Chapter 1 : Tasks Accomplished and Scripts Developed

1. **OCR :** Extract meaningful text from scanned pdf files in british-ufo-dataset
    a. **separate-pdf.sh :** Script separates all pdf files in given input path/folder into separate page pdf files
    b. **pdftotext.sh :** Automates the generation of xyz_gs.tiff, xyz_im.tiff, xyz_gs.txt and xyz_im.txt using both GhostScript and ImageMagick for .tiff file generation and Tesseract for .txt file generation
    c. **extract_text.sh :** Runs through all extracted text files in outtxt/ and pools locations into files_list.txt
    d. **ocr-pipeline.sh** : automation of text extraction using tesseract on pdf files with just ImageMagick to observe different results
    e. **extract-text.sh:** Fetches data from each of the files in outtxt/ and creates output.json file with key - filepath and sub keys im (data collected from Imagemagick extraction) and gs (data collected from Ghostscript extractions)
    f. **extract.py :** Fetches data from each of the files in outtxt/ and creates output.json file with key - filepath and sub keys im (data collected from Imagemagick extraction) and gs (data collected from Ghostscript extractions)
    g. **extract-ocr-final.py :** This script reads the output.json file in the split-british-ufo-files folder in data (which contains the mapping of individual file extractions and their locations) extracts data from all the listed files and provides clean output for the parser in json format for each page of pdf files
2. **Text Parser:** Extract meaningful information from OCR text
    a. **clean_ocr.py** :Attempts to filter text in “im” and “gs” fields by removing unicode characters that are not printable, replace punctuation with spaces and remove multiple spaces.
    b. **pythonParser2.py** : Parses over the output of previous python program to extract named entities by using a ‘Named Entity Recognition Parser’ from texts and adds content (location, date, description)to British ufo files (as extracted by OCR). Also populates NER fields with Named Entities
3. **Scrapping ufostalker, Object detection and image captioning** : Scrape ufostalker.com and fetch image urls and relevant text like location, description, lat,long and use object detection (tika dockers) to populate UFO-v1 with more rows and relevant fields
    a. **ObjectRecognitionParser.java:** Added a block of code that appends list of objects or captions to a file
    b. **scrp.py** : Selenium crawler that hits events on ufostalker.py to extract image urls only. We used this initially to hit 9000 events and this took more than a day to complete due to blocking issues and had to be split amongst teammates
    c. **ufo_stalker_json.py** : Modified professor’s script to match our needs. Used ufostalker’s json api to fetch data and append results with relevant details(like lat, long, county, location etc) along with corresponding urls and zip codes.
    d. **filter-relevant-images.py** : Filters relevant images and compares total overall crawled images to results from scrapy.py and pics the intersection of urls
    e. **clean_ufotalker.py** : Takes object, captions and urls from the results of object detection and creates a mapping (into cleaned_ufostalker_content_urls askeys-2.json)
    f. **version2.2-ufo.py** : Takes the resultant image to caption and object mappings and populates the data fetched for each entry into version2.2 with location, description, object shape etc.
    g. **parser3.py** : This code runs the extracted descriptions from ufostalker on stanford NLP tagger

to fetch and populate named entities (NER) Parses the 'description' field from the 'object detection' output and applies Named Entity Recognition to recognise different entities and populate relevant fields.

   h. **get_images.py** : Downloads the images to a directory from the list of links given through a file
   i. **objects.py** : Performs object detection and image captioning for all the UFO images using Tika. Creates objects.txt and caption.txt file.
   j. **check_extension.py** : Cleans the URLs received from UFO site. Only png, jpeg and gif are considered.

