VIETTEL DIGITAL TALENT

viettel

CHART2TABLE WITH MULTISTAGE AND END2END DEEP LEARNING MODEL

Mentor: Nguyen Quang Tuan,
Pham Thai Hoang Tung

Article: Nguyen Dac Hoang Phu

viettel
Theo cách của bạn

Table of contents

01 - Problem Definition

Definition

- Chart images can be easily found in news, web pages, company reports and scientific papers,... Automatic analysis of these data can bring us huge benefits, including scientific document processing, automatic risk assessment based on financial reports, and reading experience enhancement for visually impaired people.
- Extracting the raw data table from chart images is the key step for understanding the chart content, which would lead to better analysis of related documents
- Input is chart image and output is table which has information about this chart (include information follow x axis, y axis)

DEFINITION

USE CASES

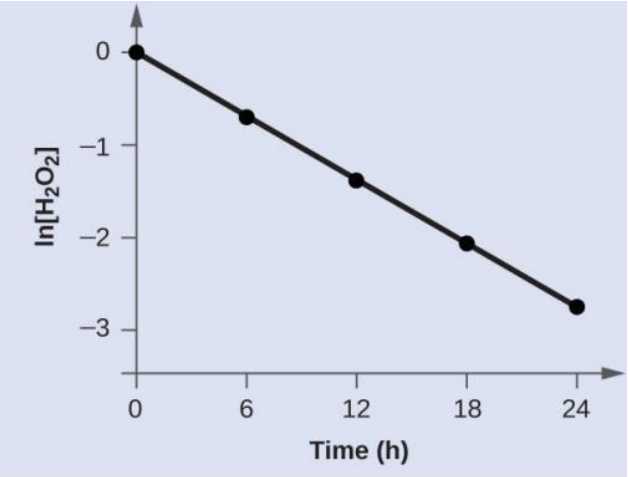


Chart input

Time	ln[H202]
0	0
6	-0,8
12	-1,4
18	-2,1
24	-2,8

Table output

- The goal is to improve STEM education for visually impaired students by utilizing text-to-speech technology to provide access to chart information in an auditory format, and developing a QA model to facilitate interaction and learning.
- In office-related fields, charts contain valuable information in reports. However, summarizing and describing this information can be a tedious and time-consuming task. To tackle this issue, the chart2table application can be used to automatically generate a summary describe for the chart.
- In finance-related fields, with the chart2table application, we can incorporate additional QA models and train them to infer and discover insights within a specific type of chart.

02- Data Overview

EDA

The dataset includes 60,578 instances which are divided into 4 main types: Bar, Line, Scatter and Dot

Bar chart have 2 variants: Horizontal Bar, Vertical Bar. Class Horizontal Bar is imbalance

Dataset has no legend => don’t contain stack-bar chart, multiple line chart, multiple bar chart

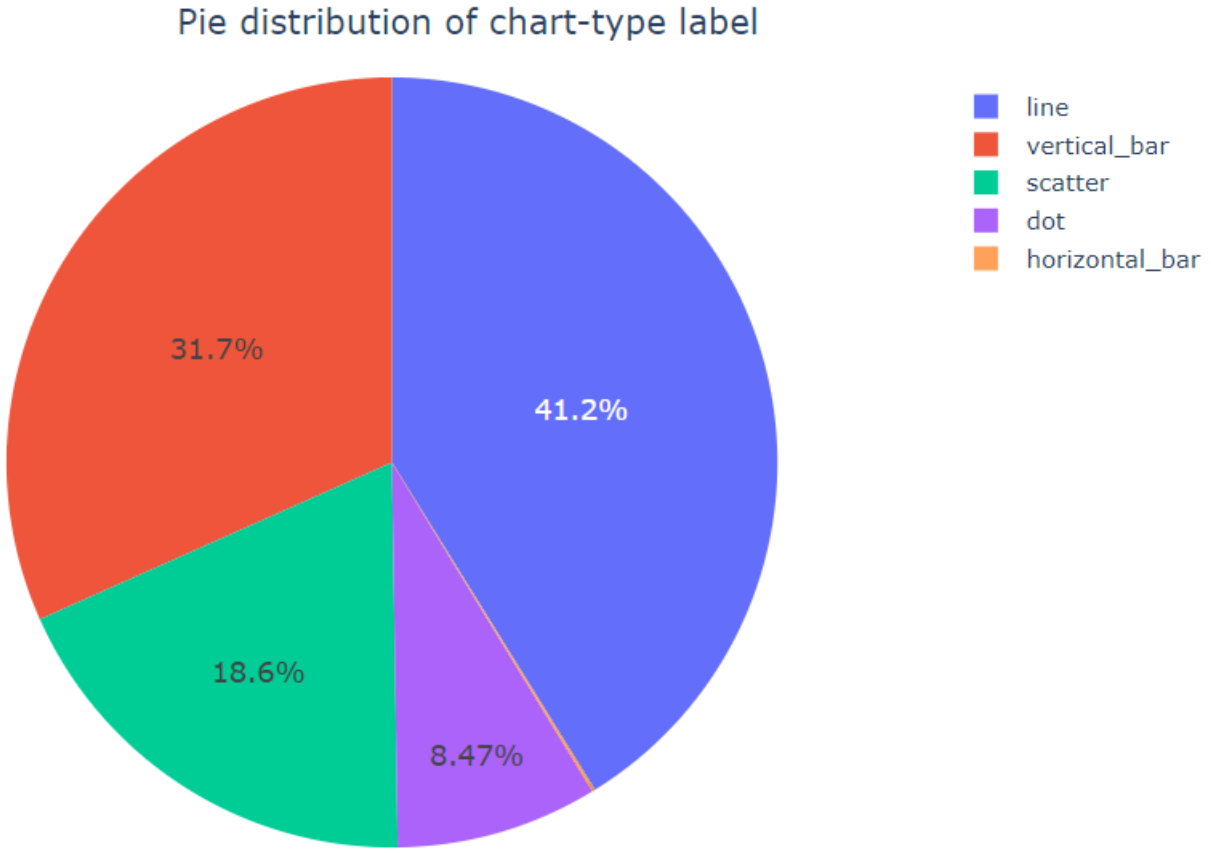
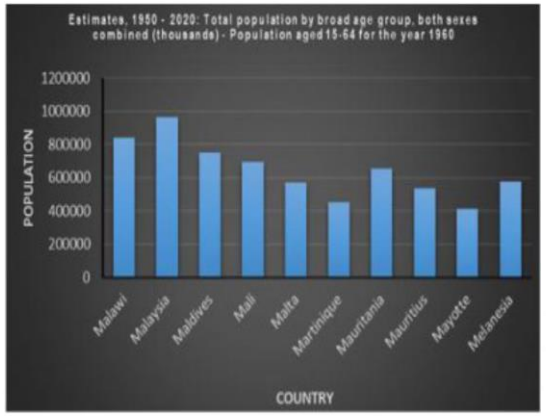
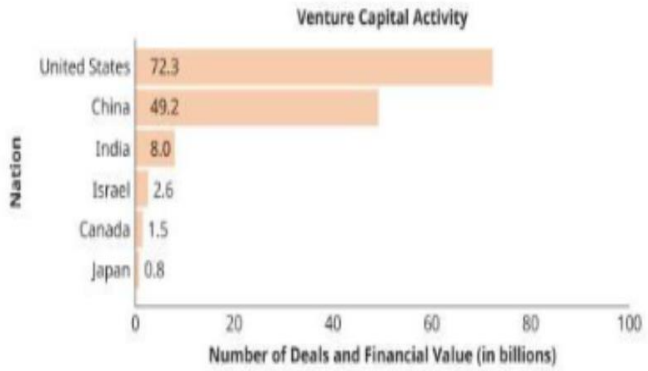


