Data Extraction Pie Chart Images

Ayush bhagta *IIT2019501* | Mohammad Monish *IIB2019033* | Sanket Kokude *IIB2019034* | Harshit Kumar *IIB2019035* | Viful Nirala *IIB2019036*

***Abstract*—Pie charts are commonly employed in digital documents to display relevant numerical data due to their perceptual advantages. For further processing of pie chart data, automatic extraction of underlying slice data is required. In this research, an unique technique for detecting pie charts in document images and extracting chart data is provided. A Region-based Convolutional Neural Network (RCNN) model has been trained with 2D pie chart pictures to recognise pie charts in documents. Then, using picture gradients as one of the major features, different slices of a pie chart are evaluated and the values of different slices are computed.**

I. Introduction

With the emergence of high-quality cameras with mobile devices and advancements in document scanners, scanning document images using these devices has become common practice. In documents, pie charts are a useful technique to express a variety of numerical facts. Due of their perceptual advantages over textual representation, pie charts are particularly popular. Each slice's arc length (and thus its centre angle and area) in a pie chart is proportionate to the quantity it represents. Charts are used in scientific research publications to illustrate various types of experimental data and analysis reports. Pie charts are also widely used as a graphical approach to convey information in presentation presentations. The primary focus of study is on developing an efficient image processing-based solution for identifying pie charts and extracting slice data from document images.

Because of the many different shapes and representations of pie charts, this study focuses on the processing of pie chart images. The suggested technique assumes that each slice is uniformly coloured with a single colour and does not have any hatch patterns or textures. The data extraction method of the proposed algorithm generates a data table from an input chart image, with each row representing the quantity of each pie chart slice. Following data extraction, the chart data can be utilised for further research or other representations of the chart which may involve addition of new data, deleting current data, modifying the graphic's design, or reusing chart data A block diagram of the proposed algorithm is shown in Fig. 1.

II. Pie Chart Identification

A machine learning model named VGG-19 was used to automatically classify and identify chart regions in document images. The VGG-19 is a Convolution Neural Network that classify the image into different categories. A pre-trained VGG-19 model was utilised for chart classification, and the model was trained with a pie chart, barchart, venn diagram and other chart images for classification. The input image size for training VGG-19 model is 224x224. Our database consisted of over 4000 images of various types of charts including bar graphs, piecharts and also some images of 3D piecharts.

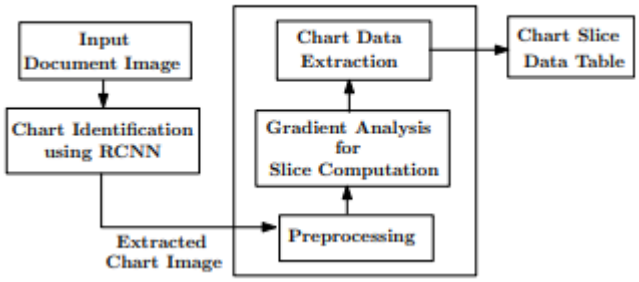

Fig. 1. Block diagram of the proposed data extraction process from pie charts.

III. Data extraction from 2D pie chart

A unique image processing-based approach is used for automatic data extraction from pie charts . For this aim 2D pie charts have been examined. The following are the specifics of the suggested algorithm for extracting the underlying data from a 2D pie chart image.

A. Preprocessing

In document image processing, image de-noising is very important. Due to the quality of the document paper, lighting circumstances, or while capturing photos during scanning of a document by a mobile device or document scanners, noise may be introduced into document images. Salt-and-pepper noise, background noise, marginal noise, clutter noise, and other sorts of noises can be found in document photographs. It is assumed that the salt-and-pepper noise dominates in the region corresponding to the pie chart image. The spatial averaging with a Gaussian smoothing, which is an ideal low pass filter, was used to reduce salt-and-pepper noise as a preprocessing step in this work.After that, the denoised image is transformed to its grayscale image.

B. Gradient analysis

Different solid colours represent different slices of a pie chart. As a result, the common boundary between two neighbouring slices is first approximated by examining the chart image's local gradients in the region. To retrieve all the slice information from a pie chart, gradients are computed from the grey image.. We have compared Laplacian, sobelX, sobelY, sobelCombined filters to get gradients images as