4. **NER :** Ran the descriptions from v1 on 5 different Named Entity Recognizer packages with Tika app and analyzed results
   a. **CoreNLP.py**: Takes "description" field in v1 as the input and uses Tika app plus Stanford CoreNLP to extract name entities from it.
   b. **OpenNLP.py**: Takes "description" field in v1 as the input and uses Tika app plus OpenNLP to extract name entities from it.
   c. **MITIE.py**: Takes "description" field in v1 as the input and uses Tika app with MIT information to extract named entities from it.
   d. **NLTK.py**: Takes "description" field in v1 as the input and uses Tika app with natural language toolkit (a python library) to extract named entities from it.
   e. **Grobid.py**: Takes "description" field in v1 as the input and uses Tika app with Grobid Quantities (a JAVA library) to extract named entities from it.
      Each script generates a file storing the name entities parsed by each package.
   f. **integrate_datasets.py**: Takes the files generated by the scripts above as inputs, creates new fields in v2 ('NER_*').
      CoreNLP, OpenNLP, MITIE : NER_LOCATION, NER_DATE, NER_MONEY, NER_ORGANIZATION, NER_PERCENTAGE, NER_TIME, NER_PERSON
      NLTK: NER_NAMES
      GROBID: NER_MEASUREMENTS, NER_MEASUREMENT_NUMBERS, NER_MEASUREMENT_UNITS

5. **Retraining Last Layer for Image2Text** :
   Run retrain.py provided by Tensorflow according to the website:
   https://www.tensorflow.org/tutorials/image_retraining
   We used default hyperparameters to retrain the model.

6. **Merge datasets:** merge v1, british ufo sightings, ufostalker
   a. merge.py : merges v1-withNER.json, british-ufo-withNER.json, ufo_stalker_withNER.json to create v2.json (version 2)
      This merges 60095(v1-withNER) , 1732 (british-ufo-withNER) and 8563 (ufo_stalker_withNER) to produce 70390 (v2.json)
   b. jsonToTSV.py: converts v2.json to v2.tsv

Final dataset has 35 columns/keys:

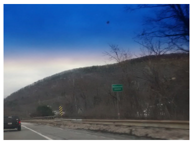
['CO Mean', 'NER_DATE', 'NER_LOCATION', 'NER_MEASUREMENTS', 'NER_MEASUREMENT_NUMBERS', 'NER_MEASUREMENT_UNITS', 'NER_MONEY', 'NER_NAMES', 'NER_NORMALIZED_MEASUREMENTS', 'NER_ORGANIZATION', 'NER_PERCENTAGE', 'NER_PERSON', 'NER_TIME', 'O3 Mean', 'SO2 Mean', 'airport_distance', 'airport_name', 'cancer_incidence_counts_allraces', 'cancer_incidence_counts_hispanic', 'cancer_incidence_counts_white', 'county', 'death rate', 'description', 'duration', 'image-captions', 'image-objects', 'image-url', 'latitude', 'location', 'longitude', 'population', 'reported_at', 'shape', 'sighted_at', 'zipcode'] image-* columns added from object detection and caption generation of ufo-stalker images.

NER_* columns added from NER performed using 5 different models.

Chapter 2 : Analysis

1. **What questions did your new joined datasets allow you to answer about the UFO sightings previously unanswered?**
   By adding new rows to the UFO v1 dataset we had more data to analyze UFO patterns. This was also supported with new Columns or Features in the sightings. The most important features were the objects detected from images. We discovered a few keywords (like nematode, spotlight etc - please see bar chart on next page) that indicated accurately the presence of orbs in the sky. This helped us understand the nature of the sightings as to how other features perform in the presence of these keywords in the object field. Adding Named Entity features to all the rows helped us analyse the authenticity of some sightings. This was particularly helpful in analyzing the British UFO data which had poor description. Given we applied the best techniques within our discretion to enhance OCR implementation, it still lacked enough data extraction from the scanned PDFs. Also, in places where no named entities were recognized or where descriptions were less than three words, we could conclude that these reportings added little value to the dataset.
2. **How well did the image captions accurately describe the UFO object types?**
   One would expect the image captioning model to predict captions like ‘an unidentified flying object spotted’ or ‘something strange flying in the sky’. The models being trained on general images had little data on UFO’s to predict. So we observed different factors to decide if a caption was accurate or not.
    1. The caption identifies the presence of an object in the sky (if image has one)
    2. The caption uses at least one of the objects detected from the image
    3. The caption closely describes the location or scenery of the image

