INTRODUCTION

Data is the new oil. In this project, we were given scanned PDFs that could be turned into a rich source of UFO data. By leveraging Imagemagick, Ghostscript, Tesseract, Tika-vision, and several open-sourced Named Entity Recognition software, we produced a dataset comprised of OCR'ed documents and auto-captioned images.

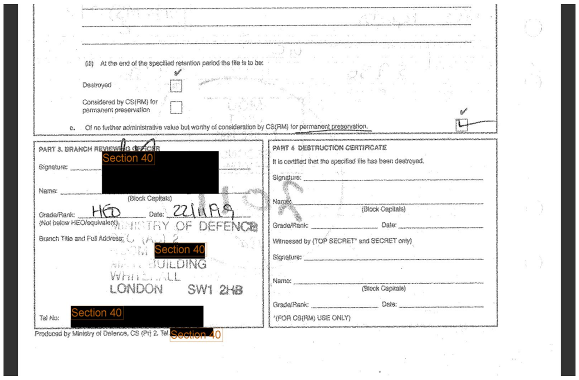
IN-DEPTH EXPLANATION OF THE PROCESS

OCR-PIPELINE:

Tesseract is the tool used to perform OCR on a document, but for it to output meaningful text our group attempted to make the input as machine readable as possible. This was done by attempting all suggestions found in the tesseract improve quality page.

1) **Rescaling** - Tesseract works better when input images have a DPI of 300
2) Binarisation - Since tesseract does this internally by default, it's best to preprocess the page in such a way that binarization doesn't make parts of the page unreadable
3) **Noise Removal** - Many scanned PDFs have dots or lines that make it tough for Tesseract to process that particular region - removing this noise would improve output. This was done by compressing the image (pixelated it) and then sharpening the image (smooths out rough edges) -- used Ghostscript. Also made use of -median filter in Imagemagick
4) Automatic rotation - a very tough task that we did not implement because there is no one-size-fits-all command that generalizes to all scanned PDFs
5) Border removal - On individual test pages, improved Tesseract output significantly but could not generalize to all PDFs so left out

We tried several other techniques - local threshold, gaussian blur, automatic detection of rotation, automatic text block detection, and parameter tuning for Imagemagick and Ghostscript but found that a simple blur, sharpen, and median filter generalizes best to our scanned PDFs. Note that individual PDFs could be enhanced significantly using these commands if done manually.



(iii) At the end of the specified retention period the file is to be:

Destroyed ☑

Considered by CS(RM) for permanent preservation □

c. Of no further administrative value but worthy of consideration by CS(RM) for permanent preservation. ☑

PART 3. BRANCH REVIEWING OFFICER

Signature: Section 40 [redacted]

Name: (Block Capitals)

Grade/Rank: HEO Date: 22/11/09
(Not below HEO/equivalent)

Branch Title and Full Address: MINISTRY OF DEFENCE
C (AD) 2
ROOM Section 40 [redacted]
MAIN BUILDING
WHITEHALL
LONDON SW1 2HB

Tel No: Section 40 [redacted]

PART 4 DESTRUCTION CERTIFICATE

It is certified that the specified file has been destroyed.

Signature:

Name: (Block Capitals)

Grade/Rank: Date:

Witnessed by (TOP SECRET* and SECRET only)

Signature:

Name: (Block Capitals)

Grade/Rank: Date:

*(FOR CS(RM) USE ONLY)

Produced by Ministry of Defence, CS (Pr) 2. Tel. Section 40 [redacted]

We also used Fred's ImageMagick script: Textcleaner and attempted to tune parameters for our dataset but finally did not use this tool. However, it is worth noting that we were able to get rid of **Section 40** that appears in pdf using the following command "convert -density 300 5.pdf -depth 8 -background white -flatten +matte -sigmoidal-contrast 10,50% tiff/5.tiff"

We also tuned Tesseract parameters, eventually using automatic page segmentation (not the default) and **trained Tesseract on new fonts**- typewriter and cursive fonts that would potentially help Tesseract pick up fonts similar to those found in the PDFs.

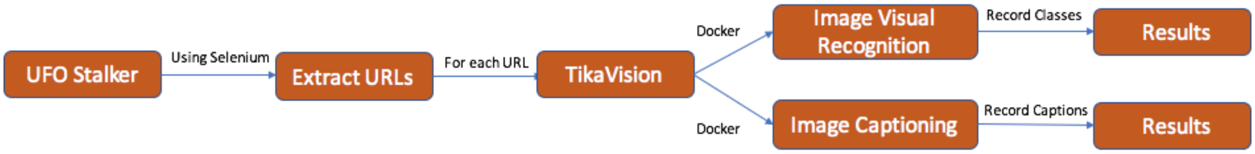
TIKA NER:

NER for date, duration, location: Leveraged Core-NLP local server to extract date, duration, and location entities from text/description. Used NEs to chunk and assign to appropriate features.

