Project Elective

Choropleth Maps

Vineet Priyedarshi & Nilay Kamat

June - December 2023

Abstract

This project aimed to develop an automated classification and annotation system for images containing choropleth maps and discrete legends/colour bars. The dataset consisted of two types of map images: those with discrete color legends and others with a colourbar indicating statistical values. The workflow began with manual classification, separating the dataset partially into images with legends and images with colour bars. It then proceeded to annotation(only on the images with discrete legends), wherein the contents of each image were annotated based on the title, map and legend. We automated the annotation using machine learning. Subsequently, a classification model was trained to differentiate between the two types of maps.

The methodology involved preprocessing the images, extracting relevant features, and employing machine learning algorithms for classification. The model demonstrated robust performance, achieving high accuracy in distinguishing between maps with discrete color legends and those with colorbars. The results suggest that the proposed approach can effectively automate the classification of diverse map types.

Acknowledgements

We would like to express our sincere gratitude to Prof. Jaya Sreevalsan-Nair for her guidance and support throughout the completion of this project. Their contributions and assistance have been crucial to the success of this project.

A special thank you to Srinivas Manda, Sarthak Harne and Karanveer Singh, who helped us during this project. Their guidance and assistance helped us achieve success in this project.

Contents

Chapter 1

Introduction and Problem Statement

Geospatial data plays a pivotal role in diverse fields, from urban planning to environmental monitoring. In this context, maps serve as indispensable tools for conveying complex spatial information. Efficiently classifying and annotating maps, particularly distinguishing between those with discrete colour legends and those featuring colourbars, is crucial for automated geospatial data analysis.

The existing challenge centers around the manual and time-intensive methods traditionally employed for map classification. Manual annotation and data collection processes are prone to inefficiencies and potential inaccuracies, limiting the scalability and efficiency of map evaluation. This project addresses these challenges by introducing an automated solution that utilizes advanced techniques for map classification and annotation.

Key Aspects of the Problem Include:

1. Manual Processes: The reliance on manual annotation and classification methods introduces inefficiencies and the potential for inaccuracies in identifying distinct types of maps.
2. Data Volume: The increasing volume of geospatial data requires efficient processing methods to classify and annotate maps accurately.
3. Complex Map Dynamics: Maps exhibit intricate characteristics influenced by factors such as legend types and colour representations. Accurate classification and annotation are essential for effective geospatial data interpretation.

The objective of this project is to design and implement an automated solution for the classification and annotation of maps, specifically focusing on distinguishing between maps with discrete color legends and those featuring colorbars. By automating these processes, the project aims to enhance the efficiency and accuracy of geospatial data analysis across various applications and industries.

This project also focuses on leveraging choropleth map image processing techniques to identify and analyze the data that was used to create the images. By automating the classification, annotation and extraction of relevant features from the choropleth map images, valuable insights can be made on the data.

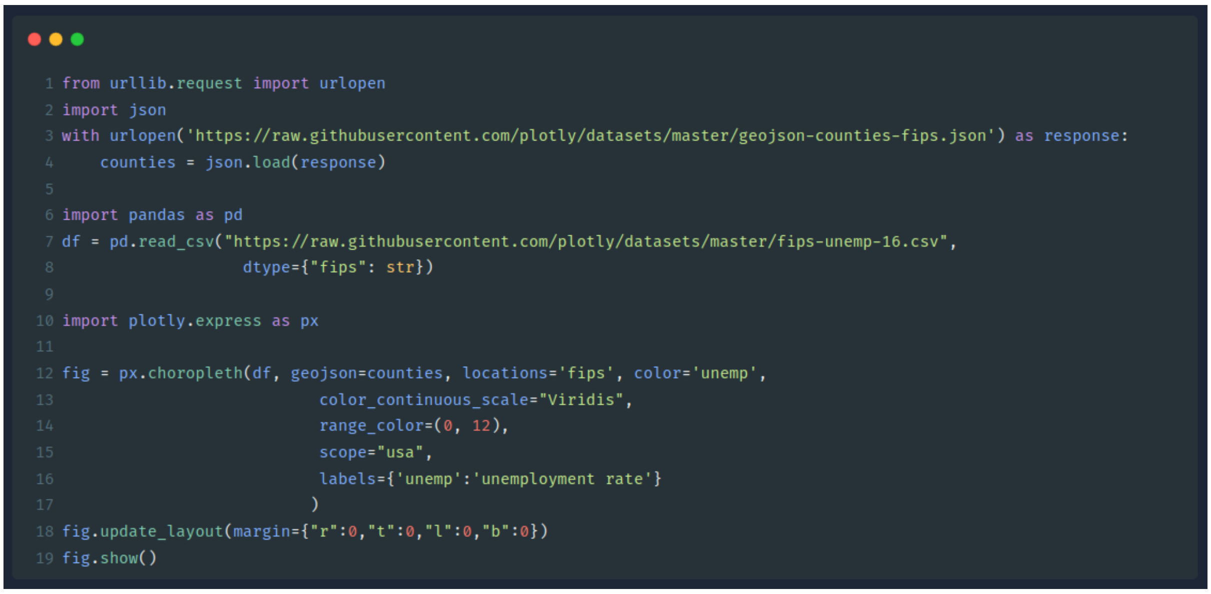
The GitHub repository with codes, research papers and our learnings : Cartograph Classification

