

Title:
Module -2 Report

Name:
Afolami Timothy Olawale

Date:
28-04-2024

Objectives

Task 6: CNN Model Training

Objective: Develop a CNN model from scratch using only the products mentioned on CNN_Model_Train_Data.csv to identify products from images.

Key Actions: Train the model using scraped images and clean data without using pre-trained models.

Endpoint 3: Image-Based Product Detection

Functionality: Use the CNN model to identify products from images and match them using the vector database.

Input: Product image.

Output: Product description and matching products in a format consistent with other endpoints.

Implementation Flow

1. Model:

1. Created two models. AlexNet and ResNet18 from scratch
2. Load my image data from the folder they are stored in. I have eight image folders in total.

3. Performed some transformation and augmentation. Resized the image to 90*90 (this is most efficient at the moment)
4. Trained both models using Pytorch and Cuda (to speed the process)
5. ResNet18 performed better than AlexNet.
6. Saved the ResNet18 model into the memory.
7. Created an inference file function that takes in just the model weights and the image path, preprocesses the model and then makes predictions with the model.
8. Created separate files for each of these processes.

2. Endpoint:

1. Created an html file for frontend
2. Created an endpoint for the model inference.
3. Integrated other helper functions like get_price, get_stock_code and get_country.
4. Images come from the frontend page, sent to the endpoint, the model inference makes a prediction and sends the predicted class name.
5. The class name is passed as a query to the product search function, and returns the top 4 products that are similar to it.
6. Get the response (comment for the products), stock code, prices and country.
7. These are returned as output for the endpoint. Presented as a table on the frontend.

Challenges and Solution:

1. The model performance is not perfect, using pretrained model weights will help improve this performance.
2. The endpoint integration kept giving an error when the image was uploaded. So i created a separate method to handle the image upload. I ensured that the endpoint is working effectively as expected.

Conclusion:

The model performance is 70% accuracy on the test set, and we do have a very good stsyem that handles images and returns the expected product, response, and other information.

Reference:

Pytorch

module - 1

module - 2