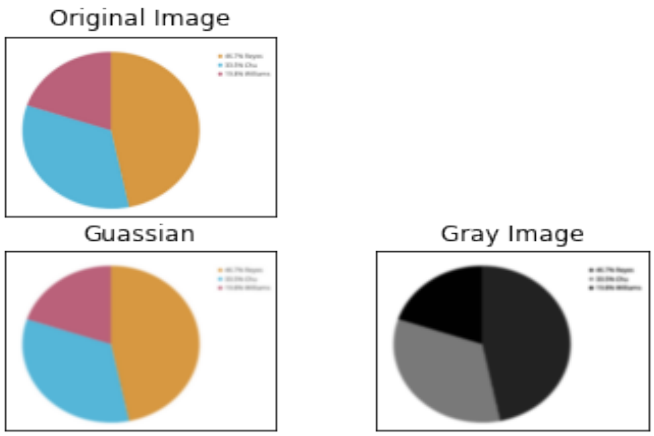

Fig. 2. Pre-processing: (a) Original (b) Guassian (c) Gray Image.

shown in Fig. 3. By observation, the sobelCombined filter gives the best gradient image. Therefore we have used gradient image produce by sobelCombined filter.

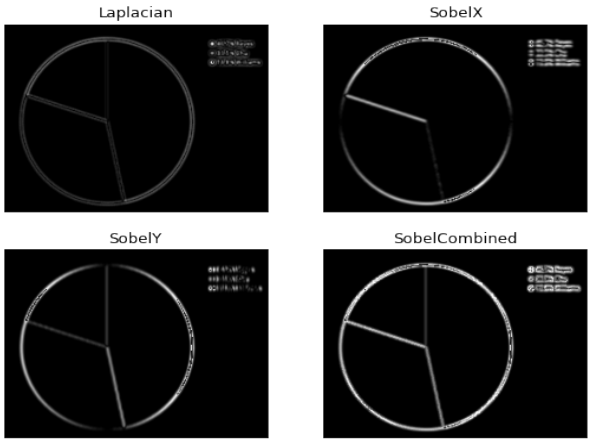

Fig. 3. Gradient analysis: (a) Laplacian (b) SobelX (c) SobelY (d) SobelCombined.

The gradient image is then binarized by using a simple thresholding technique with a suitable threshold value. The gradients of the input pie chart image are shown in Fig. 4(a). The binarized gradient image contains the pie chart boundary, common boundaries of different adjacent slices, text information of each slice, as well as other unwanted noise (refer to Fig. 4(a)). Thus, it is required to remove these unwanted noises from binarized gradient images. For this purpose,we have used connected component analysis(CCA) based algorithm on the binarized gradient image and the size of each connected component is determined. Here, it is assumed that the size of the component containing the pie chart is greater than all the other components in the binarized image. CCA alorithm provides the label information and size of each connected component and based on the component size the largest component is extracted. Fig. 4(c) depicts the binarized gradient image after small component removal.

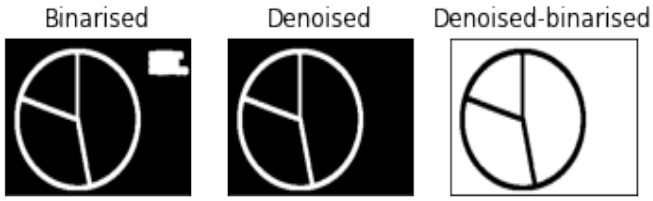

Fig. 4. Gradient analysis: (a) Binarised image (b) Denoised image (c) Denoised-binarized image.

Now after analysis we have found that there is while removing undesired components there is a possibility that text inside the pie-chart can be found connected by CCA algorithm.In order to remove this possibility we first apply erosion on input image to detect undesired component.This allows us to seperate any unwanted component from pie chart.However after the output from function is recieved we again apply dilation so that further calculation is not affected.

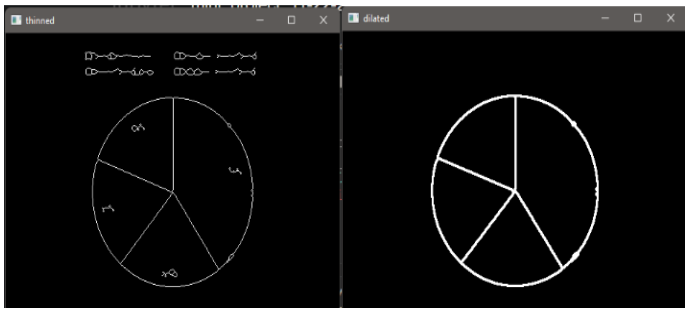

Fig. 5. Result of 2D pie chart: (a)Input image and (b) The extracted slice data table using the proposed algorithm.

C. Slice computation

Once the binarized chart image containing the pie chart boundary and common boundaries of different adjacent slices is obtained, the next step is to identify each slice of the pie chart. .For automatic data extraction of the pie chart, such boundary touching components should be removed. For that, consider the inverted image and carry out a connected component analysis (CCA). After CCA, we get each slice component as shown in Fig. 5 . From the image, it can be observed that the image only contains different slices of the pie chart image.