Extra Credit: Explored named entity recognition(NER) on the extracted text after running through the OCR pipeline. To achieve this we downloaded the binaries for Apache OpenNLP and Stanford Core-NLP tools and ran it through the **tika command line utility** (tikaCLI). We wrote shell scripts to loop through each text file and executed tika cli command to record the results (NER) for each file.

IMAGE-CAPTIONING:

Things needed for captioning: Tika vision and Docker. The building and activation of the Docker plays a key role, as it allows for generating captions. From the UFO-stalker website we extract the urls using selenium and then by using Tika Vision and Image Captioning we retrieve the results for the entries.



Finally, we merge the csvs resulted from the Tika Vision and Image Captioning processes.

TENSORFLOW:

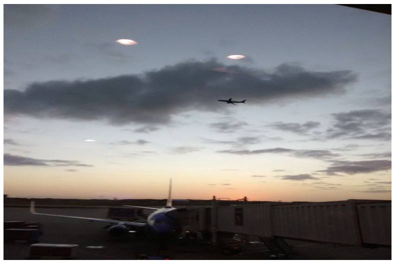
TensorFlow is an open-source software library which is being used for various tasks involving tensors, mainly Deep Learning tasks. Generally, Deep Learning tasks need a lot of data and also requires lot of time for training. Here, we would like to leverage the power of Deep Learning as we have scraped about 3k images of UFO sightings and see how it can classify new UFO sightings. But instead of training it from scratch, we would try **transfer learning** and retrain an existing model. We use a model already trained on ImageNet dataset. We modify the last layer with our custom classes and train it on UFO images dataset scraped from ufostalker. The steps for this are mentioned here .
To get good results, we need a lot of images per class. So, instead of training it on many classes, we decided to train on the classes for which we had maximum images. So, based on the classes given by TikaVision, we pick 3 classes with maximum images, which are: spotlight, roundworm, nematode worm.
After retraining, it generates image_label and output_labels.

QUESTIONS AND ANSWERS:

Q1.What did you notice about the dataset as you completed the tasks?
Ans. The UFO sightings reported, like any other real-world dataset, was very messy. The PDFs had light vertical lines, border, typed content, cursive handwriting, skewed table, text was very blur and not legible and words had shadow. Sometimes, there was overlapping text. Because of these general characteristics, Tesseract outputted unreadable, meaningless text. As detailed above, we explored various things that can improve the performance of Tesseract. We also attempted to train tesseract on new fonts that appeared in our pdfs. All these helped improve the quality of results obtained through tesseract for OCR.
The other dataset was images of UFO sightings. We scraped images from ufostalker website. Apart from images, we got gifs, docs, video files.We used TikaVision to process the images obtained and thus obtain captions and also classnames for a given image.

Q2. What questions did your new joined datasets allow you to answer about the UFO sightings previously unanswered?
Ans. The newly joined datasets helps us in giving a better interpretation of the data entries of the British UFO sightings. In assignment 1, the truthfulness of the data could not be verified as the dataset source was not a government issued, whereas the British UFO Sightings data was provided by the government and thus the authenticity of the recorded information is there, but the valid data entries of the sightings cannot be verified as they still come from the general public. The assurity of the dataset entries was formed in this assignment. Though the conclusion about the UFO sightings being true is still the same as assignment 1 i.e. valid explanation can be provided for most of the entries of not being a UFO.

Q3. How well did the image captions accurately describe the UFO object types? What about the identified objects in the image?
Ans. The image captioning performances on the overall data set was not accurate. The captioning results were not relevant with the UFO image sightings and the summary. For example: In the data summary - “A silver/ red object came out of nowhere when viewing star” but the caption generated was “a black and white photo of a person holding a soccer ball” . We infer that the captioning done is not accurately.
There are instances where the caption recorded describes the surrounding objects. But the number of correct outcomes is very low if compared to the entire dataset. For example: In this given image, we can witness the two prominent light spots in the sky near the airport. But the caption generated was: “a large jetliner flying in the sky”, which is partially correct.



After seeing the results of TikaVision, we tried to use deep learning approach for this task. Deep neural networks trained on large dataset of UFO sightings will give better results.