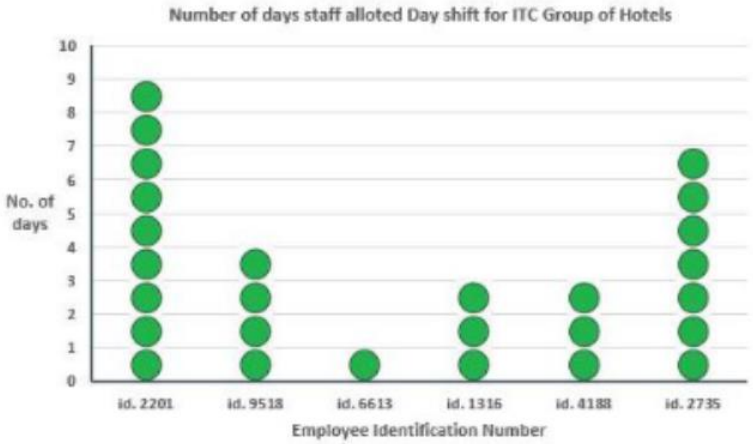
CHART SHAPE



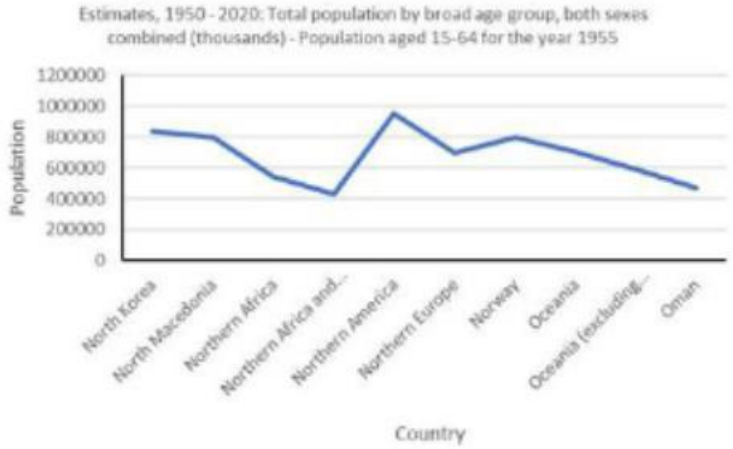
Vertical Bar



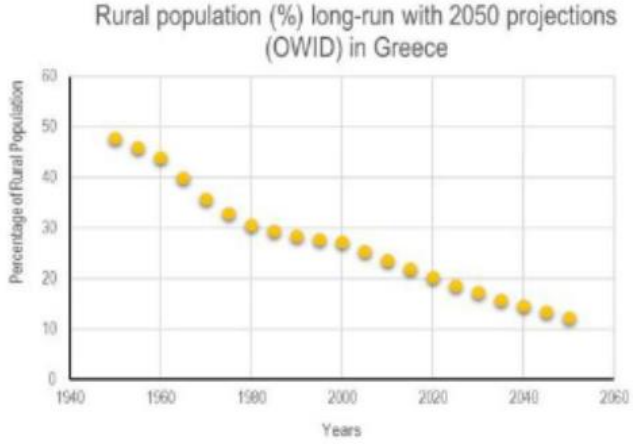
Horizontal Bar



Dot Plot



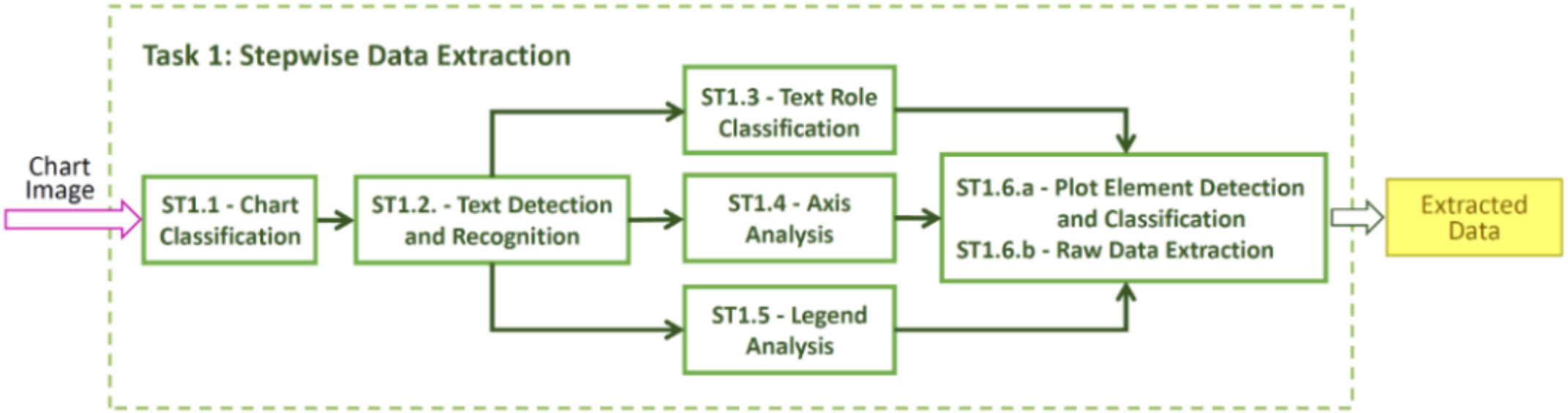
Line Chart



Scatter Chart

03- Approach

OVERVIEW



TYPICAL APPROACH

COMMON APPROACH

(RELATED WORK)

Use rule-based hand-craft

- Detect strong component and axis of chart
- Only handle: line, bar and pie
- Some model: chartSense, Revision,...

Use multistage model

- 3 main stage: object detection, text recognition and data extraction
- Affected by propagation error
- Some model: chartOCR, propose model for raster image,..

