import dashscope

import os

import warnings

import csv

from dotenv import load\_dotenv

from dashscope import MultiModalConversation

import re

# OSS Bucket details

bucket\_name = "YOURBUCKETNAME"

folder\_path = "um"

region = "oss-ap-southeast-3" #Replace this with your region endpoint

# Suppress warnings

warnings.filterwarnings("ignore", category=UserWarning, module='urllib3')

# Load environment variables

load\_dotenv()

# Set API key and base URL

dashscope.api\_key = os.getenv("api\_key")

dashscope.base\_http\_api\_url = 'https://dashscope-intl.aliyuncs.com/api/v1'

number\_of\_images = 5 #Total number of books in bucket

def generate\_image\_urls(number):

base\_url = f"https://{bucket\_name}.{region}.aliyuncs.com/{folder\_path}/"

return [f"{base\_url}book{i}.jpg" for i in range(1, number + 1)]

def get\_book\_info\_from\_image(image\_url):

messages = [

{

"role": "user",

"content": [

{"image": image\_url},

{"text": "Please provide the book name, author name, and publisher in CSV format."}

]

}

]

try:

response = MultiModalConversation.call(model='qwen-vl-plus', messages=messages, stream=False)

if response and 'output' in response and 'choices' in response['output']:

book\_info = response['output']['choices'][0]['message']['content'][0]['text']

return parse\_book\_info(book\_info)

else:

return ["Unknown", "Unknown", "Unknown"]

except Exception as e:

return ["Error", "Error", "Error"]

def parse\_book\_info(book\_info):

pattern1 = r'"?The name of the book is "(.\*?)"\."? It was written by (.\*?)\, published by (.\*?)(?:\.)?$'

pattern2 = r'"?The name of the book is "(.\*?)"\."? The author is (.\*?) and it was published by (.\*?)(?:\.)?$'

match = re.search(pattern1, book\_info)

if match:

return [match.group(1).strip(), match.group(2).strip(), match.group(3).strip()]

match = re.search(pattern2, book\_info)

if match:

return [match.group(1).strip(), match.group(2).strip(), match.group(3).strip()]

return [book\_info]

def save\_to\_csv(data, filename='book\_info.csv'):

with open(filename, mode='w', newline='', encoding='utf-8') as file:

writer = csv.writer(file)

writer.writerow(['Book Name', 'Author Name', 'Publisher'])

writer.writerows(data)

def main():

image\_urls = generate\_image\_urls(number\_of\_images)

all\_book\_info = []

for image\_url in image\_urls:

book\_info = get\_book\_info\_from\_image(image\_url)

all\_book\_info.append(book\_info)

save\_to\_csv(all\_book\_info)

print(f"Book information saved to book\_info.csv")

if \_\_name\_\_ == '\_\_main\_\_':

main()