Chapter 2

Starting Point

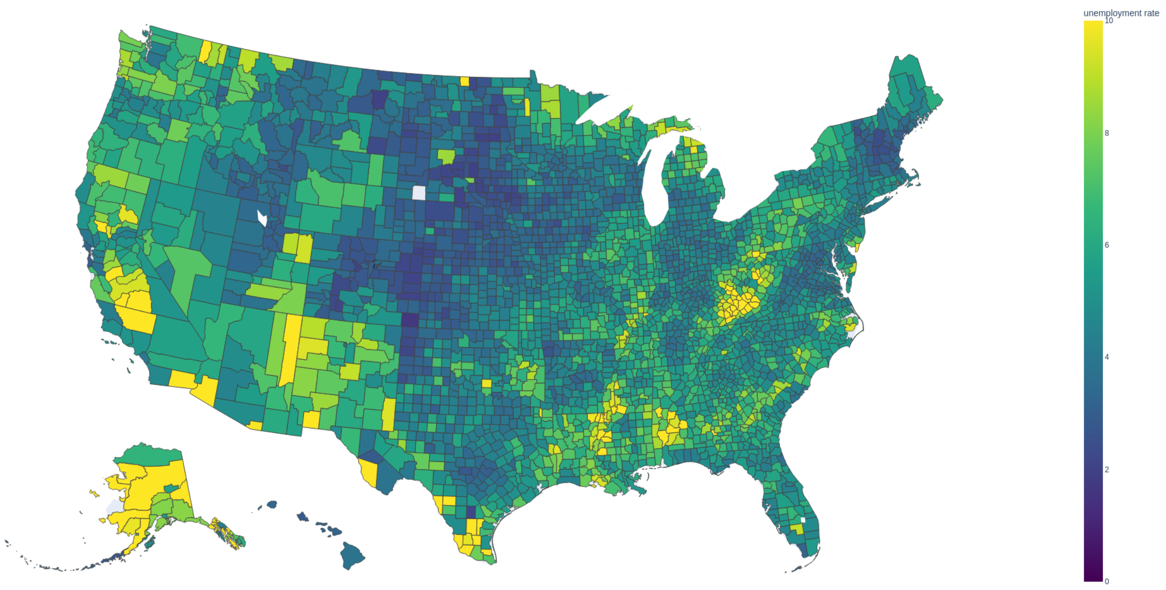
We started with our task by reading the research paper "MapQA: A Dataset for Question Answering on Choropleth Maps(1a)". This research paper explained the usage of choropleth maps of the USA that were created based on survey and questionnares, to answer related questions. It was helpful in understanding the formtion of choropleth map images and how these map images could then be used to answer questions that were related to the data that was used in creating these images.