Use end2end deep learning model

- Typical backbone - transformer
- Encoder is task image understanding
- Fine specific task in Decoder
- Some model: Donut, Deplot

MY PROPOSE APPROACH

- First, I finetune donut to extraction information all of 5 chart
- But horizontal_bar chart has score 0.0 with metric definition (will be intro later), because this chart is imbalance and this role is almost negligible in dataset

First Phase

- Realizing that, I isolated the horizontal bar and used straight model deplot for this dataset (due to too little data to train).
- At this point, it leads to another issue where the scatter chart's score is too low, so I will separate the scatter chart and use a separate model to process it

Second Phase

- Realizing that the line-chart's results were bad, I continued to separate processing like scatter and using multi-stage architecture to lead to the following proposed model.

Third Phase

MY PROPOSE APPROACH

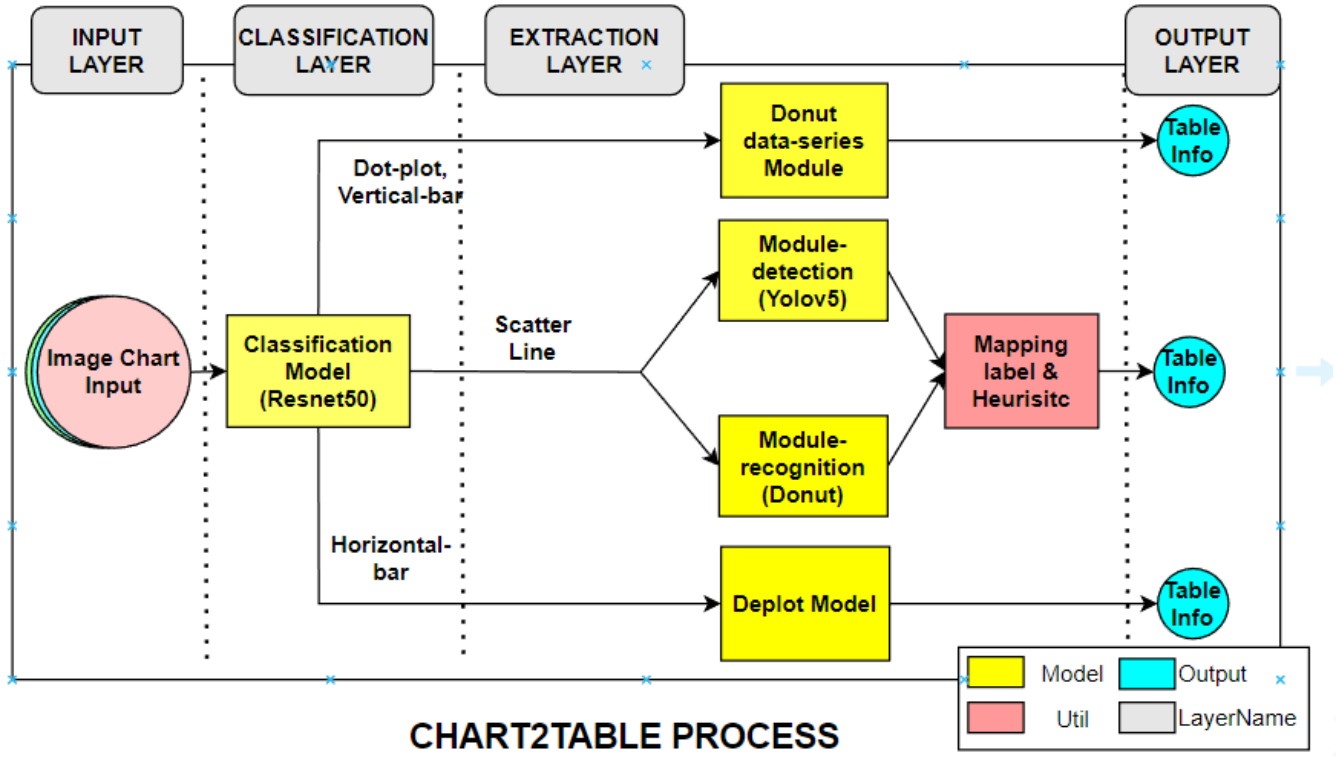


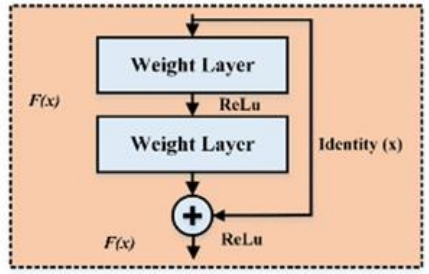
CHART2TABLE PROCESS

- Approach has 4 layer. Initially with the input chart image, first I pass it through a classification label to get chart-type
- Then with chart-type result, if chart-type result is horizontal_bar, I pass this image to pretrain deplot model. Else if chart-type is vertical_bar or dot plot, we pass it to my custom donut model to extract information.
- If chart-type is scatter or line, we pass it into multi-stage model include 3 stage: module object detection, module text recognition and module heuristic

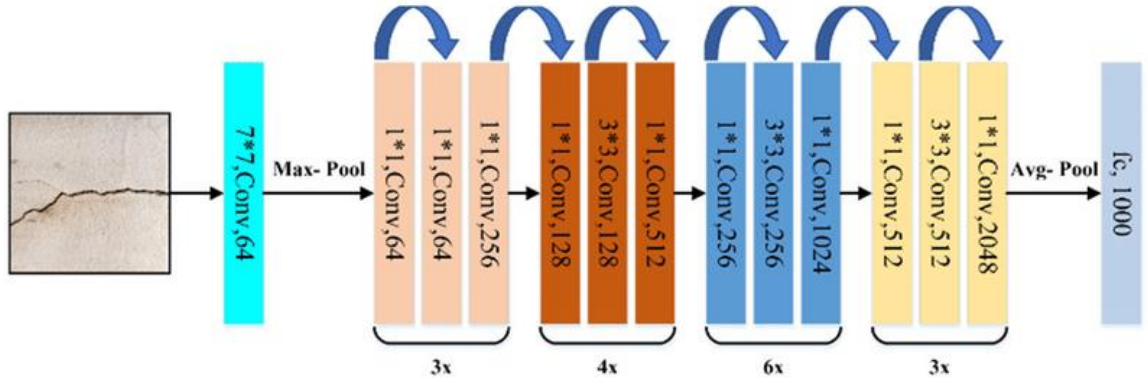
CLASSIFICATION LAYER

Classification stage I use **Resnet-50** is backbone, output will be 5 chart types corresponding to 5 chart types in the data

Faced with the **class imbalance issue** of horizontal-bar, I addressed it by augmenting the dataset with an external source with the same number of instances as vertical-bar.



Residual Learning Block



RESNET-50

DEPLOT

(For Horizontal-Bar)

Because **the imbalance in horizontal_bar** class and there is no external source which has annotation, we don't train for this class.