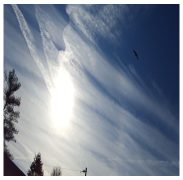
   Based on these three features as our assumptions, we observed that 71% of the captions were relevant to the images and could imply the presence of UFO objects. Examples include "a view of a city skyline from a plane", "a person flying through the air while riding a snowboard".
   Now, given we scrap our assumptions from before - the model tends to perform poorly in coordination with the object types. There were close to 21% image captions that accurately described anything close to the UFO object types. The following images and their captions show one of each results



a street sign on the side of a road [eng](confidence = 0.000289)

img_1167.jpeg

Inaccurate



a man flying through the air while riding a snowboard [eng](confidence = 0.009109)

img_1149.jpeg

Accurate

Considers flying and air, ignores man and snowboard

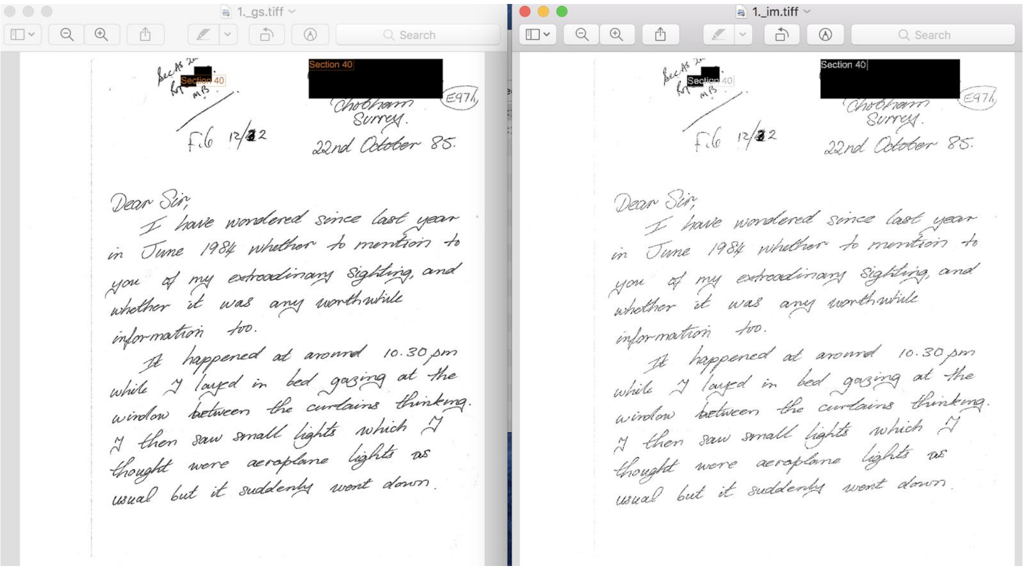
3. **What about the identified objects in the image?**
The object detection model performed with a high accuracy in detecting objects in the image. The images with "possible" UFO detection had similar set of words in their object list. These include **nematode, spotlight, jellyfish, balloon, parachute, roundworm** etc. We thus assume without any harm, that these words represent possible UFO sighting. Also, in other general images, the objects were detected with a good accuracy. For example,

img_2866.jpeg - [balloon, parachute, chute, airship, dirigible]

img_2756.jpeg - [flagpole, flagstaff, seashore, coast, seacoast, sea-coast, lakeside, lakeshore]

4. **How well did OCR work?**
OCR using Tesseract did not provide best results. We concluded that the reason was poor quality of scanned pdfs. A lot of the data was garbled and difficult to understand. We observed that pdf pages when converted to images using ImageMagick did not only lose some clarity but when applying OCR to the image made little progress in text extraction. However, applying effective data cleaning and noise reduction techniques boosted the OCR results upto 23%

