Title:

Module -2 Report

Name:

Afolami Timothy Olawale

Date:

28-04-2024

Objectives

Task 4: OCR Functionality Implementation

Objective: Develop the capability to extract text from images using OCR technology.
Key Actions: Integrate and configure an OCR tool (e.g., Tesseract).

Task 5: Web Scraping for Product Images

Objective: Scrape product images from e-commerce websites for training data CNN_Model_Train_Data.csv.
Key Actions: Automate scraping, download images, and store them systematically and make sure you have enough data to train the CNN model.

Endpoint 2: OCR-Based Query Processing

Functionality: Extract and process handwritten queries using the same logic as Endpoint 1.
Input: Image file with handwritten text.
Output: Same output format as Endpoint 1, adapted for image inputs.

Implementation Flow

1. OCR

    1. Used Python EasyOCR library.
    2. Created a function that takes in an image path and returns the text found in it.

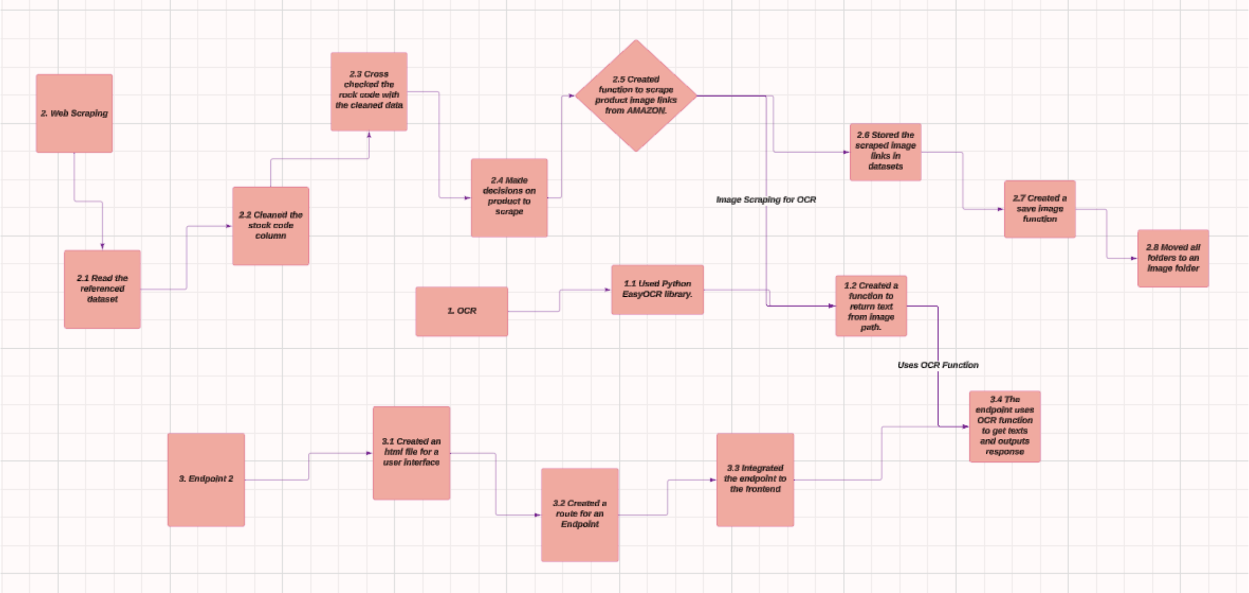
2. Web Scraping

    1. Read the referenced dataset
    2. Cleaned the stock code column
    3. Cross checked the rock code with the previous cleaned data
    4. Made some decisions on the product to scrape
    5. Created a python function to scrap product image links from AMAZON.
    6. Stored the scraped image links in respective datasets. One dataset per product.
    7. created a save image function that automatically creates a folder and saved the product images
    8. Moved all the folders in an Image folder

3. Endpoint 2.

    1. Created an html file for a user interface
    2. Created a route for an Endpoint
    3. Integrated the endpoint to the frontend
    4. The endpoint uses the function created earlier to get the texts in the image, then it sends back to the query reviewer and outputs response functions created in module 1.

Flowcharts


2. Web Scraping
2.1 Read the referenced dataset
2.2 Cleaned the stock code column
2.3 Cross checked the rock code with the cleaned data
2.4 Made decisions on product to scrape
2.5 Created function to scrape product image links from AMAZON.
2.6 Stored the scraped image links in datasets
2.7 Created a save image function
2.8 Moved all folders to an image folder
Image Scraping for OCR
1. OCR
1.1 Used Python EasyOCR library.
1.2 Created a function to return text from image path.
Uses OCR Function
3. Endpoint 2
3.1 Created an html file for a user interface
3.2 Created a route for an Endpoint
3.3 Integrated the endpoint to the frontend
3.4 The endpoint uses OCR function to get texts and outputs response

Challenges and Solution:

1. The products from the stock codes are five. Some has a general name while other don't. It's
difficult to
retrieve significant products from ecommerce websites like amazon and ebay. So I created a
list of products to work with.
This list of products are used for the image link scraping process.
2. EasyOCR performance is bad when it comes to hand writing recognition. A solution is to
train a deep learning
model that will be able to do this correctly.
3. The endpoint integration kept giving an error when image is uploaded. So i created a
separate method to handle
the image upload. I ensured that the enpoint is working effectively as expected.

Conclusion:

Though the ocr performance is bad, but we do have a very good stsyem that handles image input of handwriting and returns the expected product and response.

Reference:

EasyOCR
module - 1
Beautiful soup
Pandas
requests