We use one model is trained and has been verified to be suitable for this type of chart, which is **deplot** model.

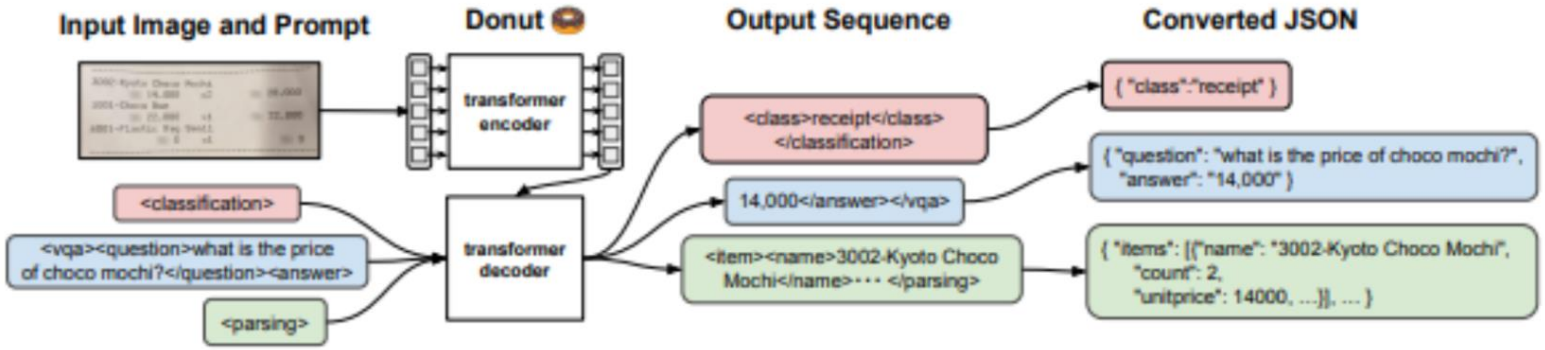
```
#get model from hugging-face
model = Pix2StructForConditionalGeneration.from_pretrained
        ('google/deplot').to(device)
processor = Pix2StructProcessor.from_pretrained('google/deplot')

#transform input to match format model input
inputs = processor(images=image, text="genData", return_tensors="pt"
inputs = {key: value.to(device) for key, value in inputs.items()}
```

```
#prediction and decode information
predictions = model.generate(**inputs, max_new_tokens=512)
data = processor.decode(predictions[0], skip_special_tokens=True)
```

PROCESSING STEP

DONUT

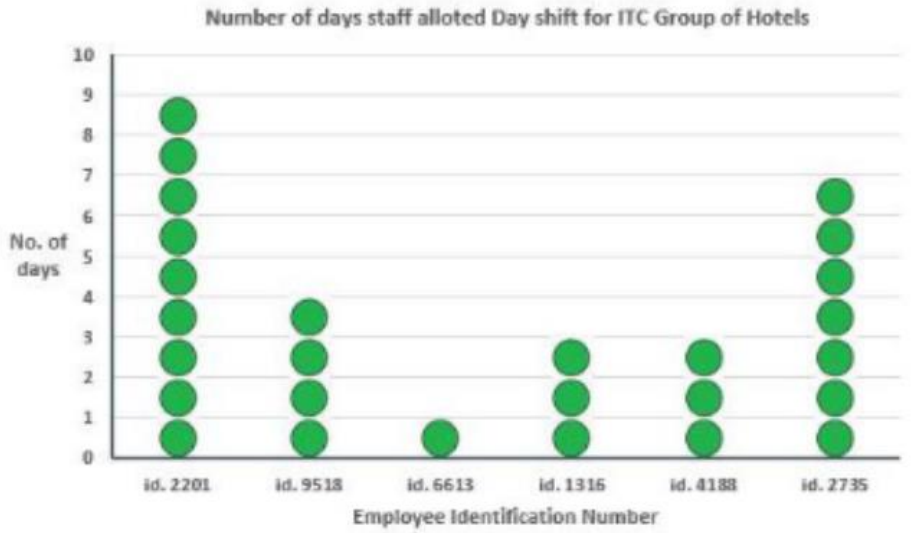
(For Vertical-Bar, Dot-plot)



To specify a specific task for the Donut, we need to use a **pair of tag values** that can be represented as follows:
<|PROMPT|><chartType><x_start>…<x_end><y_start>…<y_end><|PROMPT_END|>

Add field chart-type in this because I want the model to learn which type of chart will have its own way of reading and double check with classification stage

EXPLAIN



DONUT DATA SERIES MODEL

X: id.2201, id.9518, id.6613, id.1316, id.4188, id.2735

Y: 9, 4, 1, 3, 3, 7

MODULE TEXT-RECOGNITION MODEL (DONUT)

X-axis: id.2201, id.9518, id.6613, id.1316, id.4188, id.2735

Y-axis: 10,9,8,7,6,5,4,3,2,1,0

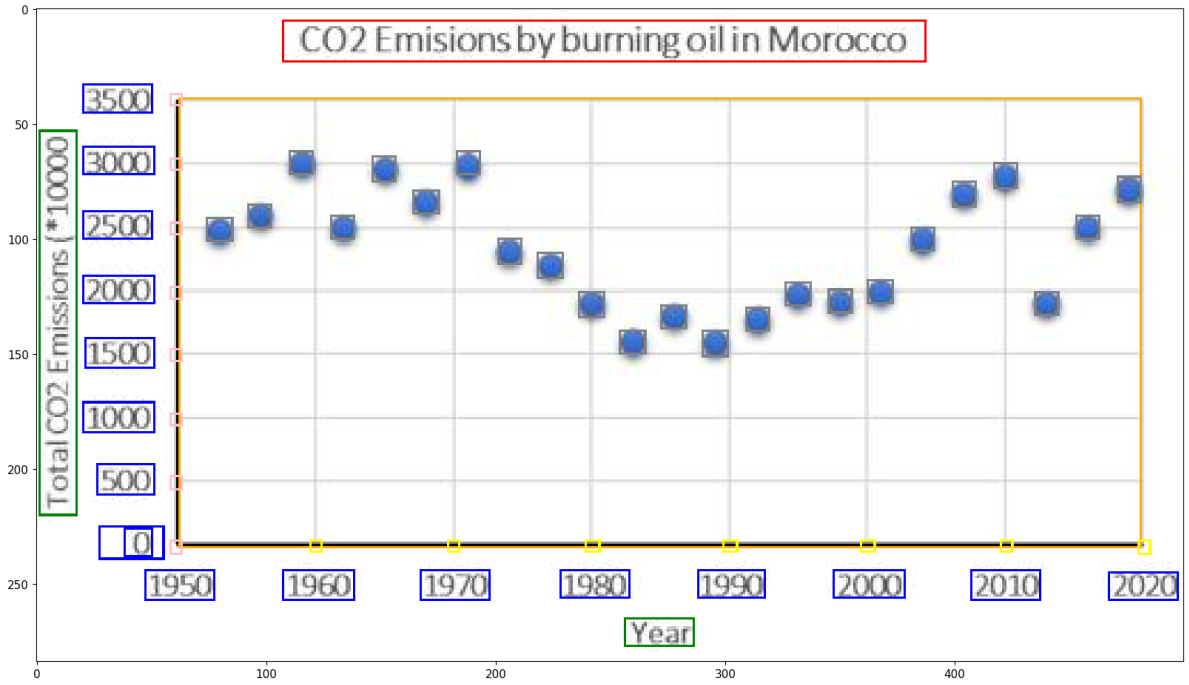
TITLE: number of days staff..

