Image Search Engine

Viraj Pawar
Dublin City University
Dublin, Ireland
viraj.pawar2@email.dcu.ie

ABSTRACT

In this assignment, an image search engine is implemented. The assignment is implemented in two parts: 1. Web crawling and indexing images, and 2. Inverted Index of textual surrogate of images and Retrieval of images using front-end UI. A web crawler is implemented to collect images and their textual surrogates from a publicly accessible image archive. A total of 1907 images were downloaded and annotated using YOLO - object detection algorithm [1]. The annotation from YOLO (YOLO3) is added to the textual surrogate to augment the textual description of the images. The augmented textual description of the images, image source URLs, and the image name is mapped to an image ID and stored in a dictionary. An inverted index of the textual surrogate is created to perform text-based searches that return relevant images in a UI. Vector Space Model (VSM) is used to perform image search. The front-end UI is implemented using streamlit python package and deployed at the following link https://testassignment-7gvdxvytrtxmrqqwito2ze.streamlit.app/.

KEYWORDS

VSM, TF-IDF, YOLO, streamlit

1 INTRODUCTION

Text-based image search retrieval systems use text as a query to search relevant images. A text-based image search retrieval system utilizes textual information associated with images to perform image search operations. In a text-based search retrieval system, annotation of images is required to retrieve images or similar images based on the text search query. In contrast, a content-based image retrieval system relies on image semantics for image search. The challenges involved in text-based image search are: 1. word ambiguity and context of the text description

- `Ambiguity and context`: In search query, words can have different meaning. for example, searching for an image of an apple fruit instead of an Apple logo. The textual description of the images may not be accurate or fully capture the visual content of an image.
- `Quality of metadata`: While annotating images using Machine learning algorithms is not completely accurate, manual annotation of images is time-consuming
- `scale and efficiency`: Maintaining performance of large amount of indexed images.

1.1 Architecture

The architecture of the implementation of the image search engine consists of two parts: 1. a web crawler and annotation; 2. Image retrieval.

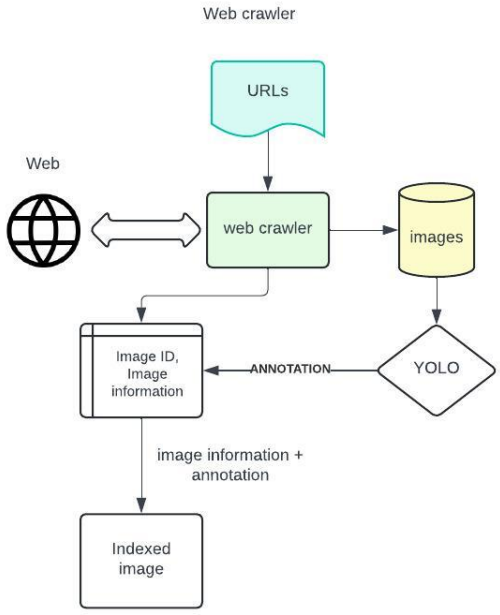

Figure 1: Web crawler

1.1.1 Web Crawler. The images were crawled from a publicly available image archive https://openphoto.net. Python package **mechanicalsoup** was used to crawl images. Mechanicalsoup https://mechanicalsoup.readthedocs.io/en/stable/ is an open-source Python package used for automating interaction with websites. The textual surrogate of the images extracted using the web crawler are 'KEYWORDS' which describes each image on the URL and the photographer's name. To enhance textual surrogate, image annotation was done using You Only Look Once (YOLO3) image annotation technique. YOLO is an object detection algorithm which predicts multiple bounding boxes and class probabilities for those boxes

simultaneously using a single convolutional neural network (CNN) [1]. YOLO annotation is used in the implementation of the image search engine to enhance textual surrogate by detecting objects in the images, which is later indexed to improve search engine performance in when a query is performed.

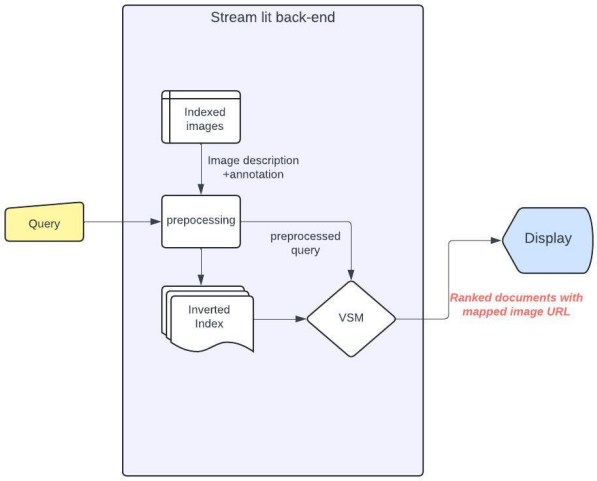

Figure 2: Image retrieval

1.1.2 Image retrieval. The above figure shows the architecture of the image retrieval part. The image retrieval part consists of text pre-processing of textual surrogates of images, inverted indexing of textual surrogates, query text pre-processing, a Vector Space Model (VSM) for ranking the documents, and front-end UI.