D. Chart Data Extraction

To extract data from the pie chart and compute the actual slice values, consider the boundary component removed image which contains only the chart slices. For data extraction, carry out a connected component label algorithm on the image which gives us the total number of pixels in each slice then we calculate the total area of pixels in pie chart by adding the

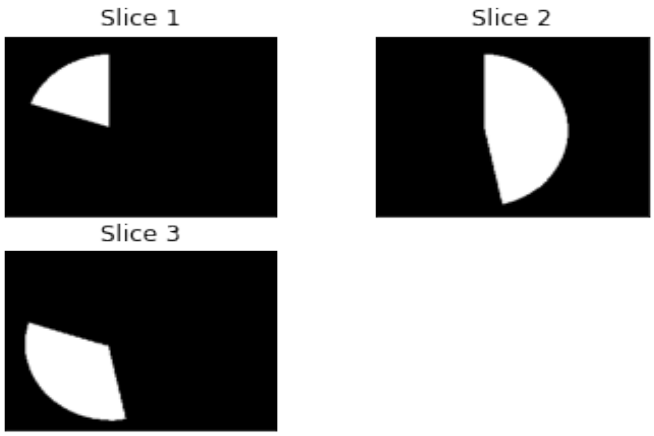

Fig. 6. Slice computation

number of pixels in each slice. Let the total area of pixels be totalArea. We are also computing the centroid of each slice and then we determine the colour of the centroid which will be the colour of that slice which is in the original pie-chart. We are labeling the percentage of each slice by its colour. To compute the percentage of each slice:

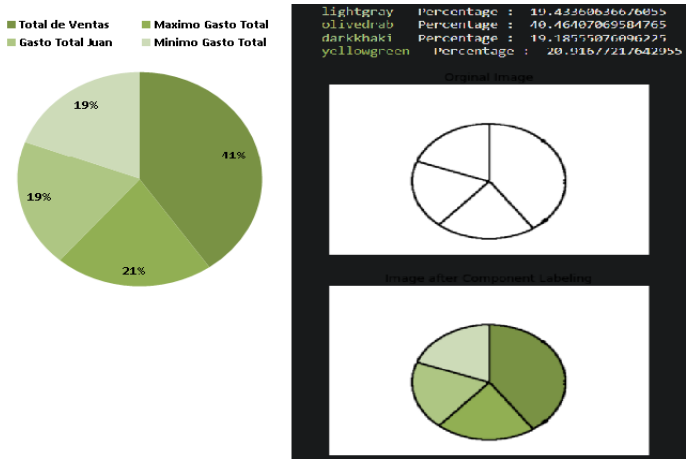

Fig. 7. Result of 2D pie chart: (a)Input image and (b) The extracted slice data table using the proposed algorithm.

- Determine the area of each slice. Let the area of each slice be sliceArea.
- Compute the percentage of the slice as:

$$percentage = (sliceArea * 100)/totalArea$$

- Display its percentage along with the respective color label.

IV. Data extraction from 3D pie chart

3D pie charts are also often used to depict a variety of data kinds. A 3D pie chart needs be handled independently to extract its slice data due to the presence of 3D structures and their perspective projected depiction. A unique technique is proposed for automatic processing of such 3D pie charts, which automatically determines if an input pie chart is a 2D pie chart or a 3D pie chart before computing the 3D pie chart slice data. The following is a description of the suggested algorithm.

A. Preprocessing

We used spatial averaging with gaussing smoothing to denoise the 3D pie chart image. The denoised image is then converted to its grayscale image. The gradient picture will be computed after that, and we will compare Laplacian, SobelX, SobelY, and SobelCombined filters to obtain gradient images. The sobelCombined filter, by far, produces the best gradient image. As a result, we used a gradient image created by the sobelCombined filter. After that, the gradient image is binarized using a basic thresholding technique and a suitable threshold value. The undesired noises are then removed from the binarized gradient image using a connected component analysis (CCA) based algorithm.

B. Removing 3D elements from pie chart

In 3D representation of pie charts,pie charts have extra boundary component for 3D representation as shown in Fig . These boundary components are not present in 2D pie charts, and this fact can be used to differentiate between 2D and 3D pie charts. In 3D pie chats we need to remove the extra boundary part for better accuracy. To identifying between 2D and 3D pie charts,a algorithm is proposed which automatically identify whether a chart image contains such unnecessary components or not. First we will get the centroid of the pie chart by pop up window which will prompt the user to select the centroid on the pie chart. Then by using CCA based algorithm we are computing the centroid of each individual component and height of the pie chart. Then we will get the approximate radius by dividing the height by 2 . Then compute the distance between the centroid of the pie chart image and centroid of each component. Then if the distance is greater than approximate radius for any component than that component is the thickness of the 3D pie chart and we will ignore that component.

