Building and Exploring UFO Sightings using Data Science: Creating UFO Data Insights

Soravis Taekasem
taekasem@usc.edu

Surasit Prakunhungsit
prakunhu@usc.edu

Theerapat Chawannakul
tchawann@usc.edu

Yifan Wu
ywu352@usc.edu

Abstract

In this assignment, we explore the UFO sighting data using several visualization techniques to analyze and produce some helpful insights. Moreover, we ingest the UFO sighting data to Apache Solr to display a dynamic visualization on the web. Also, we use ImageCat and our custom scripts to use Apache Tika to extract metadata and provide OCR to get the image content into Apache Solr. Finally, we connect ImageSpace to Solr, and use FLANN plugin to find similar images and search the image forensics and OCR. In addition, we integrate ImageSpace with ElasticSearch and try some other plugins such as SMQTK and VideoSpace.

1 Introduction

Data visualization allows us to visually access to huge amounts of data in easily digestible visuals. Well designed data graphics are usually the simplest and at the same time, the most powerful. We carefully pick and design 11 visualizations using D3 JavaScript library including one dynamic visualization. Also, we explore search engine capability with Lucene Solr. We study and implement image search engine with image similarity using several tools including ImageCat and ImageSpace.

2 Data Visualization

To create a visualization, we analyze the UFO data by creating several scripts with Python and convert and summarize result into a compact JSON file to be used as an input for JavaScript visualization code. We pick 11 different visualizations to present some distinct insights. See some visualization in Figure 1 and 2.

2.1 Calendar heatmap of UFO sightings

For this visualization we select a calendar representation to display the UFO sighting density of each day during 1990 to 1999. We choose this visualization because calendar heatmap is a great tool to visualize each day in a year. We can clearly see the density on a particular day. Moreover, this is a dynamic visualization view user can click and see the detail on that day. Please note that for this visualization to work with Solr server, you need to configure the SOLR_URL in ./team13/html/html/calendar-view.html to point to Apache Solr server (including core name) before use.

2.2 UFO sightings density

For this visualization, we select a choropleth map to display the UFO sighting density of each county in the United States. We choose this visualization because choropleth map is an excellent tool for present a density of each area on the map. See Figure 1.

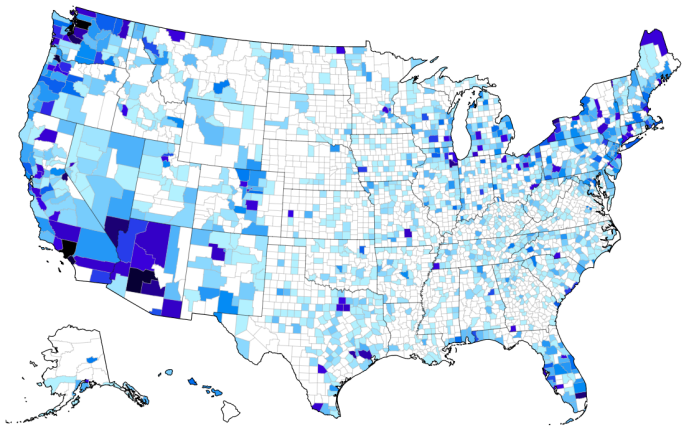

Figure 1: Choropleth Map

2.3 Distance from airport

For this visualization, we select a bar chart to display the UFO sighting distance from the reported area to the airports. We choose this visualization because a bar chart can easily show the different of each range from their height, which is a perfect visualization to see the trend and easy for comparing.

2.4 Distance from military base

For this visualization, we select a bar chart to display the UFO sighting distance from the reported area to the the military bases. We choose this visualization because a bar chart can easily show the different of each range from their height, which is a perfect visualization to see the trend and easy for comparing.

2.5 Weather condition during sightings

For this visualization, we select a pie chart to display the weather condition during sightings. We choose this visualization because a pie chart can clearly present the portion between each section in percentage. We include the percentages of snowy, foggy, windy, and rainy weather condition.

2.6 Sightings histogram

For this visualization, we select a histogram to display the amount of sightings in the areas with different population density. We choose this visualization because a histogram can clearly present the probability distribution of a continuous variable, so we get a better view on how number of sightings are affected by population density.