Y-title: No.of days:

X-title: Employee idenfi..

ABOUT MULTI-STAGE MODEL

(Object detection stage)



Result after detection bbounding box

- I use **Yolo-v5m** as backbone with format input of bounding box is (x,y,w,h).
- Preprocessing and some heuristic to handle annotation of data.
- Some instances need to be detected:
  - Gray bbox is a visual-point
  - Yellow bbox is x-axis marker
  - Pink bbox is y-axis marker
  - Orange bbox is plot-bb

ABOUT MULTI-STAGE MODEL

(Text Recognition Stage)

```
{'x_tick_label_mark': ['1950','1960','1970','1980','1990','2000',\
'2010','2020'],
 'y_tick_label_mark': ['3500','3000','2500','2000','1500','1000',\
 '500','0'],
 'chart_title_mark': ['CO2 Emisions by burning oil in Morocco'],
 'x_axis_title_mark': ['Year'],
 'y_axis_title_mark': ['Total CO2 Emissions (*10000']}
```

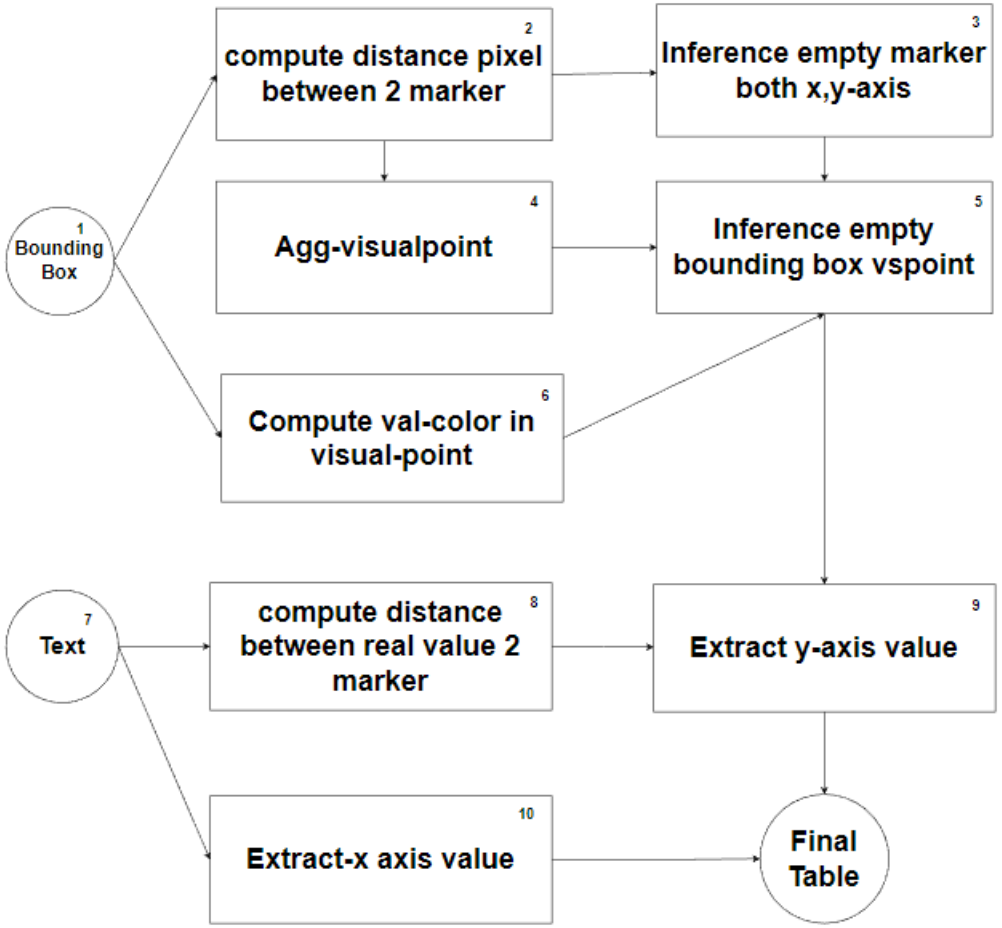
Text recognition corresponds to the image above

I finetune **Donut** for this stage. It will operate independently and not depend on the output of the text recognition module

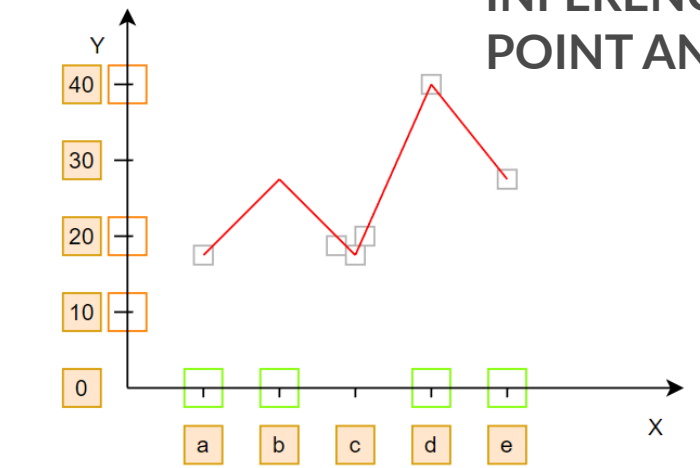
One of the benefits of **Donut** is its ability to accurately read text in any orientation, such as italics or vertical text. However, this comes with a trade-off of slower inference speed.