C. Chart data extraction

After removing 3D elements from 3D pie chart, data extraction of the 3D chart slices is performed on this image using the similar algorithm used for 2D chart data extraction to compute the data of 3D chart slices.

V. Error Analysis

Now we will do the error analysis for the 2D and 3D sample image shown in the report. OSP=Originalslicepercentage
ESP=ExtractSlicepercentage
N=Total number of slices

$$AverageError : \sum |OSP - ESP|/N$$

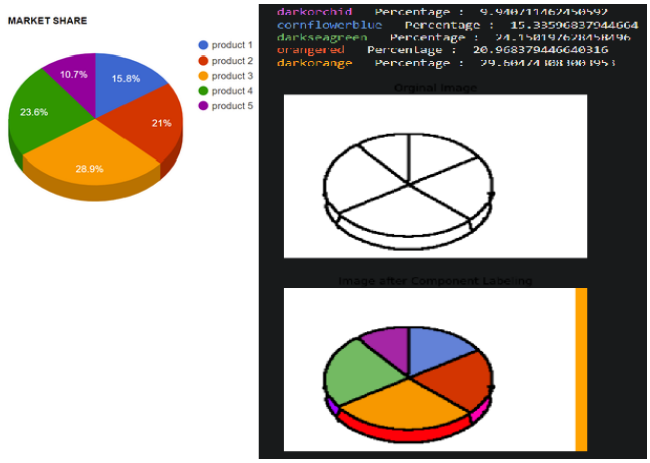

Fig. 8. Result of 3D pie chart: (a)Input image and (b) The extracted slice data table using the proposed algorithm.

For 2d image:
((19.433-19)+(41-40.46)+(19.18-19)+(21-20.9))/4=0.31325

For 3d image:
((10.7-9.94)+(15.8-15.33)+(24.1-23.6)+(21-20.96)+(29.6-28.9))/5=0.494

VI. Dataset

We used an already available database by Devi sandeep which he created for a similar project which consisted many images of each type of chart like bargraphs, venn diagrams, piecharts etc.

Plot type	Count
BarGraph	528
VennDiagram	364
PieChart	355
ScatterGraph	335

Plot type	Count
TreeDiagram	297
FlowChart	293
Map	276
ParetoChart	329

Plot type	Count
BubbleChart	311
LineGraph	300
AreaGraph	299
NetworkDiagram	321
BoxPlot	312

We also added 100 images of 3D piechart to the above database.
Dataset link: https://drive.google.com/drive/dataset

VII. Conclusion

A VGG-19 model has been used for identification of charts and classify them under different chart classes like piechart, bargraph , linegraph amd others. Implemented data extraction part for pie-charts where image gradients of a pie-chart will be computed to get different slices of the chart followed by automatic extraction of underlying data from 2D and 3D pie chart. The proposed algorithm also presents a new approach to remove the 3D element from 3D pie chart .We have tried to resolve the issues of erroneous slice data extraction due to the small difference in gray values of adjacent slices of a 3D pie chart.

References

[1] D. Jung, W. Kim, H. Song, J. Hwang, B. Lee, B. Kim and J. Seo. ChartSense: Interactive Data Extraction from Chart Images. In Proc. of the CHI conference on Human Factors in Computing Systems (ACM),
[2] M. Savva, N. Kong, A. Chhajta, L. Fei-Fei, M. Agrawala, and J. Heer. ReVision: Automated classification, analysis and redesign of chart images.. In Proc. of the 24th Annual ACM Symposium on User Interface Software and Technology(UIST'11), pp. 393–402, 2011.
[3] J. Gao, Y. Zhou, and K.E. Barner. View: Visual information extraction widget for improving chart images accessibly. In Proc. of the 19th IEEE International Conference on Image Processing (ICIP'12).
[4] Y. Liu, X. Lu, Y. Qin, Z. Tang, and J. Xu. Review of chart recognition in document images.. In IST/SPIE Electronic Imaging, pp. 865410–865410, 2013.
[5] W. Huang, R. Liu, and C. L. Tan. Extraction of vectorized graphical information from scientific chart images.. In Proc. of the 9th International Conference on Document Analysis and Recognition (ICDAR'07), pp. 521–525, 2007.
[6] W. Huang, C. L. Tan, and W. K. Leow. Model-based chart image recognition.. In International Workshop on Graphic Recognition (GREC'03), pp. 87–99, 2003.
[7] https://drive.google.com/drive/folders/1StL03yvgJsl3xkCxWbYV9x7DPI6q19-j?usp=sharing