2.7 Scatterplot

For this visualization, we select a scatterplot to display the amount of sightings in each year. We choose this visualization because a scatterplot can clearly present the changes of data in a period of time. It?s easy to see how number of sightings change when time passes.

2.8 Sighting frequency

For this visualization, we select a line chart to display the amount of sightings in different areas in each year. We choose this visualization because a line chart is not only present how sightings in one area change in each year, but also compare sightings in different areas at the same time.

2.9 Sightings by region

For this visualization, we select a grouped bar chart to display the sum of number of sightings in different areas in the period of 10 years. We choose this visualization because the grouped bar chart can give us an overview of an amount of sightings in different areas in a fairly long time, which lead to a better conclusion about how sightings affected by areas.

2.10 Shape sightings of region

For this visualization, we select a radar chart to display the distribution of sighted UFO shapes among different areas. We choose this visualization because the radar chart gives us the information about where some UFO shapes are more common and what shape is more frequent.

2.11 Word cloud

For this visualization, we select a word cloud to display the amount of sightings with different NER fields, which are mostly places. We choose this visualization because the word cloud can give us an instinct and impressing view of the data, where larger words mean larger frequencies. See Figure 2.

3 Image Search Engine

3.1 Prerequisite

- Download and install Docker and Docker-Compose that will be used for managing different software containers and images
- Git clone ImageSpace, The ImageSpace github includes the docker-compose.yml file that can be used later for installing Apache Solr docker, ImageSpace docker, and MongoDB docker
- Install tangelo and other dependencies

3.2 Apache Solr

Apache Solr is a search engine platform built on top of Apache Lucene. In this assignment, we use it to index our UFO data from TSV v2 dataset and the image contents for using by ImageSpace.

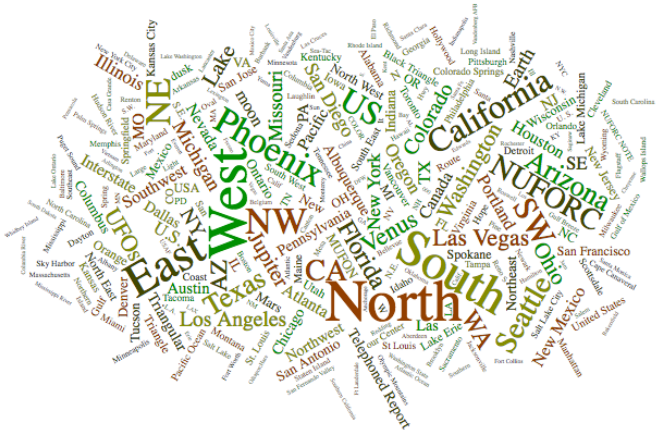

Figure 2: NER Location Word Cloud

3.3 ElasticSearch

ElasticSearch is a distributed, JSON-based search and analytics engine. In this assignment, we try to modify some ImageSpace REST server script and configurations to make ElasticSearch accessible to ImageSpace.

3.4 ImageCat

ImageCat is an Apache OODT RADIX application that uses Apache Solr, Apache Tika and Apache OODT. In this assignment, we use ImageCat to extract Apache Tika metadata from the UFO stalker images and provide OCR of the images to get content which could be indexed by Apache Solr.

3.5 ImageSpace

ImageSpace is an application built on top of ImageCat and a plugin of Girder. It allows histogram and D3-based visual search, free text search and retrieval, and performs image similarity metrics using computer vision techniques and metadata-techniques. Figure 3 shows a search request in ImageSpace to the Apache Solr server. The result shows images which contain some similarity with the query word.

3.6 MongoDB

MongoDB is a free and open-source cross-platform document-oriented NoSQL database. It is needed for Girder, a free and open source web-based data management platform.