ABOUT MULTI-STAGE MODEL

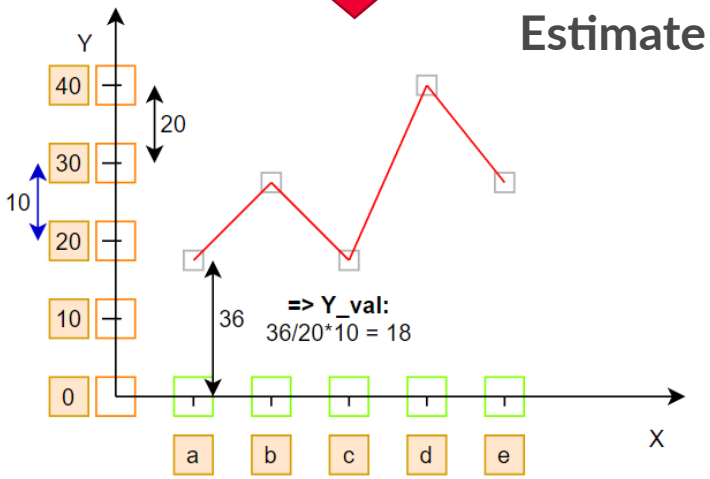
(Heuristic Stage for line-chart)



INFERENCE EMPTY POINT AND MARKER



Estimate value



04-Result

Metric Benchmark

Nlev for category value

$$\mathrm{NLev} = \sigma\left(\frac{\sum_i \mathrm{Lev}(y_i, \hat{y}_i)}{\sum_i \mathrm{length}(y_i)}\right)$$

Lev determine the minimum number of operations required to change one given string into another

Sigmoid Function

$$\sigma(x) = 2 - \frac{2}{1 + e^{-x}}$$

NRMSE for numeric value

$$\mathrm{NRMSE} = \sigma\left(\frac{\mathrm{RMSE}(y, \hat{y})}{\mathrm{RMSE}(y, \bar{y})}\right)$$

RMSE penalties are heavier than MSE if the error is high

Performance Benchmark

result /model	Val score↑	Hor score↑	Ver score ↑	Scatter score ↑	Line score↑	Dot score↑
(1)	0.33831	0.0	0.4783	0.0152	0.4275	0.9809
(2)	0.46141	0.54204	0.5405	0.06213	0.447	0.99076
(3)	0.64447	0.54204	0.57031	0.85818	0.43776	0.94915
(4)	0.6996	0.54204	0.53674	0.85818	0.69673	0.98957

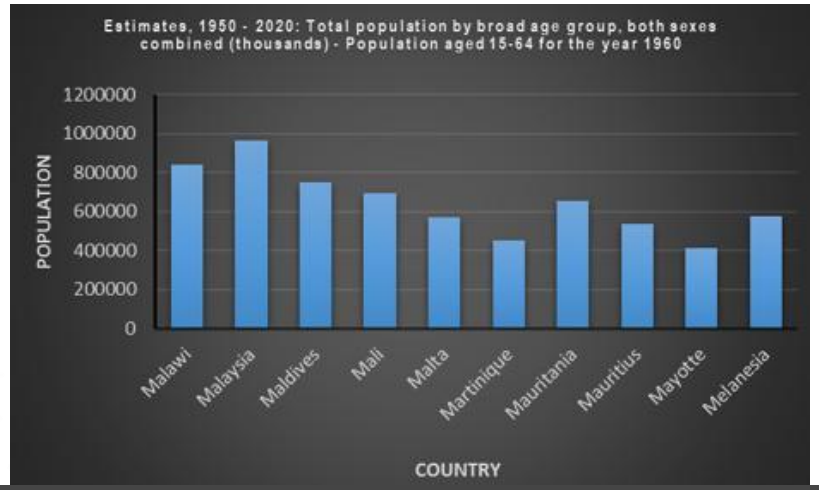
(1) - Phase 1: Donut for all chart-type
(2) - Phase 2: Deplot for horizontal-bar, Donut for the remaining charts
(3) - Phase 3: Deplot for horizontal-bar, Multi Stage for scatter and Heuristic for the remaining
(4) - Phase 4: Propose method

a

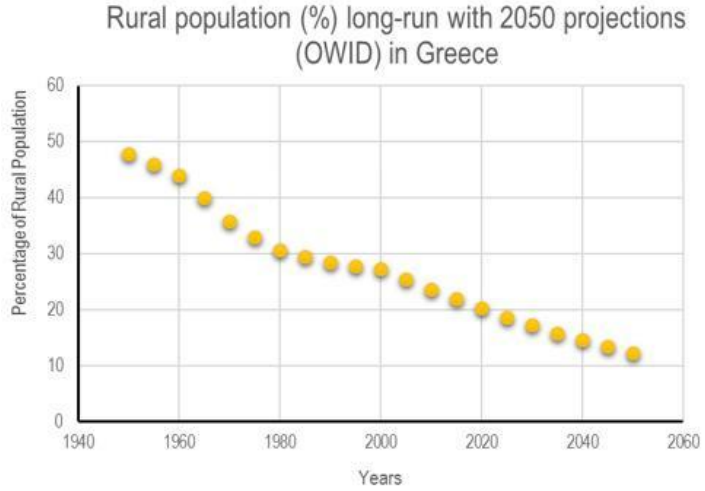
The **val_score** will be calculated as the average all instance with either Levenshtein or RMSE metric, depending on the type of the corresponding series (numeric or categorical value)

b

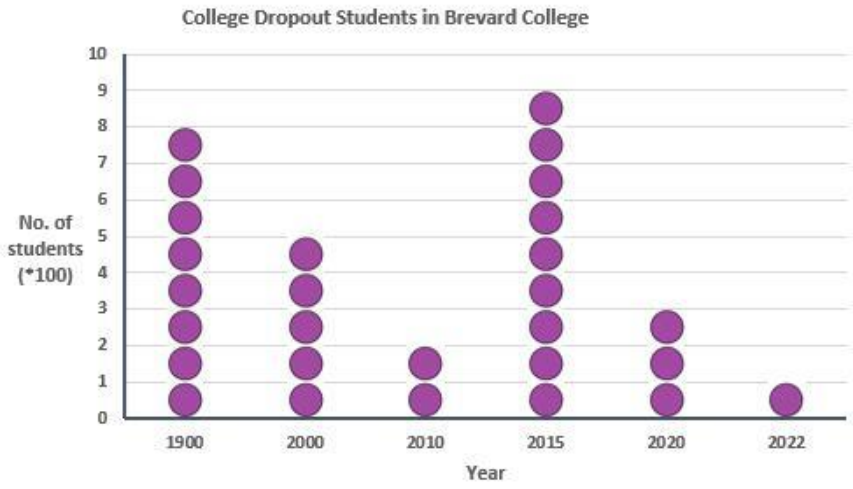
the scores for each chart will be calculated similarly to the **val_score**, value in range [0,1] but the instances will be based on the corresponding chart type



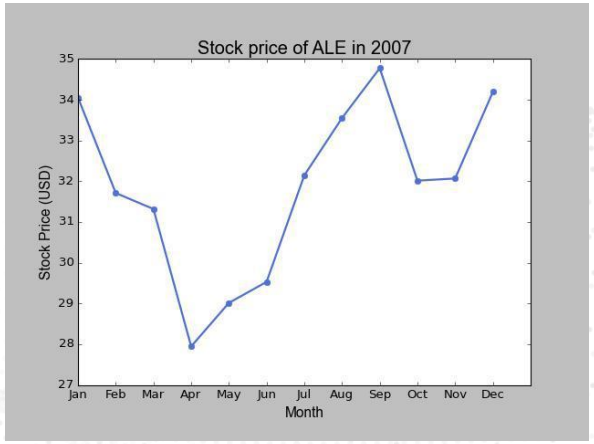
Malawi;Malaysia;Maldives;Mali;Malta;Martinique...	vertical_bar
839090.5;974390.5;743780.7;694390.5;574390.5;4...	vertical_bar



