CMA: An End-to-End System for Reverse Engineering Choropleth Map Images

Prince Nileshbhai Butani, Jaya Sreevalsan-Nair *Senior Member, IEEE*, and Nilay Kamat

***Abstract*—**
Choropleth maps are widely used geovisualizations owing to their simplicity, especially for applications involving political, climate, and other geospatial data for contiguous regions. There is a need for automated data extraction from such maps to aid the human-in-the-loop in handling cognitive overload from large-scale visualization generation and visual impairments. There are gaps in generalizing such a system for choropleth maps with different types of color legends. We propose the Choropleth Map Analytics (CMA) system to address these gaps using a six-step workflow involving deep learning architectures and tools. We propose a novel method for color-to-data mapping for different color legend types. We finally demonstrate the usability of CMA for a set of choropleth images in climate research for a text summarization application. Our work is a step towards reverse engineering choropleth visualizations. Our code and curated datasets are at https://github.com/GVCL/Choropleth-CMA

***Index Terms*—Choropleth maps, Geovisualization, Classification, Segmentation, Deep learning, Text extraction, Inverse color mapping, Text summarization**

I. Introduction

CHART images are an output of visualization processes, which involve a human-in-the-loop. However, given the advancements in computing technology, there is an explosion of visualizations available for human consumption, leading to cognitive overload. In parallel, visualizations are not accessible to the visually impaired [1]. These reasons necessitate automated data extraction from chart images or *reverse engineering* visualizations [2]. Choropleths are widely used geovisualization techniques, owing to their simplicity. Choropleths use color mapping of geographical (contiguous) regions, including political units like states/countries. Given the significant variation in choropleth map generation, deep learning (DL) techniques are required for the task of data extraction from choropleths [3]. There is limited work in the problem statement concerning choropleths, though the same is well-studied for other simpler and ubiquitous visualizations, such as, bar charts, scatter plots, etc. [1].

There has been recent work on a question-answering system for choropleth images [3], recoloring of isarithmic maps [4], and extraction of visual encodings from color bars of isarithmic maps [5]. However, there still exists a gap in a generic implementation of data extraction from choropleths. There is also minimal work in addressing data extraction of map images without coordinates, which is how most maps are rendered [5].

To address these gaps, we propose the Choropleth Map Analytics (CMA) system, which is implemented using an end-to-end six-step workflow. CMA workflow includes the use of state-of-the-art DL architectures for segmentation and text extraction. We observe that there are different types of color legends used in choropleths, namely, the continuous color bar and quantized color boxes. Thus, CMA is designed to handle different map classes. We finally showcase the use of our CMA system for a set of related images to give a text summarization, thus providing a natural language interface for human consumption of a joint inference of multiple choropleth images. Our contributions are:

- Design and implementation of CMA for data extraction from choropleth map/chart images,
- A novel method for color-to-data mapping based on the type of color legend, as the continuous color bar and quantized color boxes in a separated layout, and
- Design of evaluation of CMA using multiple images associated with a joint text summarization task.

II. The Workflow of CMA

Our proposed end-to-end system, Choropleth Map Analytics (CMA), integrates optimal deep learning models for data extraction from choropleth maps using a six-step workflow (Fig. 1). Data extraction is needed for reverse engineering charts to improve accessibility [2].

S_1: Color Legend-based Chart Classification: We consider two classes of the choropleth images based on the color legend, namely, the *quantized color legend in separated layout* ($\mathbf{C_{sep}}$, Example-1 in Fig. 1) and *continuous color bar* ($\mathbf{C_{con}}$, Example-2 in Fig. 1). The color bar can also be quantized in color bar rendering style or joint layout [5], thus forming a third class. But, we have considered only two classes here due to data availability. Since these classes are visually discernible, we use one of the efficient state-of-the-art deep learning (DL) architectures. Here, we have experimented with VGG16 [6] and ResNet-50 [7]. VGG16 has a fixed kernel size and uses fewer parameters and lower training time than its predecessors. With 16 convolutional layers and 3×3 kernels, VGG16 gives a favorable tradeoff between the number of parameters and accuracy, amongst its contemporary DL architectures, *e.g.,* Alexnet, ResNet, etc. On the other hand, ResNet-50 can be

P. Nileshbhai Butani and J. Sreevalsan-Nair are with the Graphics-Visualization-Computing Lab (GVCL), International Institute of Information Technology, Bangalore (IIIT-B), Karnataka 560100, India. Lab website: https://www.iiitb.ac.in/gvcl/ | email: jnair@iiitb.ac.in

N. Kamat is with IIIT-B.

Manuscript received on May 24, 2024.