2 ANNOTATION

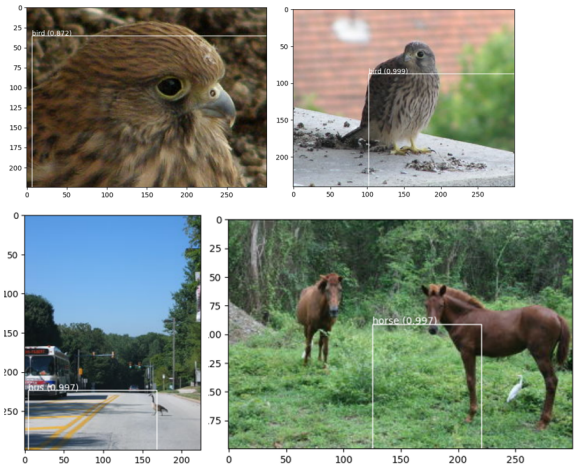

Figure 3: Image annotated using YOLO3

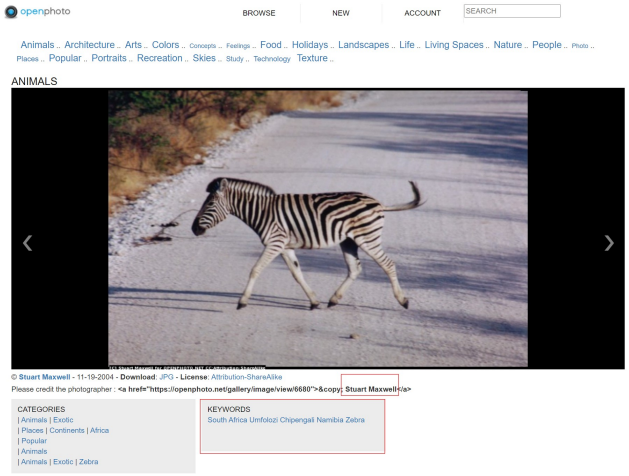

Figure 4: Image information highlighted in red used as textual surrogate

Using the web crawler, text information of the image is scraped from 'KEYWORDS', and information of the photographer is added to the text information. Figure 4 shows text description of the image as keywords. Image annotation is performed using YOLO3 object detection algorithm. Figure 3 shows annotated images from YOLO3. Code from https://github.com/patrick013/Object-Detection---Yolov3/tree/master was utilised to create annotations. The annotation from YOLO is used to enhance the textual surrogate. The annotation of each image is added to the image's textual details as a text surrogate. The object detection is limited to a small list of labels and produces false positives and misses some objects in the image.

3 INDEXING

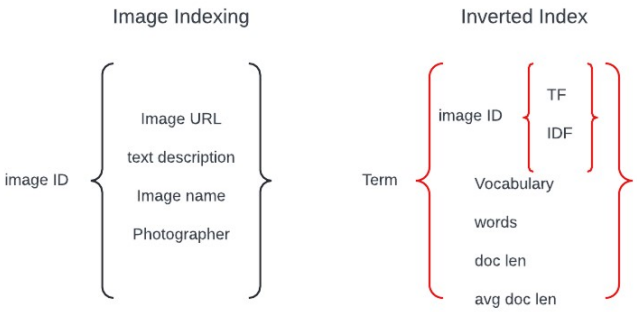

Figure 5: Index images and inverted index

The images are indexed by creating a dictionary containing textual information about the image, image URL, image name and photographer. The image text information is pre-processed. Tokenisation, stop word removal, text normalization, and converting

numbers to text are performed on the text description of the images. Words specific to the image description, e.g. "stock", "openphoto.meEgret", "download", etc., which add noise to the text data, are also removed.

An inverted index for the text description of the images is created using the InvertedIndex class from assignment 1.
Figure 5 shows indexed images and the inverted index structure.

4 RETRIEVAL

A web app is created using streamlit https://streamlit.io/. The indexed images consist of a dictionary with image ID as key and lists of image URL, text description, image name and photographer as values. An inverted index of the text description is created in the retrieval process. The query text is pre-processed, and the VSM model is used to rank the image ID. Based on the rank, the most relevant URLs are called, and the images are produced for those URLs.

Figure 6 shows the front end of the image retrieval system. When a search query is entered in the text box, the most relevant image with its metadata is displayed.

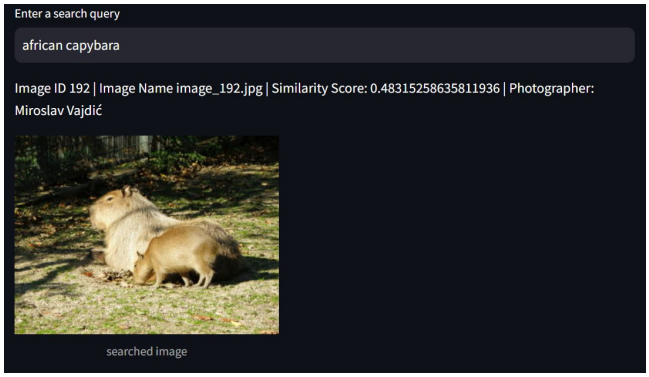

Figure 6: Font-end of image search engine

The Github link to the code: https://github.com/virajpwr/assignment_2.git

REFERENCES

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788.