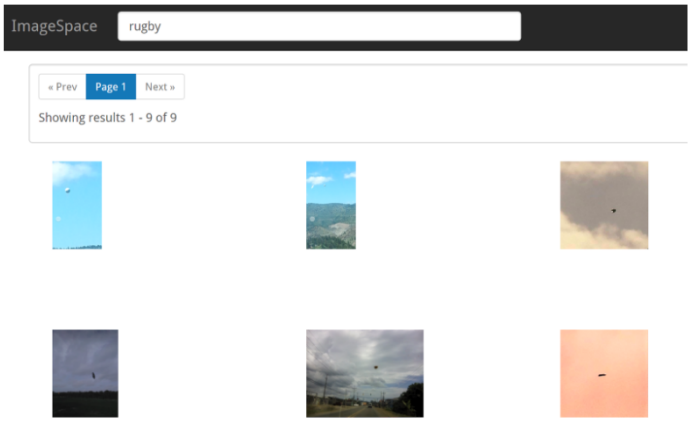

Figure 3: ImageSpace Query

3.7 Docker and Docker-Compose

Docker is a computer software that performs operating-system-level virtualization of images and containers. The process is known as containerization. In this assignment, we run our program inside docker containers to make things faster and cleaner. Docker-Compose combines multiple docker images together and make them easier to manage.

3.8 Implementation

- Convert our TSV v2 to JSON format.
- After we downloaded all required file in the prerequisite section, we can run all docker containers including the ImageSpace-Girder docker, Solr docker, MongoDB docker, ImageCat docker, ElasticSearch docker, PostgresDB and SMQTK server, by navigating to "image_space/scripts/deploy" folder. In that folder, we edit the docker-compose.yml file to match our preference. The original file contains MongoDB, Solr, and ImageSpace. We add and edited the Dockerfile so that it will automatically install and enable required library and plugins such as (Tangelo, OpenCV, flann_index, imagespace_smqtk).

  We also add the following code to install ElasticSearch in the docker networks,

```
elasticsearch:
  image: docker.elastic.co/elasticsearch/
    elasticsearch:6.2.4
  container_name: elasticsearch
  environment:
    - cluster.name=docker-cluster
    - bootstrap.memory_lock=true
    - "ES_JAVA_OPTS=-Xms512m -Xmx512m"
  ulimits:
  memlock:
    soft: -1
    hard: -1
  volumes:
    - esdata1:/usr/share/elasticsearch/data
  ports:
    - 9200:9200
  networks:
    - imagespace-network

elasticsearch2:
  image: docker.elastic.co/elasticsearch/
    elasticsearch:6.2.4
  container_name: elasticsearch2
  environment:
    - cluster.name=docker-cluster
    - bootstrap.memory_lock=true
    - "ES_JAVA_OPTS=-Xms512m -Xmx512m"
    - "discovery.zen.ping.unicast.hosts=elasticsearch"
  ulimits:
  memlock:
    soft: -1
    hard: -1
  volumes:
    - esdata2:/usr/share/elasticsearch/data
  networks:
    - imagespace-network
```

- We then run docker-compose up to initialize and run all images.
- Now that we have all services running, we then feed the TSV v2 JSON file to Apache Solr server. After Solr interpret the file, it then generate the non image index for us.
- Feed UFO Images to the ImageCat that used Tesseract OCR and Apache Tika to generate Object Information from the images. This information are then feed into the Apache Solr server.
- Then we configure the ImageSpace Environment attribute to point to the Solr server installed by the ImageCat docker so we can retrieve the Object Information.
- Run the import-images.sh shell script in "image_space/scripts/deploy" folder to feed images into Solr server.
- Now Solr should contains the images index.
- Install FLANN index by editing the listing.txt file in "image_space/index/" folder with full image paths.
- Install extra plugin SMQTK.
- Test running the ImageSpace by going to "localhost:8989", then try to search with any terms.

4 Extra Credit 1: ImageSpace with ElasticSearch

In the extra credit 1, first, we add the ElasticSearch to our docker-compose.yml file. Then, we add data to it for indexing using the CURL command. In the case of the ElasticSearch, before we add the data sets, we need to set up mappings for the fields. They divides the documents in the index into logical groups and specifies a field's characteristics.