0000–0000/00$00.00 © 2024 IEEE

This article has been accepted for publication in IEEE Geoscience and Remote Sensing Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LGRS.2024.3444600

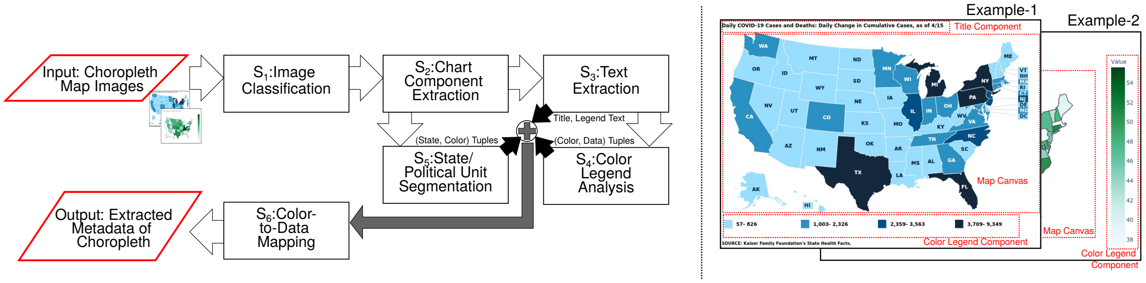


Fig. 1. (Left) Six-step workflow of Choropleth Map Analytics (CMA) end-to-end system. (Right) Components of a choropleth map image that are extracted in $\mathbf{S_2}$. Example-1 and 2 show quantized color legend with separated layout ($\mathbf{C_{sep}}$) and continuous color bar ($\mathbf{C_{con}}$), respectively, in the map of mainland USA.

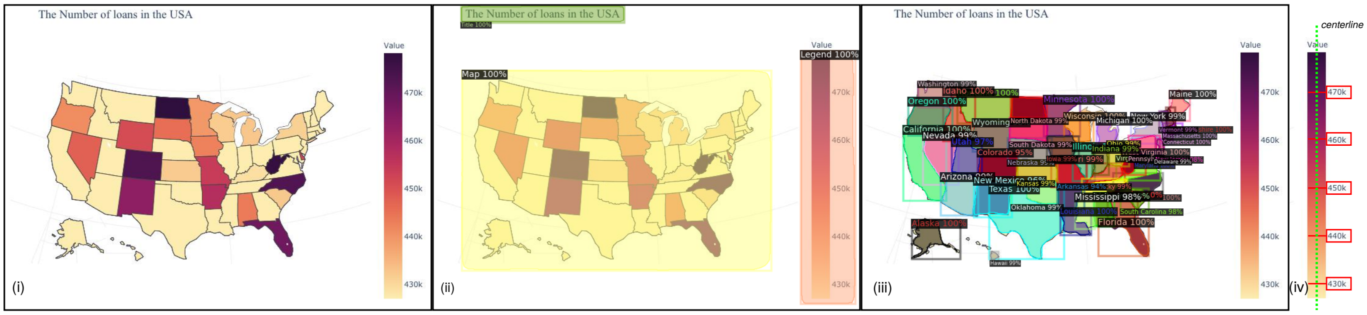


Fig. 2. The outputs of the Mask R-CNN of Detectron2 segmentation for a (source) choropleth image of mainland USA (shown in (i)) for: (ii) component extraction ($\mathbf{S_2}$), and (iii) state segmentation ($\mathbf{S_5}$). The color at tick marks in the continuous color bar in $\mathbf{S_4}$ is determined using the text boxes from $\mathbf{S_3}$ and drawing a (red) line leftwards till the color bar boundary, as shown in (iv). The colors are selected from the (green) centerline.

trained more efficiently with much fewer parameters but more layers, *i.e.,* 50 layers, by using shortcut (residual) connections. In our implementation, the outputs are written as a CSV file, with columns as filename and its corresponding class, namely, *Discrete or Continuous*.