For the initial step, we understood how choropleth maps can be generated and implemented a Python script that used the library Plotly express and a unemployment rate in the USA dataset to generate choropleth maps. Doing this helped us in understanding what data was present in the choropleth maps and what would need to be extracted from the map images in the MapQA dataset in order to make an analysis.



```python
from urllib.request import urlopen
import json
with urlopen('https://raw.githubusercontent.com/plotly/datasets/master/geojson-counties-fips.json') as response:
    counties = json.load(response)

import pandas as pd
df = pd.read_csv("https://raw.githubusercontent.com/plotly/datasets/master/fips-unemp-16.csv",
                   dtype={"fips": str})

import plotly.express as px

fig = px.choropleth(df, geojson=counties, locations='fips', color='unemp',
                           color_continuous_scale="Viridis",
                           range_color=(0, 12),
                           scope="usa",
                           labels={'unemp':'unemployment rate'}
                          )
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```

This was the choropleth map plot generated upon running the code:



This initial step helped us understand the features we would need to take a deeper look at to make a good analysis on the choropleth map data.

Chapter 3

Annotation of Images

3.1 Objective

We focused on the automated annotation of map images to detect the presence and location of 3 distinct features - Title, Map and Legends. Explored and leveraged the use of Machine Learning models to identify and annotate the required features, thus automating the task.

3.2 Base Dataset