Below is the sample mapping that map certain fields such as speaker as a keyword and line_id field as an integer.

```
curl -X PUT "localhost:9200/shakespeare"
  -H 'Content-Type: application/json' -d
'{"mappings": {
  "doc": {
  "properties": {
  "speaker": {"type": "keyword"},
  "play_name": {"type": "keyword"},
  "line_id": {"type": "integer"},
  "speech_number": {"type": "integer"}}}}}'
```

Now that we finished mapping, we can add the datasets to the ElasticSearch database for indexing. Below is the command to add the dataset,

```
curl -H 'Content-Type: application/x-ndjson' -XPOST
'localhost:9200/index_name/doc/_bulk?pretty'
  --data-binary @Filename
```

Now that we have the datasets with indexes, we test the ElasticSearch Search API in the ImageSpace server as follow,

- Use the command docker ps to list all the docker processes, find the image_space image and keep its container_id
- Then use the command docker exec -it CONTAINER_ID /bin/bash to get into the image_space server bash
- Check the IP address of ElasticSearch server using docker inspect CONTAINER_ID command where CONTAINER_ID is the ElasticSearch container_id
- Verify the connection between ImageSpace and ElasticSearch server using the following command, "curl IPADDRESS:9200/_cat/health". Where IPADDRESS is the ElasticSearch IP address from the last step. If both server run and connect correctly. You should see the health status of the ElasticSearch.
- Try to run CURL to test the ElasticSearch Search API as follows, Then replace the QUERY_STRING with the search words and the API will return a list of 10 results or you can add the SIZE parameter to get more results.

```
curl -X GET "IPADDRESS:9200/_search?q=QUERY_STRING"
  -H 'Content-Type: application/json' -d' { "query":
  { "match_all": {} } }'
```

- We also tried to modified the image_space source files to be able to call the API directly through the webpage but have not yet reached the final state.

5 Extra Credit 2: ImageSpace Plugins

For this section, we did try both FLANN index and SMQTK. For FLANN index, we installed the images paths to the listing.txt file in "image_space/index/" folder and get it to worked. By looking at the source code, the FLANN index plugin uses image histogram extracted from imagefeatures_rest.py to find nearest neighbours. So, the images with same amount of highlight and shadow are similar to each other.

For SMQTK, it first trains the models using CNN (Convolutional Neural Network) method named AlexNet. Then, using that model, it finds similar images from SMQTK server and use Nearest Neighbours method on field sha1sum_s_md to retrieve list of images. Top images we got by using SMQTK are very similar to each other as the position and shape of an object are nearly identical.

6 Questions & Answers

Why did you select your 10 D3 visualizations? How are they answering and showing off your features from assignments 1 and 2 and the work you did?

Well designed data graphics are usually the simplest and at the same time, the most powerful. We carefully pick and design 11 visualizations using D3 JavaScript library including one dynamic visualization. Each visualization have its own strength and downfall, such as choropleth can clearly show the density of each county on a map, or a bar chart can easily display the different between each component (See more detail in section 2). We implement each visualization using HTML, CSS with JavaScript and read the summarize data from JSON files. All script and data are in team13 directory.

Did Image Space allow you to find any similarity between UFO sightings images that previously was not easily discernible based on the text captions and object identifications you did?

Yes, with human observation, the ImageSpace allow us to find the results we needed faster. For images, human identifies strange object that likely to be an UFO more efficiency than machines. By using ImageSpace with SMQTK plugin, we can detect similarity between images which can give us a head start instead of previous methods. To illustrate, we can using an image of a flying object with a clear sky to find similar images. The result are most likely to be in same category. This is different from previous assignment which identified an image as some strange objects like nematode worm.

Our thoughts on the ImageSpace and the ImageCat?

- Easy to use
  - Using Docker and docker-compose to start containers (both ImageSpace, ImageCat)
  - Provided installation scripts
  - ImageCat can extract images? features quite fast (In our opinion)
  - ImageSpace UI is simple and easy to navigate
- Not easy to use
  - Some docker files are not up to date, thus we have to add and update multiple files in order to run them
  - Not ready to use library and source code (FLANN, OpenCV2 in imagefeatures_rest.py)
  - No admin page in ImageSpace to enable/disable plugins
  - ImageCat uses Apache Solr 4 but ImageSpace uses Apache Solr 7. Some scripts were not compatible. As a result, we have to modified the docker code and point both ImageCat and ImageSpace to the same Solr server

7 Conclusion

Data visualization enables decision makers to see analytics presented visually, so they can grasp difficult concepts or identify new patterns. With interactive visualization, you can take the concept a step further by using technology to drill down into charts and graphs for more detail, interactively changing what data you see and how it?s processed. In this assignment, we explore several data visualization techniques and also take a look at some aspect of the current search engine.