

Image Detection Algorithm Task for et3

Name: Abdelaziz Gamal Edin Ahmed

GitHub: https://github.com/AZ20100/Imaged_Detection_Task/tree/main/Image%20Detection%20Algorithm

```
import json
import shutil
from PIL.ExifTags import TAGS
import os
import re
from datetime import datetime
import csv
from PIL import Image
```

At the beginning of the code, I imported the needed libraries for the algorithm.

```
##### functions |
def remove_prefix(image_name):
    match = re.search(r'^.*?-', image_name)
    if match:
        return image_name[match.end():] # Return the part of the image name after the hyphen
    return image_name # Return the original name if no match

def get_modification_type(image_path, original_width, original_height):
    with Image.open(image_path) as img:
        width, height = img.size
        if width == original_height and height == original_width:
            return "Orientation Change"
        elif width / height != original_width / original_height:
            return "Cropped"
        else:
            return "Other"
```

Then I created the necessary functions to search for the images in folders and sub-folders, and Also to modify the images.

```
#####
# extract the images and copy them to a one folder

src_foldr = "E:\\courses\\et3\\dairies"
targ_foldr = "E:\\courses\\et3\\image_dataset"

for folder, sub_folder, files in os.walk(src_foldr):
    for file in files:
        if file.endswith(".jpg"):
            print(file)
            filename = os.path.join(src_foldr, folder, file)
            if os.path.exists(filename):
                print(filename)
                shutil.copy(filename, targ_foldr)
print(len(os.listdir(targ_foldr)))
```

Then the next part is to extract the images from the folders and sub-folders and copy them to another folder. And here u need to replace the path of your **dairies (src_folder)** folder which contains the folders and sub-folders which by the end contains the images. And replace the target folder (**targ_folder**) which is going to contain the images as a copy.

```
#####
# for prefix
|
# Directory containing image files
# List all files in the directory
image_files = os.listdir(targ_foldr)

for image_name in image_files:
    new_image_name = remove_prefix(image_name)
    print(f"Original Name: {image_name} | New Name: {new_image_name}")
```

After that the prefix section which modifies the name of the images which we already copied it into a new folder, and then list it as a comparison between the original name and the new name.

```

# for modification and report
# CSV report file path
csv_report_path = 'E:\\courses\\et3\\image_dataset\\image_report.csv'
# Original dimensions (Replace these with the original image dimensions before any modification)
original_width = 1920
original_height = 1080

# List all files in the directory
image_files = os.listdir(targ_folder)

# Create or overwrite the CSV file
with open(csv_report_path, 'w', newline='') as csvfile:
    csvwriter = csv.writer(csvfile)

    # Write header row
    csvwriter.writerow(['Image Name', 'Size (bytes)', 'Last Modification Date', 'Modification Type'])

    # Loop through each file in the directory
    for image_name in image_files:
        # Construct the full image path
        image_path = os.path.join(targ_folder, image_name)

        # Check if the current file is an image
        if image_path.lower().endswith(('.png', '.jpg', '.jpeg', '.gif', '.bmp')):
            # Extract image details
            clean_image_name = remove_prefix(image_name)
            image_size = os.path.getsize(image_path)
            timestamp = os.path.getmtime(image_path)
            last_modification = datetime.fromtimestamp(timestamp).strftime('%Y-%m-%d %H:%M:%S')

            # Get modification type
            modification_type = get_modification_type(image_path, original_width, original_height)
            # Write the details to the CSV
            csvwriter.writerow([clean_image_name, image_size, last_modification, modification_type])
            print(f"Image Name: {image_name}")
            print(f"Image Size: {image_size} bytes")
            print(f"Last Modification Date: {last_modification}")
            print("-----")

print(f"Report generated at {csv_report_path}")

```

Then the last part is to extract the name, size, and the last modification date for the image and then export them as a csv file containing the same data plus the type of modification happened on the image.

Here you also need to replace the csv path where you want to extract the csv file in your device.