The paper - "MapQA: A Dataset for Question Answering on Choropleth Maps(1a)", used a dataset of roughly 20,000 images for their aim and we decided to explore it and maybe use it to further our task. The map images were of 2 formats :

1. Ones with discrete legends
2. Ones with colour bar

We decided to work only on the map images with the discrete legends. Our classification mechanism is thus useful in helping separate the 2 types of images.

3.3 Observations

The dataset contains a large amount of map images, hence we manually segregated 5000 images into 2 separate directories - one for the images with discrete legends and one for the ones with the colour bar.

For the choropleth map images observed, there were 3 main features:

1. The title of the image
2. The map in the image
3. The data values in the image

Upon observing each image in the folder of images with discrete legends, there were certain observations that were made:

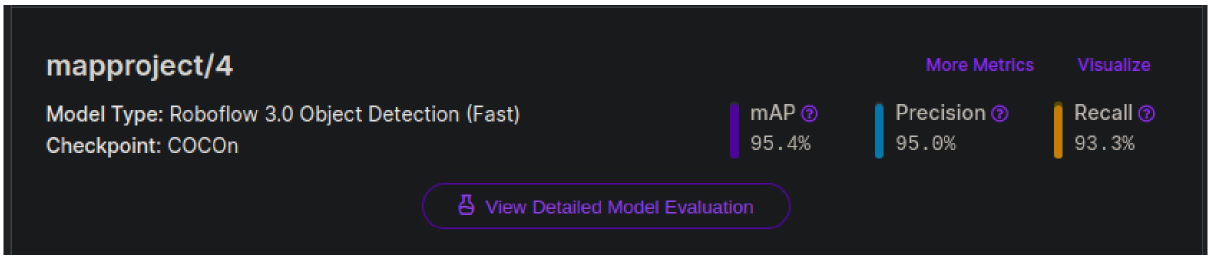
1. The title of the image represented the data that was used in creating the choropleth map.
2. The map of the USA is always found in the center of the image.
3. The discrete legends were either found in the bottom of the image, below the map or to the right hand side, alongside the map.
4. The colour corresponding to the value denoted by the discrete legends is in a box to the left of the numerical value.
5. The title of the image is always found at the top of the image. The map of the USA followed it in the verical direction.

3.4 Dataset Preparation and Roboflow Model

Realising the need to be able to separate and extract these features from the map images, we decide to take help of annotation tools to help annotate the parts of the image that would be useful to our work. We initially used LabelStudio(with the help of DagsHub, where we created a repository MapProject) for our work but then switched to Roboflow and annotated around 351 images with the 3 classes - Title, Map and Legends.