5. **What did you have to do to clean up the noise in the data?**
We explored GhostScript as an alternative to ImageMagick and found out that using GhostScript to convert pdf to tiff files on images with white backgrounds was more productive as compared to ImageMagick, where as on images with black/dark backgrounds, the opposite was true. So we compare the results as we can see alongside where left is a conversion of PDF to TIFF using GhostScript(GS) and right is a conversion of the same PDF to TIFF using ImageMagick(IM). Note the clarity difference in the text on white background and the text on black background. This analysis supported our decision to developed a script - pdftotext.sh that would generate both tiffs, using IM and GS and run tesseract on the resultant TIFF files to enable comprehensive extraction of content from scanned PDFs. Doing this



Section 40 [redacted]
Section 40 [redacted]
MB
E97/1
Chobham
Surrey.
F.6 12/22
22nd October 85.

Dear Sir,
I have wondered since last year in June 1984 whether to mention to you of my extraodinary Sighting, and whether it was any worthwhile information too.
It happened at around 10.30 pm while I layed in bed gazing at the window between the curtains thinking. I then saw small lights which I thought were aeroplane lights as usual but it suddenly went down.

Section 40 [redacted]
Section 40 [redacted]
MB
E97/1
Chobham
Surrey.
F.6 12/22
22nd October 85.

Dear Sir,
I have wondered since last year in June 1984 whether to mention to you of my extraodinary Sighting, and whether it was any worthwhile information too.
It happened at around 10.30 pm while I layed in bed gazing at the window between the curtains thinking. I then saw small lights which I thought were aeroplane lights as usual but it suddenly went down.

allowed us to choose the best results from both tools and improved the number of accurate word extractions by 23%.

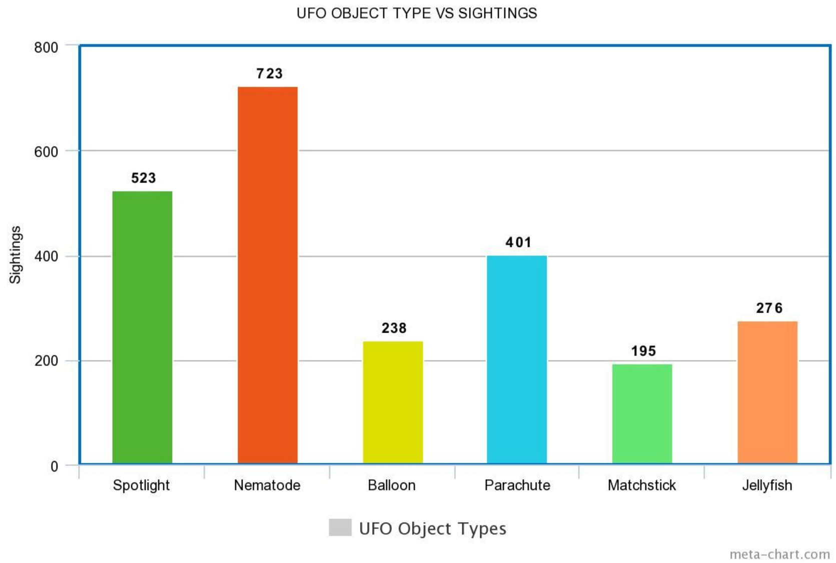
6. **Of the incorporated British UFO sightings, how many of them could also similarly be explained akin to the sightings from the first assignment?**
   1732 entries were similar out of a total of 1968 entries. We evaluate similarity of sightings based on the description extracted, and shape of objects detected. Having done NER on the dataset (both v1 and british ufo files) we could establish a better similarity measure between the two datasets by using the NER features (like NER_PERSON, NER_LOCATION and NER_ORGANIZATION)

7. **Were there any new object types introduced by the British UFO sightings?**
   Yes. “Bowl”, “lamps”, “roundworm”, “coin”, “globe” were some of the new shapes that were found.