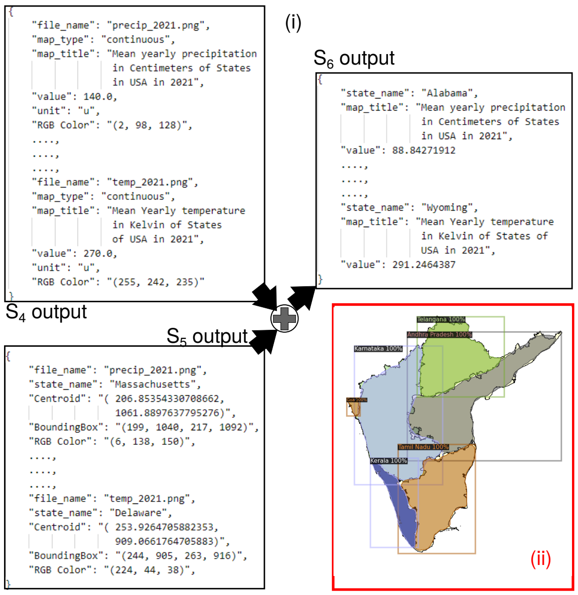
$\mathbf{S_2}$: Chart Component Extraction: The choropleth image has three salient regions, namely, the title, map canvas, and legend regions (Fig. 1). We generate the ground truth for the training set and use Detectron2 [8] to train the model. Detectron2 supports state-of-the-art object detection and segmentation algorithms and often uses DL architectures, *e.g.,* Mask R-CNN (Region-based Convolutional Neural Network). Mask R-CNN extends Faster R-CNN by adding a branch for pixel-level mask outputs, thus providing pixel-level segmentation. Thus, Mask R-CNN gives segments with fine-grained boundaries, which we coarsen using rectangular bounding boxes (Fig. 2). In our implementation, the output is stored in the form of a data frame with four column names as file names, and bounding boxes for each component.

$\mathbf{S_3}$: Text Extraction: Text is equally important in making sense of a visualization as its visual encoding, namely, marks and channels. Text extraction involves detection and recognition, which are done using Optical Character Recognition (OCR). The state-of-the-art in OCR uses DL models. We choose a state-of-the-art OCR model, namely, PaddleOCR [9]. It uses a PaddlePaddle framework which incorporates attention mechanisms in CNNs. PaddleOCR is robust at handling different map layouts, fonts, and graphical elements. PaddleOCR is effective in handling a variety of choropleth charts, including ones where the legend is rendered within the map canvas in a hierarchical format. The chart components of $\mathbf{S_2}$ are fed as input to PaddleOCR, where the images are cropped by their bounding boxes. Thus, the title and legend sub-images are separately processed. The title is extracted from its bounding box using PaddleOCR in $\mathbf{S_3}$, and stored as the chart image title. The color legend is processed further in $\mathbf{S_4}$.

$\mathbf{S_4}$: Color Legend Analysis: This step involves preliminary processing of the (cropped) sub-image of the color legend to map the data values available as text, to their corresponding colors, after extracting the data values in $\mathbf{S_3}$. It is implemented differently for maps with legend as quantized color boxes in separated layouts ($\mathbf{C_{sep}}$) and as the continuous color bar ($\mathbf{C_{con}}$).

For $\mathbf{C_{sep}}$, the data values corresponding to the discrete colors represent a data histogram. Thus, in the legend, its text corresponds to a color box and usually contains a hyphen. Focusing on numeric value extraction, we discard the non-numeric characters, *e.g.,* the hyphen. Then, we determine one or two float values, against each color box. In our implementation, the output is a data frame with file name, choropleth class, and title, along with tuples of RGB values from color boxes with their corresponding numeric value, *i.e.,* average of the interval in the data histogram. Hence, there is information loss in data extraction for such maps owing to the quantization during the choropleth generation.