Hence, we curated a dataset consisting of 351 map images with discrete legends. These images were split it into testing, training and validation sets, each image annotated to indicate the presence and location of the 3 classes - Title, Map and Legends.

With the dataset curated, we used Roboflow's own model - Roboflow 3.0 Object Detection, with these images and it provided us with positive results. The following image shows the accuracy metrics obtained on training the model with just 81 images.



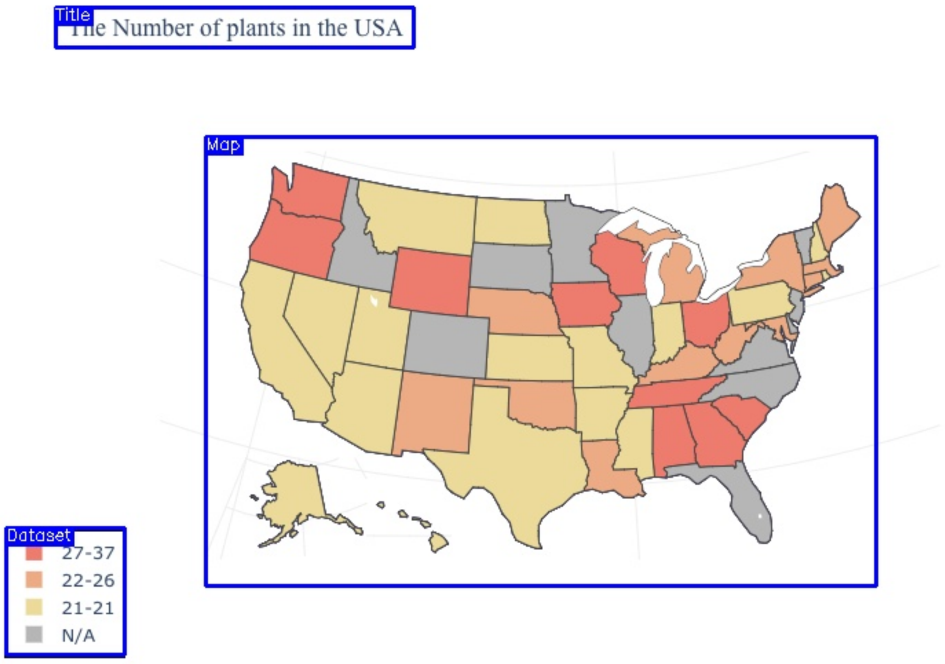
mapproject/4

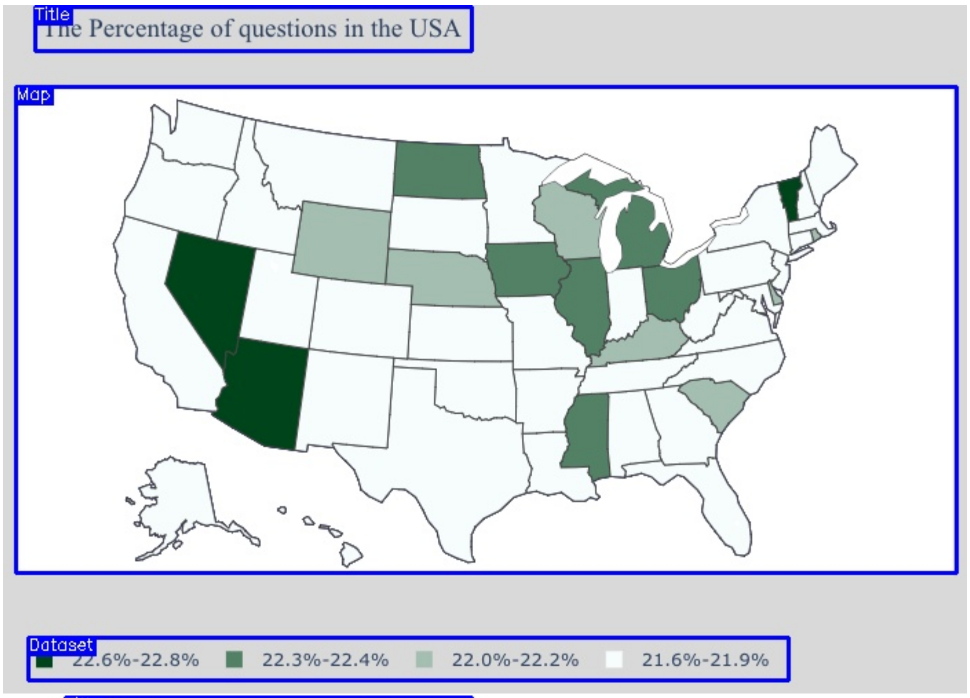
Model Type: Roboflow 3.0 Object Detection (Fast)
Checkpoint: COCOn

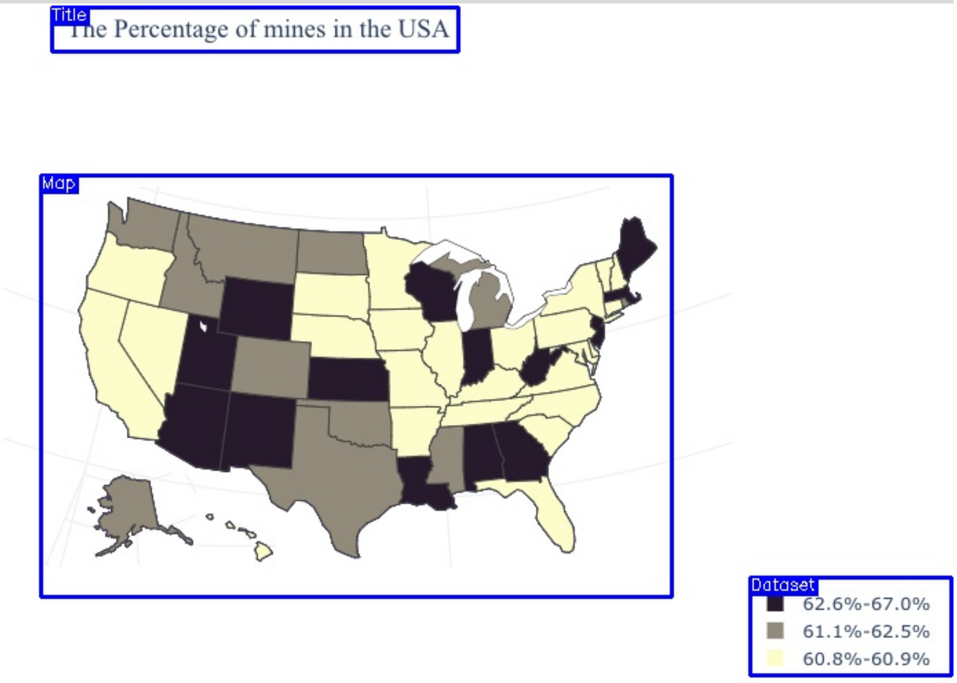
mAP 95.4%
Precision 95.0%
Recall 93.3%

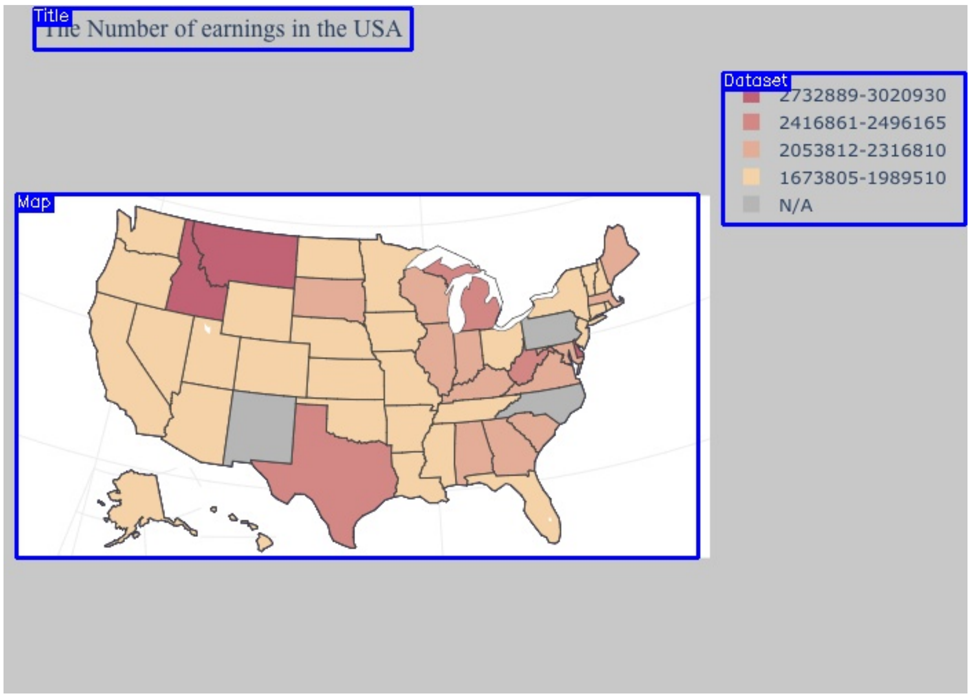
Roboflow also provided the API to use the model used for the annotations and thus, we used the API in a Kaggle notebook along with a larger dataset created with 351 images. Predictions were made on around 90 images with this model to check the accuracy metrics.

A few examples of the model's annotations:




Title
The Percentage of questions in the USA
Map
Dataset
22.6%-22.8%
22.3%-22.4%
22.0%-22.2%
21.6%-21.9%


Title
The Percentage of mines in the USA
Map
Dataset
62.6%-67.0%
61.1%-62.5%
60.8%-60.9%

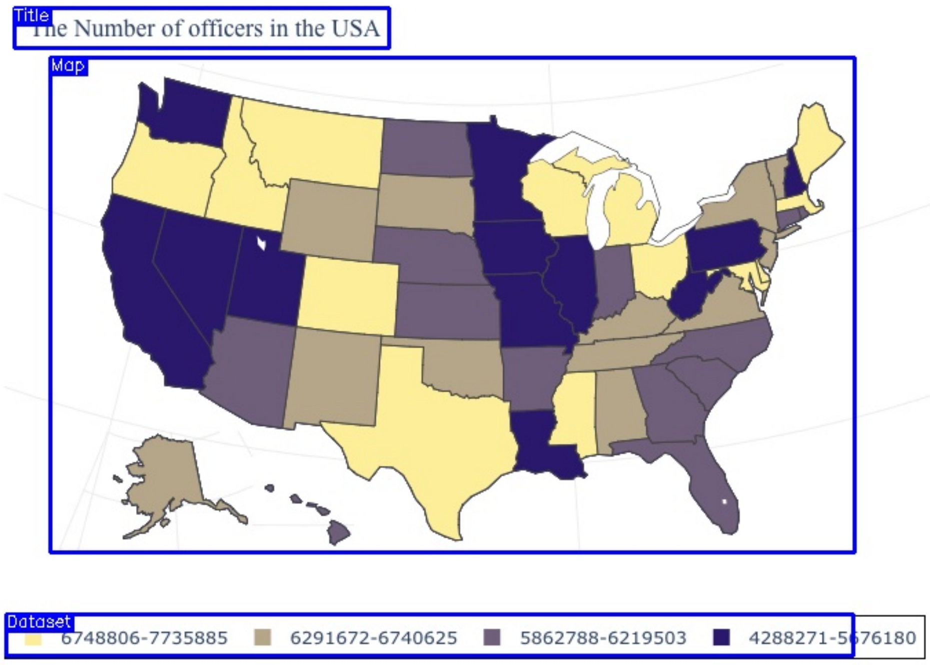


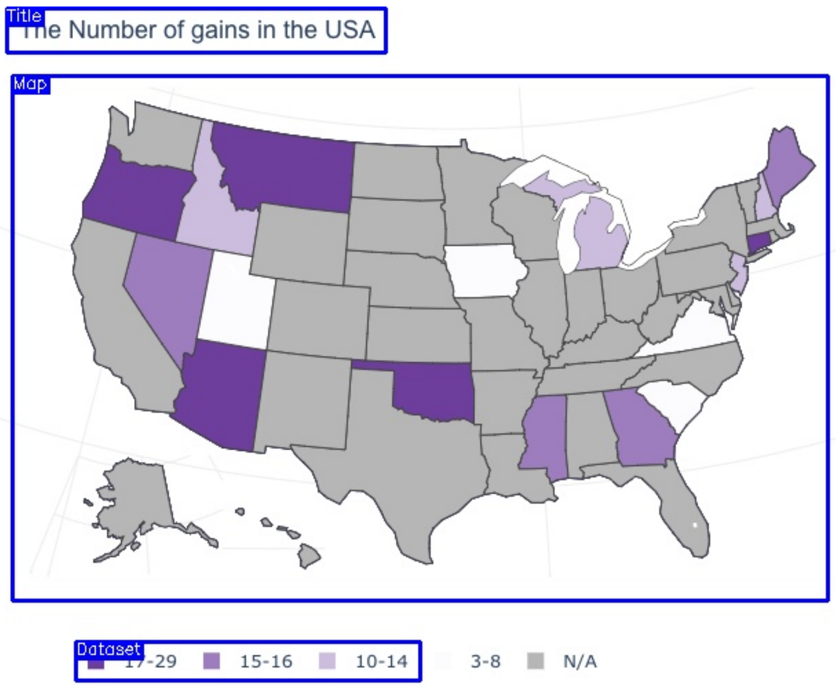
3.5 Inference and Issues:

Observations made on the model's predictions(manual observations):

1. The title was accurately predicted and the bounding box was very similar to the one done manually.
2. The map was also predicted with decently high accuracy.