1950.359;1955.366;1960.324;1965.458;1970.431;1...	scatter
47.879;45.958;43.949;39.926;35.644;32.864;30.5...	scatter



1900;2000;2010;2015;2020;2022	dot
8;5;2;9;3;1	dot



Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec	line
33.84401368700768;31.826516711022244;31.387525...	line

Limitation

- Not performing task axis analysis can affect our ability to apply the technique to real-scenario charts, which often have multiple instance types.
- Rule-based algorithms that are hand-crafted by humans may fail in some cases when run automatically
- End-to-end models require a sufficiently long training time and a large dataset to produce good results. There were not enough resources or data to train Pix2Struct or Donut to meet initial expectations

05- Conclusion

SUMMARY

1. Within the time constraints of the project, I have been able to implement a complete approach for this task to process and provide good results, although there is still room for improvement.

2. The current method has performed well on the majority of the dataset with line and scatter charts, but there are still some special cases that require more error analysis to improve the heuristics.

3. The method has worked well on dot plots and horizontal bars, and with enough training on vertical bars and donut charts, it may produce good results.

Future Work

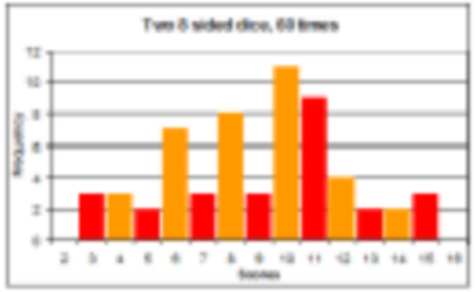
- Developing and improving the proposed heuristic algorithms to automatically adjust fixed hyperparameters can help to address more special cases
- Collecting additional data to predict more types of charts, such as pie charts, radar charts, or charts with legends, can help to adapt to more real-world contexts
- The system can be used for the next steps of building a Question Answering (QA) module on data extracted from charts or generating comments to describe the semantics of the corresponding charts
- Improve the transformer's inference time, or add modes for user feedback and chart interaction


LIVE DEMO

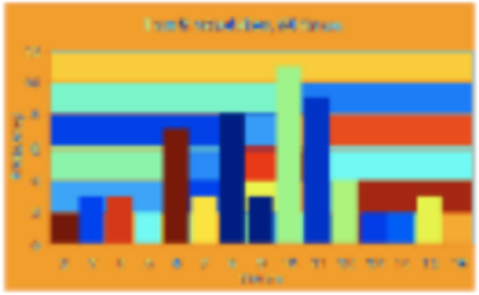

Q&A

Rule-based approach

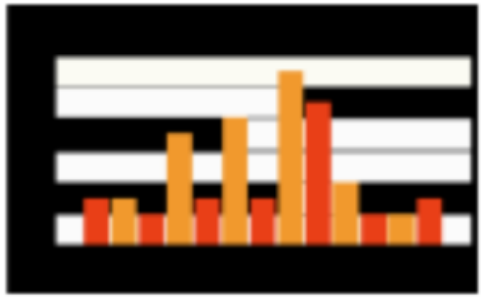
(Revision Model)



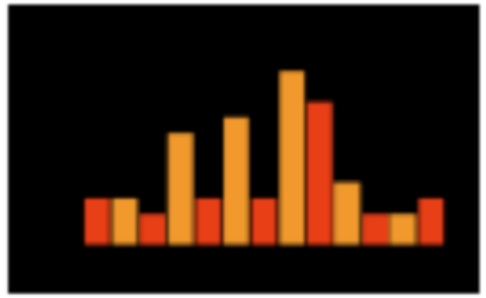
(a) Input chart



(b) Connected components

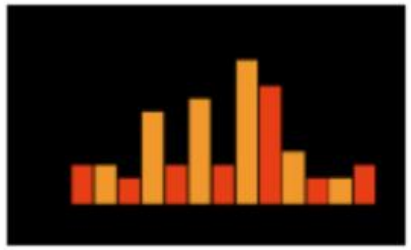


(c) Rectangular components

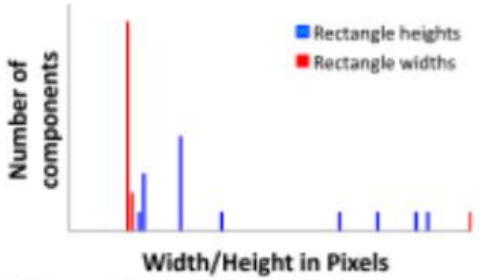


(d) Candidate Bars

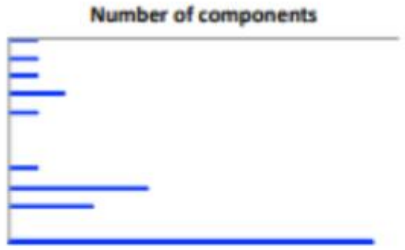
Detect strong component



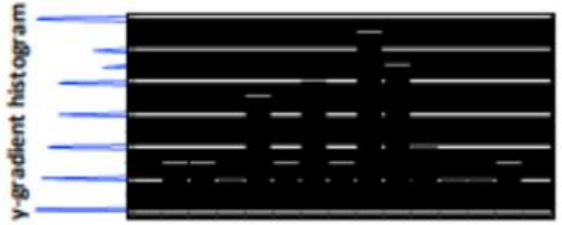
(a) Candidate bars



(b) Width and height histograms



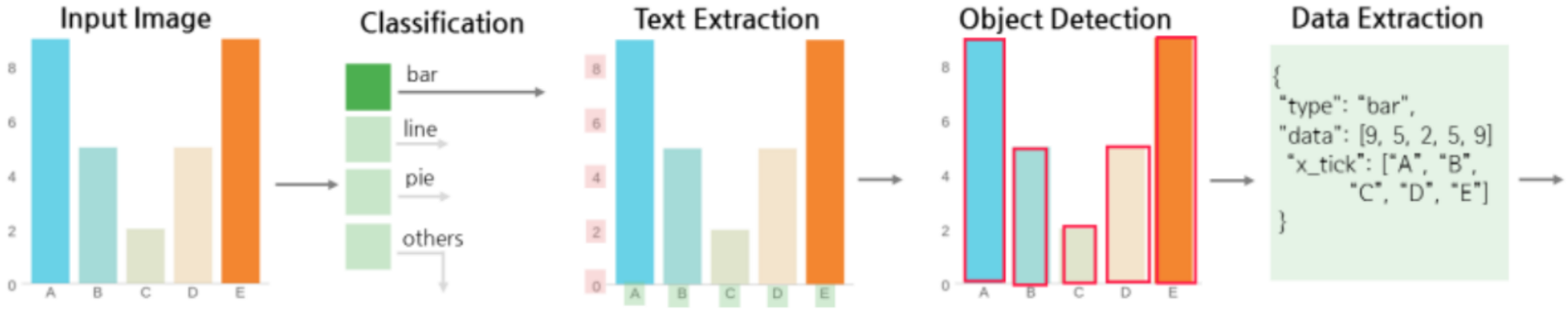
(c) Histogram of top and bottom y-values of candidate bars