For $\mathbf{C_{con}}$, the bounding box of the text from OCR is used to localize the data value of a tick mark or *color interval* boundary. We assume that the orientation of the color bar is *vertical*, which is predominantly seen in practice. The centerline along the height of the text box is extended leftwards until the boundary of the color bar in the sub-image (Fig. 2). Identifying colors corresponding to the text provides intervals



```
{
    "file_name": "precip_2021.png",
    "map_type": "continuous",
    "map_title": "Mean yearly precipitation
                  in Centimeters of States
                  in USA in 2021",
    "value": 140.0,
    "unit": "u",
    "RGB Color": "(2, 98, 128)",
    ....,
    ....,
    ....,
    "file_name": "temp_2021.png",
    "map_type": "continuous",
    "map_title": "Mean Yearly temperature
                  in Kelvin of States
                  of USA in 2021",
    "value": 270.0,
    "unit": "u",
    "RGB Color": "(255, 242, 235)"
}
```

```
{
    "file_name": "precip_2021.png",
    "state_name": "Massachusetts",
    "Centroid": "( 206.85354330708662,
                  1061.8897637795276)",
    "BoundingBox": "(199, 1040, 217, 1092)",
    "RGB Color": "(6, 138, 150)",
    ....,
    ....,
    ....,
    "file_name": "temp_2021.png",
    "state_name": "Delaware",
    "Centroid": "( 253.9264705882353,
                  909.0661764705883)",
    "BoundingBox": "(244, 905, 263, 916)",
    "RGB Color": "(224, 44, 38)",
}
```

```
{
    "state_name": "Alabama",
    "map_title": "Mean yearly precipitation
                  in Centimeters of States
                  in USA in 2021",
    "value": 88.84271912
    ....,
    ....,
    ....,
    "state_name": "Wyoming",
    "map_title": "Mean Yearly temperature
                  in Kelvin of States of
                  USA in 2021",
    "value": 291.2464387
}
```

Fig. 3. (i) Metadata format for storing outputs from $\mathbf{S_4}$, $\mathbf{S_5}$, and $\mathbf{S_6}$. (ii) Results of $\mathbf{S_5}$ on southern region of India segmenting six states.

of colors represented using *piecewise linear interpolation*. The inverse interpolant is subsequently used in $\mathbf{S_6}$ to compute the data value corresponding to a query color from the map canvas. The color against the text box is determined from the centerline of the color bar region (Fig. 2). The text data accompanying the numeric values usually include physical units or suffixes, such as 'K' for "kilo." Hence, we extract the numeric values alone for mapping color to data, and add all other text data in the output. The output is the same as that of $\mathbf{C_{sep}}$, except without the information loss from quantization, and is stored in a specific metadata format (Fig. 3).

$\mathbf{S_5}$: State/Political Unit Segmentation: Data extraction requires image segmentation to extract states or political units that define the choropleth. Not all maps have segment labels. Our data also does not have the latitude/longitude information, unlike those in prior work [5]. To address the challenge of limited spatial reference information, we use advanced object detection algorithms to segment states/political units. We once again use Mask R-CNN in Detectron2 for pixel-based segmentation where political unit names are used as pixel-level labels. Detectron2 gives accurate results, especially in the case of regions with small areas. Similar to $\mathbf{S_2}$, once the fine-grained boundary is extracted, a coarser rectangular bounding box is rendered, and the segments are identified using bounding box centroids. We also use the color of the centroid to find the RGB color corresponding to the political unit. However, the model has to be retrained specifically for a specific region using the corresponding ground truth (Fig. 2 and 3(ii)). In our implementation, the output is stored as a data frame with the file name, and for each segment, its name label, bounding box coordinates, box centroid, and RGB color. A specific metadata format is used for storing the output (Fig. 3).

$\mathbf{S_6}$: Color-to-Data Mapping: Similar to $\mathbf{S_4}$, this step is implemented differently for $\mathbf{C_{sep}}$ and $\mathbf{C_{con}}$. For the colors at the centroid of the bounding box of state/political unit, obtained from $\mathbf{S_5}$, we determine the data value corresponding to the colors, *i.e., inverse color mapping*. The output of this step is a data frame of state/political unit names and their corresponding data values, which is stored in the metadata format (Fig. 3).

In $\mathbf{C_{sep}}$, the query color from the map canvas, *i.e.*, the RGB color of any segment from $\mathbf{S_5}$, is compared with all the colors in the legend using Euclidean distance. The closest legend color value is matched with the segment color, and the tuple of (state/political unit name, data value) is stored as output.