3. The legends/data values were not always predicted well. There were certain cases as shown in the next few images where the bounding box was incomplete and/or not quite accurate.

The issue with annotation of the discrete legends can be seen in the following images:





In order to work on fixing this issue, on the advice of Prof. Jaya, we decided to use the model's predictions only on the annotations of the Title and the Map. In order to evaluate the legends and their corresponding colour, we decided to use OCR models(in Chapter 4 to gain the numerical value and also obtain the bounding box of the numbers.

Chapter 4

Accuracy Metrics

4.1 Objective

Upon annotating the map images, we focused on obtaining accuracy metrics on the annotations predicted in order to give us an idea of the reliability and validity of the model that we have employed for our use case.

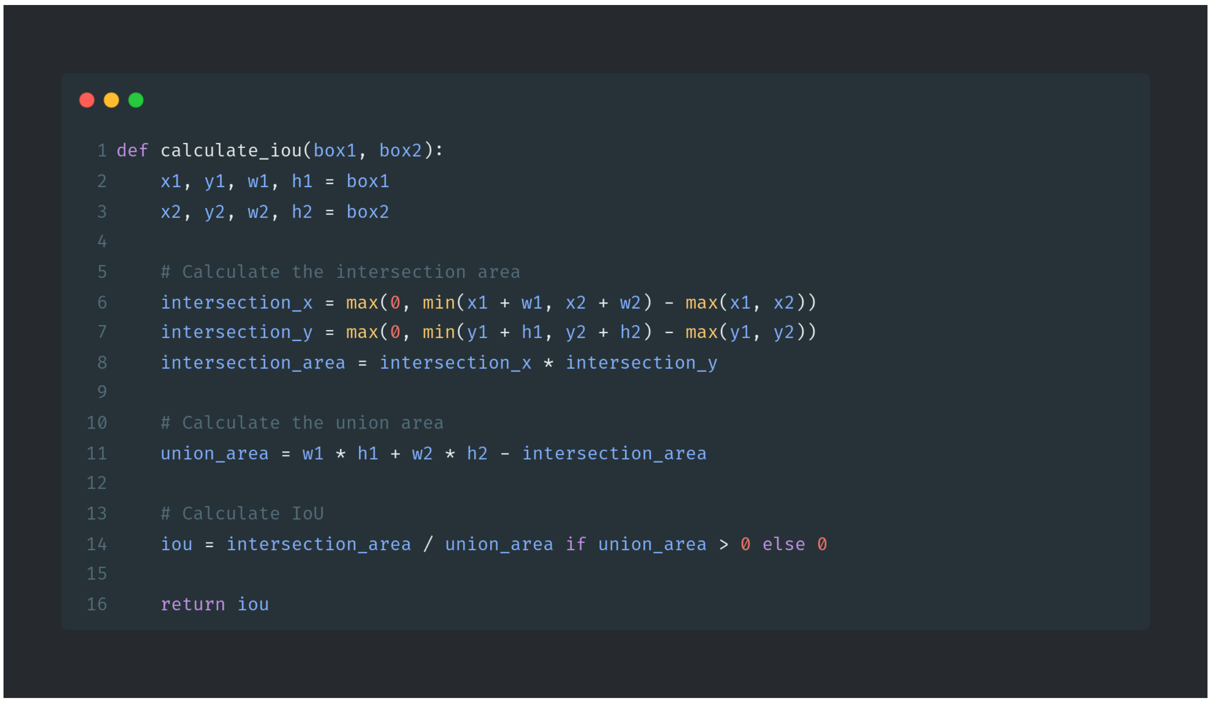
4.2 Intersection Over Union(IOU)

In order to test the accuracy of the model's annotations/predictions on the title and map, we used the Intersection over Union(IOU) technique which compares the bounding boxes of the annotations in the test dataset and those predicted by the model.

Intersection over Union is evaluated using the bounding boxes predicted by the model and that of the annotation made in the ground truth. Consider A to be the predicted bounding box and B as the bounding box in the ground truth. IOU is then calculated as:

$$IOU = \frac{Area\ of\ Overlap}{Area\ of\ Union} = \frac{A \cap B}{A \cup B}$$

We saved predictions made by the model on around 80 images that we had previously annotated manually and used Interesection over Union(IOU) code as follows:



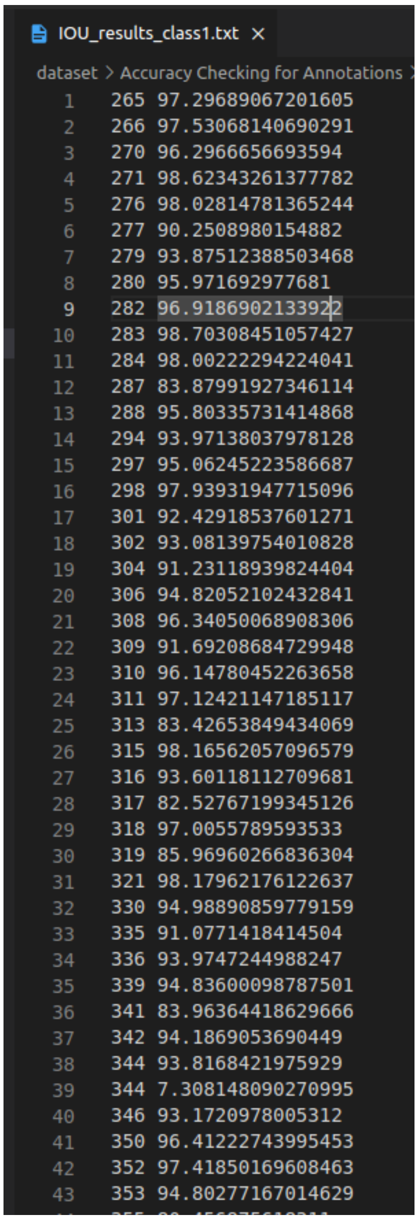
```python
def calculate_iou(box1, box2):
    x1, y1, w1, h1 = box1
    x2, y2, w2, h2 = box2