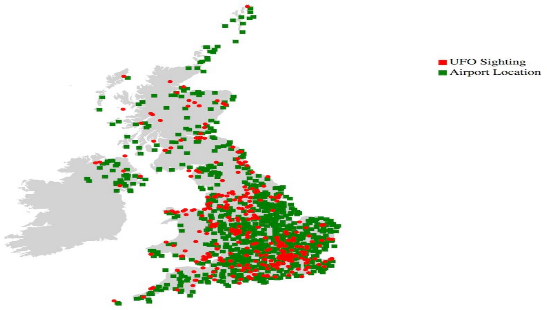
Q4.How well did OCR work?
Ans. We can get good results for OCR if the text in documents are typed text. OCR using Tesseract could not pick up anything when the text was handwritten or cursive handwriting- as Tesseract is trained on only machine fonts. Also, many pages in pdfs were blur, black vertical lines, skewed tables, bad handwritten font, etc.
Workaround: We used different filters like. -median to lighten vertical lines, -negate to increase the sharpness of page by flipping black and white colors. These commands in imagemagick, and playing with different parameters helped us understand when does tesseract give good results. Refer to top of report for in-depth discussion about OCR process.

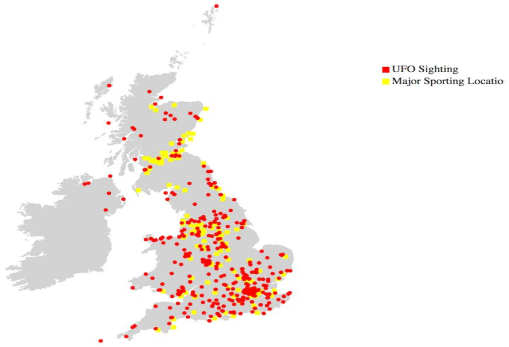
Q5.What did you have to do to clean up the noise in the data?
Ans. We followed the guidelines provided on the tesseract-ocr github repo **.** This document lists OCR best practices. This gave us the idea that we can train OCR on new fonts which are similar to the ones in pdf files. Again refer above to OCR-Pipeline in depth discussion. In short, we tried several Imagemagick and Ghostscript commands and even tried to use OpenCV.

Q6.Of the incorporated British UFO sightings, how many of them could also similarly be explained akin to the sightings from the first assignment?
Ans. In the assignment one, we had introduced many new features which helped in much clearer interpretation of the sightings. These features were as follows: closest distance to airport, closest sports metro distance name, rural or urban area, population of the state.
The inference can be made similarly with the British UFO Sightings data.

1. We inferred that sightings within 25 miles of the airport are due to the airplanes.
2. There are some major league sports team in UK. The locations as to where these teams are located, we found that the population of those areas are relatively higher as compared to the other areas and thus have higher number of sightings reported. Inferring that urban areas have higher ratio of sightings reports as compared to the rural areas.
3. The number of British UFO sightings also varied state wise depending upon the overall population of that state.

These interpretations are depicted in the plotted images.



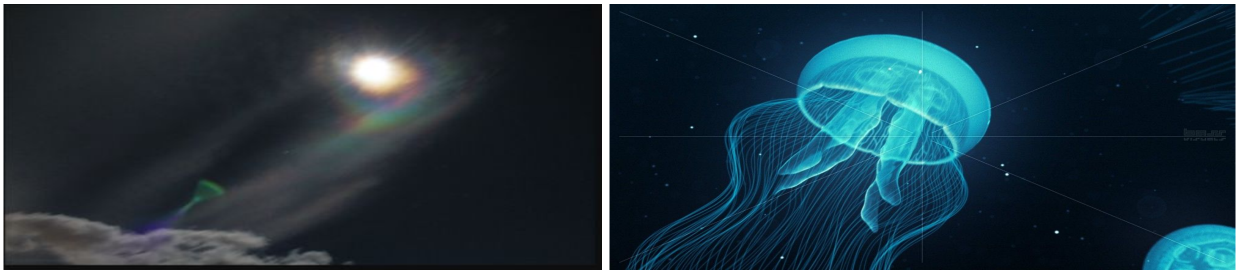


Q7.Were there any new object types introduced by the British UFO sightings?
Yes. These are the following new types we found- Mist, spire, aerial, steeples, aircraft, flashing, line, angular, saucer, comet, stars, dome, moon, box, shining, hull, ring, pulsating and halo. Refer to /NER/word_frequency.csv for a word frequencies for all non-stop words. Many of the objects were described as ‘object’ - refer to the *Statistics* section below for a bar chart of word frequencies.

Q8.How well were the British UFO sightings described? Was there a lot of missing data?
Ans. Overall, the Pdf data which had the British UFO sightings was not consistent as there was too much noise.. The pdf files had irrelevant representations like vertical lines, handwritten information, skewed tables, not clearly visible information, section-40 [black areas] etc . The subtopics of the data were not consistent. Some of the data had unreadable handwritten entries.