In $\mathbf{C_{con}}$, an interval of colors is defined by consecutive lines along the width of the color bar (Fig. 2). We assume that the bottom and top horizontal lines in the color bar represent the min-max data values of the choropleth. For every interval, two consecutive lines mark the min-max of the interval itself. Let C_{min} and C_{max} be the lower and upper bounds of an interval, then we find the parametrized representation α of the query color C_q. We deconstruct the linear interpolant equation to compute α using each color channel in the RGB color model.

$$\frac{C_q^R - C_{min}^R}{C_{max}^R - C_{min}^R} = \frac{C_q^G - C_{min}^G}{C_{max}^G - C_{min}^G} = \frac{C_q^B - C_{min}^B}{C_{max}^B - C_{min}^B} = \alpha.$$

If α satisfies the condition $(0 \leq \alpha \leq 1)$ in a color interval, starting from the bottom of the color bar, we compute the data value v_q of C_q using the linear interpolant of values in the interval bounds, namely, v_{min} and v_{max},

$$v_q = (1 - \alpha) * v_{min} + \alpha * v_{max}.$$

We observed that in a few cases, the query colors returned no valid α values in all color intervals. This was because the query color matched with values either below the lowest or above the highest tick marks. This happens in cases where the tick marks do not represent the min-max values of the colorbar. In such cases, we use a rule and default to clamping the query color to the appropriate bound of the color bar.

III. Implementation Details

We have used the MapQA dataset [3] for implementing CMA. Since the dataset features the choropleth maps of mainland USA, *i.e.,* excluding states of Alaska and Hawaii, we have designed our case studies based on the same.

Dataset: We have used the choropleth map images generated from synthetic datasets, as published in MapQA (MapQA-S) [3] for our work owing to their high-resolution and homogeneity in designs. To ensure class balance, we chose 50 images each of $\mathbf{C_{sep}}$ and $\mathbf{C_{con}}$ choropleth maps. The data is split into 80:10:10 ratio for training, validation, and testing.

Case Studies: Here, we design case studies to demonstrate the working of CMA where multiple choropleths are used to extract data for a joint analysis. We regenerated maps from climate data of the United States Geological Survey (USGS) and Climate Research Unit (CRU), UK, for two different case studies. Our regenerated maps are of high resolution, unlike the ones scrapped from the Internet. Using existing real-world maps from documents or the Internet for case studies will require data curation, which is in the future scope of this work.

The USGS and CRU data give a multivariate dataset of state-wise climate variables of mainland USA, namely, pre-

This article has been accepted for publication in IEEE Geoscience and Remote Sensing Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LGRS.2024.3444600

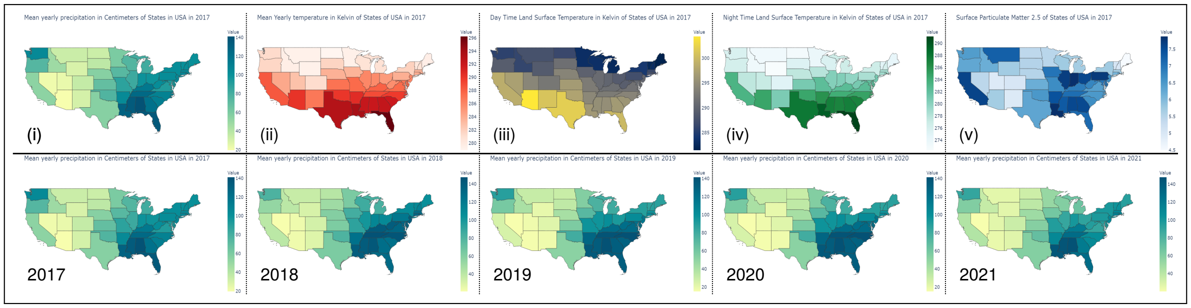

Mean yearly precipitation in Centimeters of States in USA in 2017
Mean Yearly temperature in Kelvin of States of USA in 2017
Day Time Land Surface Temperature in Kelvin of States of USA in 2017
Night Time Land Surface Temperature in Kelvin of States of USA in 2017
Surface Particulate Matter 2.5 of States of USA in 2017
Value
(i)
(ii)
(iii)
(iv)
(v)
Mean yearly precipitation in Centimeters of States in USA in 2017
Mean yearly precipitation in Centimeters of States in USA in 2018
Mean yearly precipitation in Centimeters of States in USA in 2019
Mean yearly precipitation in Centimeters of States in USA in 2020
Mean yearly precipitation in Centimeters of States in USA in 2021
2017
2018
2019
2020
2021

Fig. 4. The $\mathbf{C_{con}}$ choropleth maps used for our case studies. (Top) Multivariate choropleths of climate variables of mainland USA in 2017, with mean values of (i) precipitation, (ii) temperature, (iii) daytime temperature, (iv) nighttime temperature, and (v) PM 2.5 concentration. (Bottom) Choropleth maps of time series of mean precipitation values of USA during 2017-2021.