    # Calculate the intersection area
    intersection_x = max(0, min(x1 + w1, x2 + w2) - max(x1, x2))
    intersection_y = max(0, min(y1 + h1, y2 + h2) - max(y1, y2))
    intersection_area = intersection_x * intersection_y

    # Calculate the union area
    union_area = w1 * h1 + w2 * h2 - intersection_area

    # Calculate IoU
    iou = intersection_area / union_area if union_area > 0 else 0

    return iou
```

4.3 Issue faced in implementation

We could not directly implement IOU on the predicted images since the model's predictions were provided in the format x,y,w and h in the ranges 200-700 but the dataset had these same values in the range 0-1(ranging up to 6 decimal places). The conversion was neither apparent in the dataset nor in the documentation provided by Roboflow.

While preparing the dataset, we were resizing the images to 640x640 pixels but upon trying this method, wherein we multiplied x,y,w and h in the test data values with the scaling factor of 640, we were expecting to get high accuracy values but that turned out to not be the case.

Upon looking deeper into the dataset and roboflow's website, we came across the figure's dimensions in the section where we can manually annotate the image. The dimensions were 500x700 and upon changing the multiplying factor with this, we achieved the accuracy results that we expected as shown in the below image.



IOU_results_class1.txt

dataset > Accuracy Checking for Annotations

```
265 97.29689067201605
266 97.53068140690291
270 96.2966656693594
271 98.62343261377782
276 98.02814781365244
277 90.2508980154882
279 93.87512388503468
280 95.971692977681
282 96.9186902133922
283 98.70308451057427
284 98.00222294224041
287 83.87991927346114
288 95.80335731414868
294 93.97138037978128
297 95.06245223586687
298 97.93931947715096
301 92.42918537601271
302 93.08139754010828
304 91.23118939824404
306 94.82052102432841
308 96.34050068908306
309 91.69208684729948
310 96.14780452263658
311 97.12421147185117
313 83.42653849434069
315 98.16562057096579
316 93.60118112709681
317 82.52767199345126
318 97.0055789593533
319 85.96960266836304
321 98.17962176122637
330 94.98890859779159
335 91.0771418414504
336 93.9747244988247
339 94.83600098787501
341 83.96364418629666
342 94.1869053690449
344 93.8168421975929
344 7.308148090270995
346 93.1720978005312
350 96.41222743995453
352 97.41850169608463
353 94.80277167014629
```

Chapter 5

Attempt at OCR

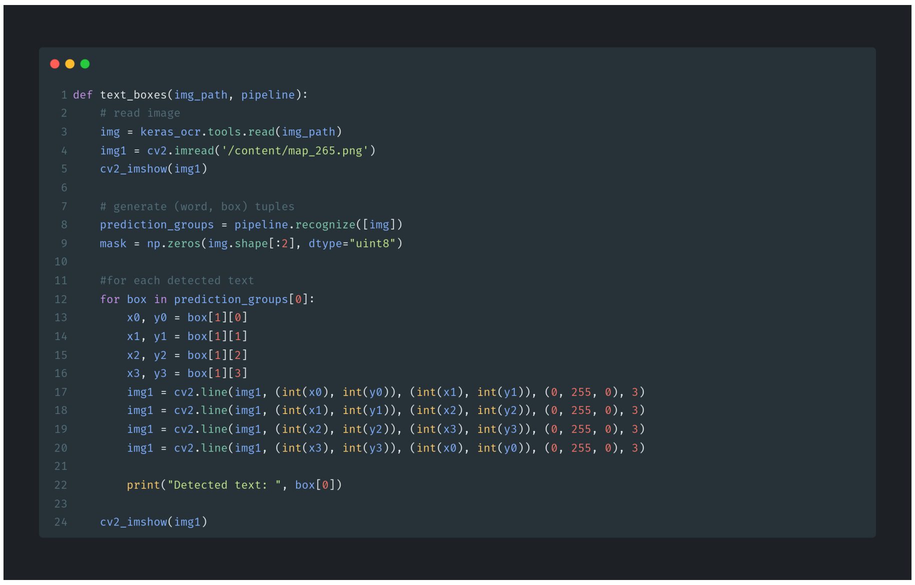
5.1 Objective

Due to the issue faced in predicting the annotations for the discrete legends, we focussed on using OCR technologies and models to help extract the discrete legends and the data values.

5.2 KerasOCR

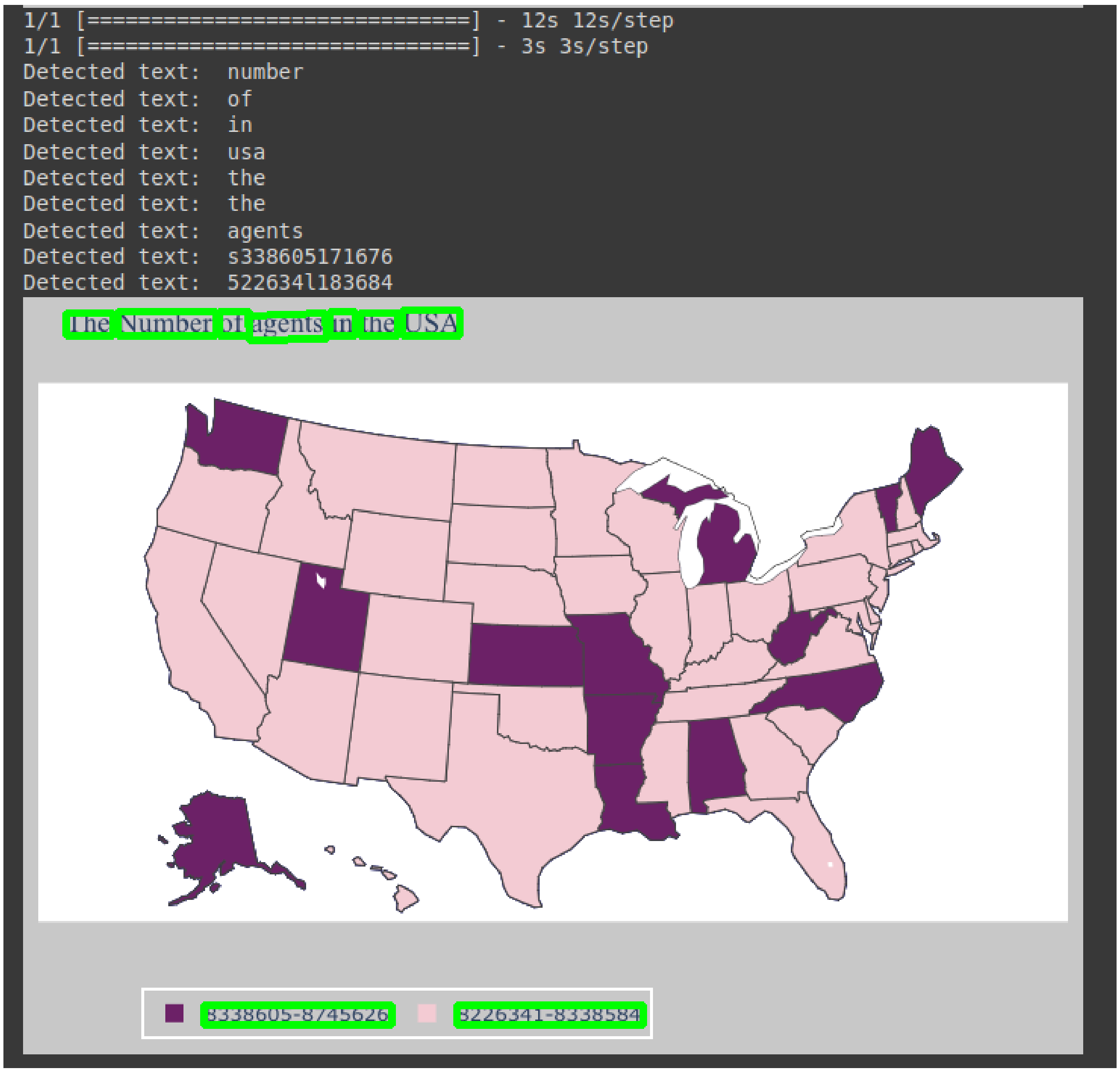
The initial attempt at obtaining the numerical data values from the map was by using the KerasOCR model. We coded a script to recognise text in the map image and form bounding boxes around them. This script can be found here : KerasOCR

The function we used to creating bounding boxes around the recognised text data is as follows:



```python
def text_boxes(img_path, pipeline):
    # read image
    img = keras_ocr.tools.read(img_path)
    img1 = cv2.imread('/content/map_265.png')
    cv2_imshow(img1)

    # generate (word, box) tuples
    prediction_groups = pipeline.recognize([img])
    mask = np.zeros(img.shape[:2], dtype="uint8")

    #for each detected text
    for box in prediction_groups[0]:
        x0, y0 = box[1][0]
        x1, y1 = box[1][1]
        x2, y2 = box[1][2]
        x3, y3 = box[1][3]
        img1 = cv2.line(img1, (int(x0), int(y0)), (int(x1), int(y1)), (0, 255, 0), 3)
        img1 = cv2.line(img1, (int(x1), int(y1)), (int(x2), int(y2)), (0, 255, 0), 3)
        img1 = cv2.line(img1, (int(x2), int(y2)), (int(x3), int(y3)), (0, 255, 0), 3)
        img1 = cv2.line(img1, (int(x3), int(y3)), (int(x0), int(y0)), (0, 255, 0), 3)

        print("Detected text: ", box[0])

    cv2_imshow(img1)
```

The results we obtained upon running this script on an image($map - 265$) was as follows:



```
1/1 [==============================] - 12s 12s/step
1/1 [==============================] - 3s 3s/step
Detected text:  number
Detected text:  of
Detected text:  in
Detected text:  usa
Detected text:  the
Detected text:  the
Detected text:  agents
Detected text:  s338605171676
Detected text:  5226341183684
```

5.3 Issue faced

As can be seen in the result, the text detection in the Title is perfect but the numerical data detected in the Discrete Legends is not accurate. It has detected the first value 8338605−8745626 as s338605171676 and similarly inaccurately for the second value.

We also tried to use the image after remove the Title by masking it's bounding box, obtained from the annotation model, but were not able to achieve it correctly.

5.4 Future plan

We plan to try out another already present model for the numerical data extraction, starting with another well-known OCR model "Tesseract OCR". We also have to obtain the distance in number of pixels that we need to move to the left hand side in order to gain information on the colour corresponding to that numerical text.

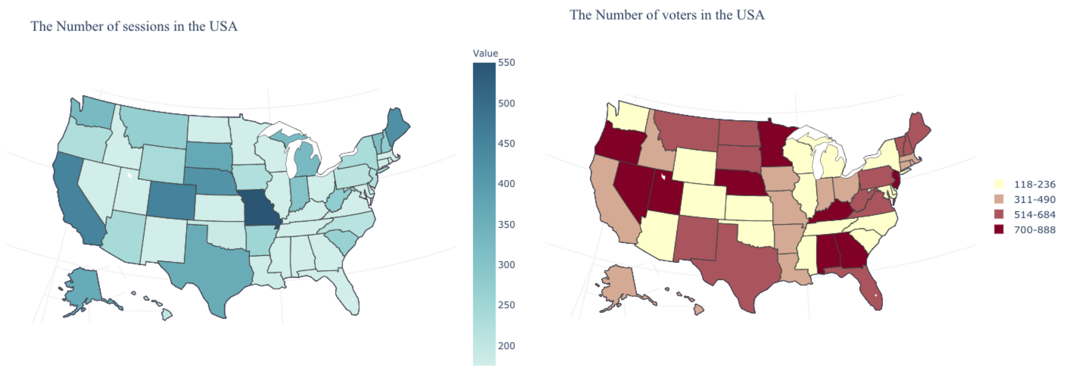
In order to effectively use OCR methodologies for our task of obtaining the numerical text, masking the Title in these map images can prove to be quite effective, so working on that is of utmost importance. We plan to mask the image before using any OCR methodologies on it and unmask the image after we extract the required data from the Discrete Legends.

Chapter 6

Map Classification

6.1 Objective

We focused on the automated classification of maps into two distinct categories: those with discrete legends and those displaying a colorbar. Utilising Roboflow's Object Detection Model, we employed a robust methodology to identify and classify these maps based on the presence and location of either discrete legends or colorbars.

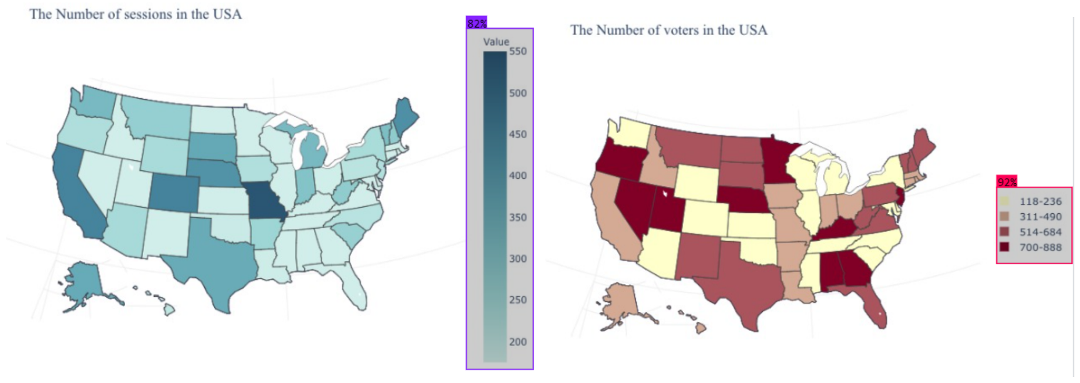

The Number of sessions in the USA
Value
550
500
450
400
350
300
250
200
The Number of voters in the USA
118-236
311-490
514-684
700-888

6.2 Data Preparation

We curated a dataset comprising of 333 map images with both discrete legends and colorbars. We split it into testing, training and validation set. Each image was annotated to indicate the presence and location of the legend or colorbar.

6.3 Roboflow's Model

Roboflow's object detection model was employed to recognize and classify the maps. The model was trained to distinguish between class 1 (discrete legend) and class 0 (colorbar) based on the identified regions in the images.


The Number of sessions in the USA
82%
Value
550
500
450
400
350
300
250
200
The Number of voters in the USA
92%
118-236
311-490
514-684
700-888

Above figure shows the classification of the two maps. Our model correctly classifies colourbar map with a confidence of 82% and discrete map with a confidence of 92%. Now in order to get an accuracy value over the dataset, we do the following post-processing.

6.4 Post - Processing

We get labels from annotated images of our dataset. The labels include class-id followed by the dimensions of the (actual) annotated boxes. Since we are just concerned with the class-id we drop the annotation dimensions. Similarly on the predicted values, we drop all the predicted dimensions excluding the class-id.

Below is how the predictions from the model is represented in JSON format.

```
{
  "predictions": [
    {
      "x": 593,
      "y": 248.5,
      "width": 90,
      "height": 91,
      "confidence": 0.924,
      "class": "discrete",
      "class_id": 1
    }
  ]
}
```

6.5 Accuracy

After we have stored the predicted and actual class-id of all the maps, we find the accuracy by simply checking the number of correct predictions over all the test dataset.
accuracy = correct predictions/total predictions * 100%

Over the 333 split data we used, we got a perfect 100% accuracy. This is a reliable accuracy because the dataset we used of the two map classes are very distinct in features (colourbar and discrete).

Chapter 7

Conclusion and Future plans

By reducing the reliance on manual classification and annotation, the project overcomes the limitations associated with human subjectivity and the inherent challenges of handling extensive map datasets. The robustness of the implemented model, demonstrated through its consistent and reliable performance, marks a significant advancement in the automation of geospatial workflows.

The implications of this work extend across diverse domains, offering a streamlined solution for professionals in cartography, remote sensing, and data visualization. The speed and precision achieved in map classification not only enhance decision-making processes but also contribute to a more objective and standardized interpretation of spatial data.

Further progress can be made in improving the automated classification and annotation by extracting the values and corresponding colour of the discrete legends from the image. Also, as mentioned in 5, there is vast progress that can be made in extracting the numerical text data from the image better and more precisely. Analysis can also be made on the data that is extracted in order to pertain to questions that can be raised on the data.

Chapter 8

References and Links

1. Research Papers and Datasets:
   (a) MapQA: A Dataset for Question Answering on Choropleth Maps
   (b) Automatic Image Annotation using Deep Learning Representations
   (c) A review on automatic image annotation techniques
   (d) MapQA Dataset
   (e) GitHub link of MapQA Dataset
   (f) Custom Dataset for Annotation Task
2. Our work:
   (a) Image Annotation and Roboflow Model
   (b) Annotation Task notebook
   (c) Annotation Model
   (d) Classification Task Notebook
   (e) Classification Model
   (f) KerasOCR documentation
   (g) KerasOCR Notebook
   (h) Our code