After extracting the information from the PDF files, we inferred that the overall resulting dataset was very sparse. Out of all the sightings , nearly ⅔ had descriptions generated by Tesseract, of which only approximately 500 had interpretable entries.
NOTE: In a clever hack, we attempted to identify location of sighting through NER but if no “NER_LOCATION” was found but a “NER_ORGANIZATION” was found we checked if that organization is located in the UK (using geopy) and if it did gave the sighting location as the location of the organization. E.g. ‘RAF’ was used frequently in descriptions. In many entries containing ‘RAF’ NER could not tag a location, but using the ‘RAF’ location entity we hypothesized that the sighting occurred near the *Royal Air Force* base in UK.

Q9.Of the UFO images, how many of the images actually generated image captions and/or objects that described the UFO and not just the background scenery?
Ans. The image caption was done on the given data which had inconclusive outcome. Later the captioning was done on the data which was trained by ImageNet dataset by using Tensor Flow. It gave us a better percentage of outcomes although they are not the correct captions.
For example: The first image is the UFO recording as per the dataset. The summary noted was: “Caught in a photo of the full moon. i could not see it with the naked eye. but when i was looking at photos. there was this strange green spaceship looking thing”. But the caption generated was “a close up of a bird flying in the air” and it gave a “Jellyfish” class as outcome. The class was recorded as it can be seen by the second image , looks a lot like the actual jelly-fish.



a) UFO Image Capture

b) Jelly Fish

We looked at a sample of 30 data entries and manually verified as to what the image captions have been generated with the summary provided. The ratio of relevant to irrelevant captions capturing was very low. We can conclude that the image captions recorded were not relevant to the UFO image and the summary.

THOUGHTS ON OCR PIPELINING

The given ocr-pipeline worked well, but we made some optimizations.
Pros: OCR reduced the time required for processing of the data from the pdfs - much faster than manually processing
Cons: It cannot recognize much of the hand written text and the percentage of identifying the exact text error is high. It is difficult to generalize on large amounts of PDFs.

IMAGE CAPTIONING/ OBJECT IDENTIFICATION

Part A : Extracting images:

1. We developed a python script to extract images using the selenium package. With our script we were able to extract urls for around 800-900 images.
2. Getting started with the pipeline was pretty straightforward, we ran into issues when we were getting timed out for making too many calls.

Part B : Class tagging and generating captions:

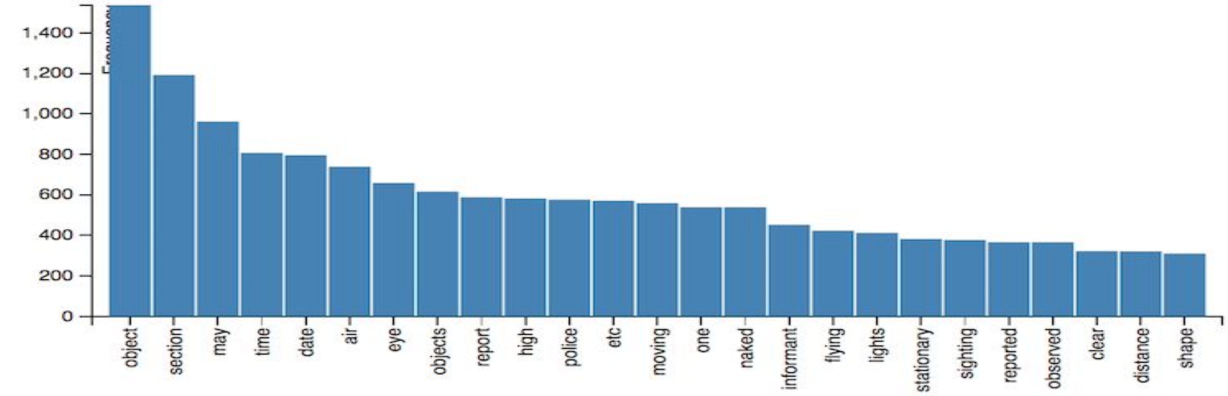
1. The dockers for Tika Vision and Tika Caption are well documented, we did not have trouble setting them up and using it.
2. Issues and scope for improvement.
    - The rest services take a really long time to respond when we provide a faulty url. And after throwing an error (HTTP 500), it doesn't recover quickly. We did put a work around in our

code. Filtered out urls which were of type "image" and only then did we send it to the rest servers.

- We are currently exploring the Python/Flask Code to handle unexpected cases and prevent the Flask server from going into a bad state.

STATISTICS

Here are some interesting statistics of the frequently occuring words in the dataset that we observed.



CONCLUSION

Information doesn’t have to come from only textual data sources. This project showed us that using images is a viable source of information that adds veracity and variety to our dataset. OCR and automatic image captioning are powerful processes increase velocity of data as we are able to ingest inputs faster - this in turns leads to a more valuable, versatile dataset.

TEAM MEMBERS:

- Khyati Ganatra
- Pavneet Kaur Mukar
- Prerana THM
- Sachin Kumar GB
- Sanjay Nadhavajhala