8. **How well were the British UFO sightings described?**
   From manual reading of PDF files of the British UFO dataset, we observed that there was a variation in the details provided in descriptions. Where some reportings were perfectly described with accurate adjectives, others had only a few words to say. We observed that 44% of the scanned PDFs had enough information to add to description apart from location and other features. The only bottleneck was the poor quality of the scanned PDFs that hindered bringing all this valuable information into clean usable format. Further, we hoped to explore handwritten text extraction to improve the results (but could not explore due to lack of time)

9. **Was there a lot of missing data?**
   Indeed there was a lot of missing data related to the shape and latitude/longitude. Our parser could identify numbers but wasn’t intelligent enough to recognize the context of the number (whether the number was for population, duration, time)

10. **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?**
    We observed that close to 43% of the images (from ufo stalker) or ~3.6k images had relevant information regarding UFO objects. Of these, 37% also included descriptions about the scenery.



11. **Also include your thoughts about OCR pipelining, and Image Captioning/Object identification – what was easy about using it? What wasn't?**
    The best part about the OCR pipeline is it helped us understand the importance of automating scripts. This is the biggest advantage when dealing with big data. The OCR pipleine improved our clarity in the steps involved and how to make a valuable sequence of commands iterative to the requires solution. Further analysis and probing helped us understand the advantages of ImageMagick and GhostScript and this we unofficially shared with some other teams too. This helped us explore different flags and support provided by the different tools.
    Image Captioning/ Object identification was effective and produced best results of the two types of data extraction techniques applied in this assignment. The tools were easy to use and had clear documentation. We also observed that adding a new flag to the tika-app to process all images in bulk would solve the object detection pipelining.

Chapter 3 : Extra Credit Analysis

1. **Extract Credit 1 : NER**
    a. Comparing the performance of Tika with different NLP packages for NER:
        i. CoreNLP, OpenNLP, MITIE packages worked best to identify specific Named Entities like location, date, money, organization, time, person and percentage.
        ii. NLTK was best able to identify names only
        iii. Grobid Quantities recognized any expressions of measurements (e.g. pressure, temperature, etc.).
    b. Processing text files is time- and space- consuming, so we processed a subset of the v1 dataset. Particularly, CoreNLP and OpenNLP consumed extensive resources and thus we decided to apply NER for close to 8000 entries in v1 dataset. In data/Results/final/v1-withNER.json holds all our additions to v1. Further we used StanfordNLP tagger to tag the british ufo files and ufostalker entries with Named entities. (this proved to be faster than other packages, though could evaluate a limited set of entities and not a variety of them)
2. **Extra Credit 2: Retrain last layer of images for Image2Text.**
    a. We re-trained the last layer of the model using 4000 images from our ufo-stalker image collection.
    b. This was time intensive and we could not completely evaluate the performance of the re trained model and thus did not submit a pull request. However, we attach code herewith and hope to evaluate the model later and submit a pull request.

Chapter 4 : Conclusion

We wrote a TikaParser (code/object-detection-ufostalker/ObjectDetectionParser.java)for object detection and have attached the code herewith. This code modifies existing code to reroute tika's abilities for multiple image inputs. We hope to clean this code and make it more general and then submit a pull request to the github repository.

Further, we could not completely write a TikaParser for the British UFO files as we faced several issues with running Tika, and some of which were open issues as observed on the github issues page. Trying to fix these issues costed us an entire week with little progress and thus we decided to write a simple python parser that works with basic string matching to detect information and also uses advanced NLP techniques like Named Entity Recognition to populate features in out v2 dataset.

Log.txt file in code/ shows the logs of running ImageMagick and Ghostscript on a series of PDFs to record success or failure of the actions.