {}

        # Attempt to retrieve the DateTimeOriginal tag
        creation_date = exif_data.get(36867) # DateTimeOriginal tag
        if not creation_date:
            # Fallback to file modification time if DateTimeOriginal is not available
            creation_date = datetime.fromtimestamp(os.path.getmtime(file_path)).strftime('%Y-%m-%d %H:%M:%S')

        metadata_window.label = tk.Label(metadata_window, text=f"File Path: {file_path}\nWidth: {width}\nHeight: {height}\nExif Data: {exif_data}\nCreation Date: {creation_date}")
        metadata_window.label.pack()

    except Exception as e:
        metadata_window.messagebox = tk.messagebox.showerror("Error", f"An error occurred while processing the file: {e}")

root = tk.Tk()
root.title("Image Metadata Reader")
root.geometry("400x400")
root.mainloop()
```

```
# Prepare metadata to display
    format_info = image.format if image.format else
"Unknown format"
    filter_info = image.mode # Image mode as the "filter"
    metadata_text = f"Image Format: {format_info}\n"
    metadata_text += f"Image Dimensions: {width} x {height}
pixels\n"
    metadata_text += f"Date and Time: {creation_date}\n\n"
    metadata_text += f"Filter (Mode): {filter_info}\n\n"

for tag_id, value in exif_data.items():
    tag_name = TAGS.get(tag_id, tag_id)
    metadata_text += f'{tag_name}: {value}\n'

# Display metadata in a label
    label = tk.Label(metadata_window, text=metadata_text,
justify="left")
    label.pack(padx=10, pady=10)
except Exception as e:
    messagebox.showerror("Error", f"Could not retrieve
metadata: {e}")

# Set up the main application window
root = tk.Tk()
root.title("Image Metadata Viewer")

# Add a button to select an image
select_button = tk.Button(root, text="Select Image",
command=select_image)
select_button.pack(pady=50)

# Run the application
root.mainloop()
```

Function of the code:

1. Select_image() function:

- This function opens a file dialog to allow the user to select an image file.
- After selecting the image, it calls the `show_metadata()` function to display the metadata.

2. show_metadata(file_path)` function:

- This function takes the selected image file path and opens it using Pillow (`Image.open`).
- It retrieves metadata such as image format, dimensions, and EXIF data.
- It also handles a fallback mechanism: if the EXIF tag `DateTimeOriginal` (the image creation time) is unavailable, it retrieves the file's last modification time from the file system.
- All metadata (format, dimensions, EXIF tags) is displayed in a new window using a `tkinter.Label`.

3. Main Application ('root'):

- The main tkinter window ('root') is created.
- A button labeled "Select Image" is added, which calls the `select_image()` function when clicked.
- The GUI runs using `root.mainloop()`, which starts the tkinter event loop.

Specifications:

Select Image: Allows the user to pick an image file from their file system.

Extract Metadata: Extracts and displays image information (format, dimensions, creation time, and EXIF metadata).

Error Handling: If the metadata retrieval fails (e.g., invalid image file), an error message is displayed.

Library Used:

1. Datetime(Standard Library): Used to handle date and time, particularly for formatting timestamps.

2. OS (Standard Library): Used for file system operations, specifically to get the modification time of the selected image file.

3. tkinter(Standard Library): A Python library for creating graphical user interfaces (GUIs). It provides tools like:

- ‘filedialog’: To open a file selection dialog for the user to choose an image.
- ‘messagebox’: To display error messages if any issue occurs while retrieving the metadata.

– ‘Toplevel’: To create new windows (used here to display image metadata).

– ‘Label’, ‘Button’: GUI components to display text and clickable buttons.

4. PIL(Pillow): A third-party library used for image processing in Python. In the code:

– Image: Used to open the selected image file and retrieve its format, dimensions, and EXIF metadata.

– ExifTags: Used to map EXIF tag IDs to human-readable names (e.g., for DateTimeOriginal).

Outcome

The code extracts and displays the following

Image metadata:

- 1. Image Format:** The type of image (e.g., JPEG, PNG).
- 2. Image Dimensions:** Width and height in pixels.
- 3. Creation Date:** The original date the image was taken (from EXIF data) or the file's last modification date if unavailable.
- 4. Filter (Mode):** The color mode of the image (e.g., RGB, CMYK).
- 5. EXIF Metadata:** Additional information from the EXIF data, such as camera model, exposure time, ISO settings, etc., if available.

OUTPUT:

Select image



—

— **Image...**

Select Image

Image Format: PNG
Image Dimensions: 580 x 580 pixels
Date and Time: 2024-10-18 12:41:38
Filter (Mode): RGBA

Git hub link:

<https://github.com/220051601051/cloning-of-xy-academy-using-html.git>