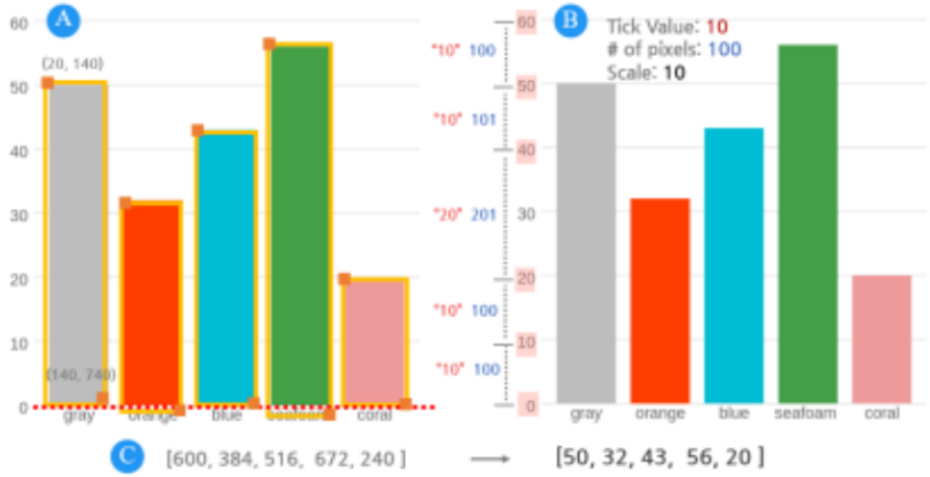
(d) y-gradient image

Detect axis

Multi-stage Model

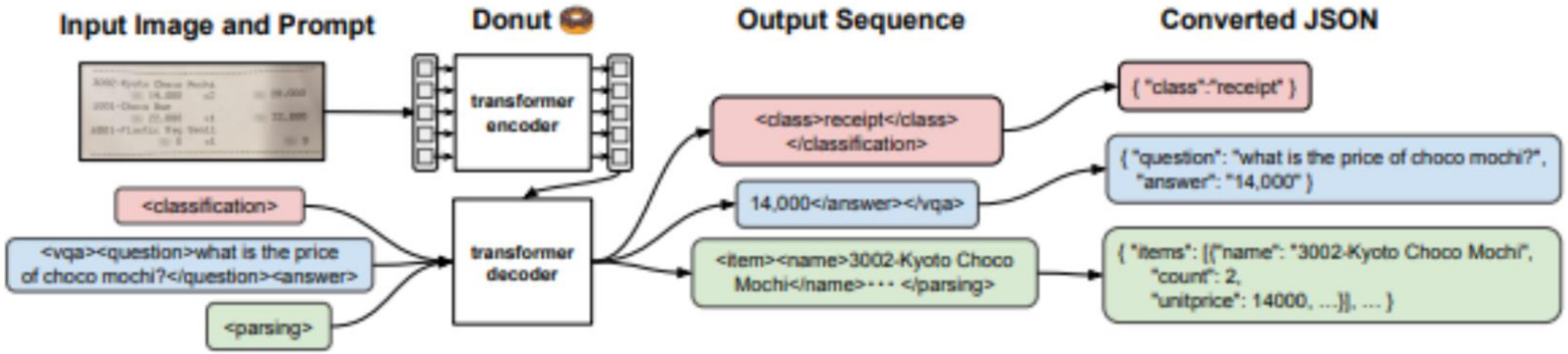


First Approach



Extract information

END2END DEEP-LEARNING MODEL



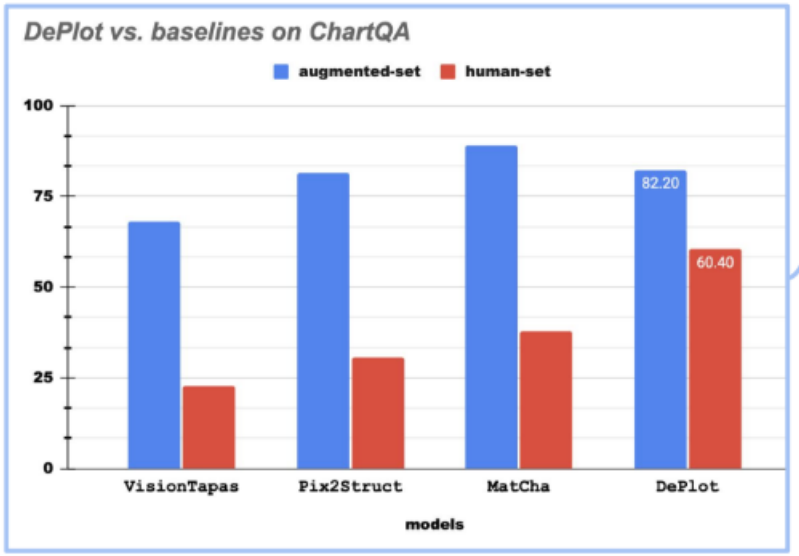
DONUT MODEL

Backbone: Encoder is swin transformer and Decoder is Bart

Pretrain decoder for 3 task: document classification, document parsing and VQA

In this task, we can finetune this if data adapt input model

END2END DEEP-LEARNING MODEL



DePlot

```
Header: models      |augmented-set|human-set
Row 1: VisionTapas  |67.2         |22.2
Row 2: Pix2Struct   |82.9         |30.4
Row 3: MatCha       |89.0         |38.0
Row 4: DePlot       |82.2         |60.4
```

LLM

Question: On average across the two sets, by how much does DePlot outperform the second strongest model?

Let's find the average of DePlot across the two sets. We add the two numbers in the DePlot row and divide by 2: 82.2+60.4=142.6/2=71.3. Let's find the second strongest model across the two sets. We find the row with the second highest average: Row 3. We find the average of MatCha across the two sets. We add the two numbers in the matplot row and divide by 2: 89.0+38.0=127.0/2=63.5. We subtract the second strongest average from the strongest average: 71.3-63.5=7.8. The answer is **7.8**.

DEPLOT MODEL