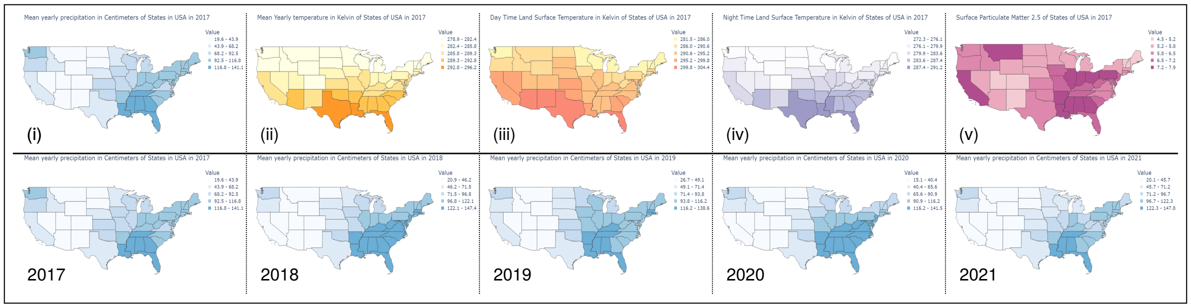

Mean yearly precipitation in Centimeters of States in USA in 2017
Mean Yearly temperature in Kelvin of States of USA in 2017
Day Time Land Surface Temperature in Kelvin of States of USA in 2017
Night Time Land Surface Temperature in Kelvin of States of USA in 2017
Surface Particulate Matter 2.5 of States of USA in 2017
Value
(i)
(ii)
(iii)
(iv)
(v)
Mean yearly precipitation in Centimeters of States in USA in 2017
Mean yearly precipitation in Centimeters of States in USA in 2018
Mean yearly precipitation in Centimeters of States in USA in 2019
Mean yearly precipitation in Centimeters of States in USA in 2020
Mean yearly precipitation in Centimeters of States in USA in 2021
2017
2018
2019
2020
2021

Fig. 5. The $\mathbf{C_{sep}}$ choropleth maps used for our case studies, corresponding to $\mathbf{C_{con}}$ ones in Fig. 4.

cipitation levels, air temperature, PM 2.5 concentrations, and daytime and nighttime land surface temperatures. Our first case study is on text summarization of choropleth maps of different variables for the same year, *i.e.,* 2017. Our second case study is on the time series analysis of annually averaged precipitation values from 2017 to 2021. We regenerated maps for our case studies in both $\mathbf{C_{con}}$ and $\mathbf{C_{sep}}$ classes, as shown in Fig. 4 and Fig. 5, respectively. For each case study, we merged the data tables extracted from each map to generate a single data table.

We make a hypothesis h_{class} stating that the $\mathbf{C_{con}}$ choropleths would give more accurate results than the $\mathbf{C_{sep}}$ ones, owing to the quantization involved in color mapping in $\mathbf{C_{sep}}$.

System Evaluation: For step-wise evaluation of the end-to-end workflow, we use appropriate metrics, as follows:

- For $\mathbf{S_1}$, we compute the classification accuracy using the 2-class confusion matrix.
- To evaluate the segmentation by Detectron2 used in $\mathbf{S_2}$ and $\mathbf{S_5}$, we compute the average precision metrics AP, and AP75 scores. AP score gives the average of precision values, *i.e.,* ratio of true positives to the sum of true and false positives. AP75 scores are the AP values achieved with a minimum threshold of Intersection over Union (IoU) of 0.75.
- For $\mathbf{S_3}$, we evaluate the accuracy of PaddleOCR by comparing the original text within our dataset against the OCR-extracted text.
- We evaluate the final output, *i.e.,* the extracted data, which in turn also provides the accuracy of $\mathbf{S_4}$ and $\mathbf{S_6}$. Using our case studies, we compare the extracted/predicted data ($\hat{y}$) with the ground truth (GT) (y), *i.e.,* the data used to regenerate choropleths for our case studies. We use nMAE, nRMSE, MAPE, and sMAPE metrics as estimators, which are normalized values used widely for comparative analysis. For n observations, we get:

Normalized Mean Absolute Error:

$$\text{nMAE} = \frac{\frac{1}{n}\sum_i |(y_i - \hat{y_i})|}{\frac{1}{n} y_i}.$$

Normalized Root Mean Square Error:

$$\text{nRMSE} = \frac{\text{RMSE}}{(max_i(y_i) - min_i(y_i))}, \text{ where RMSE}=\sqrt{\frac{\sum_i (y_i - \hat{y_i})^2}{n}}.$$

Mean Absolute Percentage Error:

$$\text{MAPE} = \frac{1}{n} . \sum_i \left(\frac{|(y_i - \hat{y_i})|}{y_i} \right).$$

Symmetric Mean Absolute Percentage Error:

$$\text{sMAPE} = \frac{1}{n} . \sum_i \left(\frac{|(y_i - \hat{y_i})|}{0.5*(y_i + \hat{y_i})} \right).$$

Implementation: The CMA workflow is implemented in Python using several deep learning libraries, such as TensorFlow, and Keras for VGG16 and ResNet-50 implementation. The entire implementation has been done on Google Colab, which was run on Intel(R) Xeon(R) Processor 2.20GHz. Image classification ($\mathbf{S_1}$) was implemented on GPU with specifications NVIDIA Tesla T4 16GB GDDR6 memory, and image segmentation ($\mathbf{S_2}$ and $\mathbf{S_5}$) on GPU with specifications NVIDIA L4 24GB GDDR6 memory.

We used the browser-based Julius AI tool [10] to convert data tables to their corresponding text summary to showcase the usability of CMA.

IV. Results and Discussion

Our system evaluation results are provided in Table I. The accuracy for $\mathbf{S_5}$ for Indian dataset (Fig. 3) is 83.84% and 94.38% for AP and AP75, respectively. We observe that

This article has been accepted for publication in IEEE Geoscience and Remote Sensing Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LGRS.2024.3444600

TABLE I
STEP-WISE AND OVERALL SYSTEM EVALUATION OF CMA OF CASE STUDIES OF MAINLAND USA

Description	Evaluation Metric	Value
$\mathbf{S_1}$: Image Classification	Classification Accuracy (at epoch 20)	93.65% (VGG16) 94.32% (ResNet-50)
$\mathbf{S_2}$: Chart Component Extraction	AP AP75	80.78% 83.16%
$\mathbf{S_3}$: Text Extraction	Accuracy	96%
$\mathbf{S_5}$: State/ Political Unit Segmentation	AP AP75	91.35% 97.32%
CMA System: Extracted Data (for Case Studies)	nMAE nRMSE MAPE sMAPE	0.032 ($\mathbf{C_{con}}$) 0.062 ($\mathbf{C_{con}}$) 0.19% ($\mathbf{C_{con}}$) 0.198 ($\mathbf{C_{con}}$)
	nMAE nRMSE MAPE sMAPE	0.058 ($\mathbf{C_{sep}}$) 0.098% ($\mathbf{C_{sep}}$) 0.27% ($\mathbf{C_{sep}}$) 0.352 ($\mathbf{C_{sep}}$)

ResNet-50 and VGG16 perform comparably, where ResNet-50 performs slightly better based on classification accuracy. Hence, we can choose an appropriate classifier based on the available resources and the compute requirements of DL models with either a higher number of parameters (VGG16) or more layers (ResNet-50). We also observe that Detectron2 provides better results for state segmentation ($\mathbf{S_5}$) than chart component extraction ($\mathbf{S_2}$), which could be attributed to the persistence of patterns in state boundaries than the chart layout. Our choice of PaddleOCR is appropriate given its high accuracy of text recognition.

For our case studies, we observe that the errors are relatively low for all choropleth images, implying the accuracy of CMA workflow. The errors are lower for $\mathbf{C_{con}}$ images than $\mathbf{C_{sep}}$ ones, thus accepting our hypothesis h_{class}. The information loss due to quantization in $\mathbf{C_{sep}}$ choropleth images causes an increase in the errors.

