Assignment 3 – Implementing Visual Search

Digital Marketing Analytics

Summer 2020

Prof. Sri Krishnamurthy

Northeastern University & QuantUniversity

Implementing Similarity search at scale

Goals

- To work with datasets using xsv
- Implement Visual Search algorithms
- Implement Similarity Search
- To be able to implement search with Elastic Search
- Evaluate design decisions
- Crisply communicate and document your findings

Case

QU analytics has hired you to as an Algorithmic marketing analyst. QU is a consulting organization specializing in marketing analytical solutions. Your client is a large e-tailer (CouchSmart) who has millions of products in its catalog. They intend to enhance the user-experience of their clientele by providing rich and engaging interfaces without leaving their couches! They are considering implementing Visual Search and has reached out to QU Analytics assist in prototyping such a solution.

Since the data is still being collected, they intend to use Cdistance's data as a proxy. (https://www.kaggle.com/c/cdiscount-image-classification-challenge

). They intend to try out a couple of different approaches and get recommendations on which approach to implement.

The specification of the project is as follows:

Ingestion and pre-processing:

1. Download the dataset and process it using the sample code provided[1]
2. Sample data using xsv[5] so you can prototype. Get at least 100 categories and 100 products in each category

Similarity Search:

Preprocess the images as needed by each method. You can have 3 separate files.

1. Implement Version 1 of Similarity search using method proposed in [3]
2. Implement Version 2 of Similarity search using the Facebook method proposed in [6]. See [2] for how they used it.
3. Implement Version 3 of Similarity search using the Spotify-Annoy method proposed in [4]

Methods:

You should implement 2 methods:

1. Single lookup: Given an image identifier, retrieve k-similar images
2. Bulk: Generate a json with k-similar images for each image

See [4] and [7] for examples:

Search:

1. Install and configure Elasticsearch
2. Using the output from the Bulk output Json, index the data so you can query Elasticsearch for k similar images

Reference app:

1. Build a reference app, (simple app using streamlit or flask[8]) to enable searches.. See [9] for ideas
2. Your reference app have the 2 modes:

a. When an indexed image (from a dropdown or randomly shown on the site) is selected, you return back k images from the elastic search index.

b. Provide an interface so you can upload a new image, this will call the function to lookup k images similar to the uploaded image and returns back the k images similar to the new image.. (You can put some samples on S3/Google drive and use those as links to point to the images)

3. Run the ref application on Heroku or AWS/Google

Deliverables:

- Full code on github
- Ref app hosted on AWS/Heroku/ any platform of your choice.
- A Google Codelabs document summarizing the insights
- Assignment is due July 10th 9.00pm. You will be presenting in the class on July 11th

Reference:

1. https://www.kaggle.com/inversion/processing-bson-files
2. https://medium.com/gsi-technology/integrating-textual-and-visual-information-into-a-powerful-visual-search-engine-c477486a18ff
3. https://github.com/ikatsov/tensor-house/blob/master/search/image-artistic-style-similarity.ipynb
4. https://towardsdatascience.com/image-similarity-detection-in-action-with-tensorflow-2-0-b8d9a78b2509
5. https://github.com/BurntSushi/xsv
6. https://engineering.fb.com/data-infrastructure/faiss-a-library-for-efficient-similarity-search/
7. https://github.com/facebookresearch/faiss/wiki/Getting-started
8. https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-xvi-full-text-search
9. https://github.com/gyang274/visual-search