Overall, the metrics selected are appropriate for our system evaluation task at both step- and overall system-level.

We generated the text summaries for the merged data tables of our case studies using the $\mathbf{C_{con}}$ choropleths. The output summaries generated using the Julius AI tool are given below. For the case study of multiple state-wise climate variables of mainland USA in 2017:

> *The precipitation varies significantly across states, with some states experiencing much higher average precipitation than others. States like Alabama show notably high precipitation, while states like Arizona have much lower values.*
>
> *As the temperature increases, there is a tendency for precipitation to increase as well, although the relationship is not very strong. This suggests that warmer states might experience slightly higher precipitation, but other factors likely play significant roles in determining precipitation levels.*
>
> *There are noticeable differences between day and night temperatures across most states, with day temperatures generally being higher. The variation between day and night temperatures can be quite significant in some states, indicating strong diurnal temperature variations.*

For the case study of time series of annually averaged state-wise precipitation values of USA during 2017-2021:

> *States like Arizona and Nevada have seen significant increases in precipitation over the five-year period, while states such as Alabama and Louisiana have experienced notable decreases. States with arid climates like Arizona, Nevada, and New Mexico, which typically have low humidity and less precipitation, have shown significant increases. This could be due to unusual weather patterns or possibly climate change effects, leading to more erratic rainfall. States with continental climates (North Dakota, Minnesota, Wisconsin) and humid subtropical climates (Alabama, Louisiana) showing decreases could be influenced by various factors including changes in oceanic currents, atmospheric conditions, or human activities such as land use changes.*

The current limitation of CMA is that it is an integrated end-to-end system that could be replaced by a single model. The legend class of quantized colors which resembles the continuous color bar is yet to be studied. $\mathbf{S_3}$ works for non-negative numerical text only, which is to be rectified in future.

V. CONCLUSIONS

In this work, we have addressed the gap in the analytics of different classes of choropleth map images, based on the color legend type. We have proposed the Choropleth Map Analytics (CMA) system with a six-step workflow involving several deep learning models. The implementation calls for a novel algorithm for color-to-data mapping using the color data extracted from the images. We have demonstrated the usability of our system using climate research data, where multiple maps are analyzed together to create a single data table. The data table is further used for text summarization. This adds to the work on reverse engineering charts, focussing on more complex geovisualizations.

ACKNOWLEDGMENT

We thank IIIT-B, GVCL, and reviewers for their support.

REFERENCES

[1] J. Choi, S. Jung, D. G. Park, J. Choo, and N. Elmqvist, "Visualizing for the non-visual: Enabling the visually impaired to use visualization," in *Computer Graphics Forum*, vol. 38, no. 3. Wiley Online Library, 2019, pp. 249–260.

[2] C. Chen and Z. Liu, "The state of the art in creating visualization corpora for automated chart analysis," in *Computer Graphics Forum*, vol. 42, no. 3. Wiley Online Library, 2023, pp. 449–470.

[3] S. Chang, D. Palzer, J. Li, E. Fosler-Lussier, and N. Xiao, "MapQA: A dataset for Question Answering on Choropleth Maps," in *Poster in NeurIPS Workshop on Table Representation Learning*, 2022.

[4] J. Poco, A. Mayhua, and J. Heer, "Extracting and retargeting color mappings from bitmap images of visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 637–646, 2017.

[5] A. Mayhua, E. Gomez-Nieto, J. Heer, and J. Poco, "Extracting visual encodings from map chart images with color-encoded scalar values," in *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. IEEE, 2018, pp. 142–149.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of 3^{rd} International Conference on Learning Representations (ICLR 2015)*. Computational and Biological Learning Society, 2015, pp. 1–14.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[8] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019, *Last accessed on May 15, 2024*.

[9] C. Li, W. Liu, R. Guo, X. Yin, K. Jiang, Y. Du, Y. Du, L. Zhu, B. Lai, X. Hu *et al.*, "PP-OCRv3: More attempts for the improvement of ultra lightweight OCR system," *arXiv preprint arXiv:2206.03001*, 2022.

[10] Julius AI, "Julius AI | Your AI Data Analyst," https://julius.ai/, last accessed on May 15, 2024.