Graph Reverse Engineering

A detailed presentation on mechanism to extract text and Image labels from variety of charts

Name: Md Arif (112669645) & Aditya Kumar(112716642)
Supervisor: Klaus Mueller
Guide: Md Naimul Hoque

Index

Imp Link:
Git Repository: https://github.com/marif1901/GraphReverseEngineering
Google drive: Link

Note: In this presentation work that are to be pursued next semester are marked by **

Section 1

Project Proposal

Applying Graph Reverse Engineering to develop voice reader for visually impaired people

Retrieve data using reverse engineering methods and beautify graph for better visualization / develop voice reader for visually impaired people

Data collection:

- The state-of-the-art graph reverse-engineering networks are not comprehensive, i.e. they work on few popular charts
- We started off by collecting images of both conventional (bar, pie etc.) and un-conventional plots (parallel coordinates, bipartite plot etc.) for detecting purposes

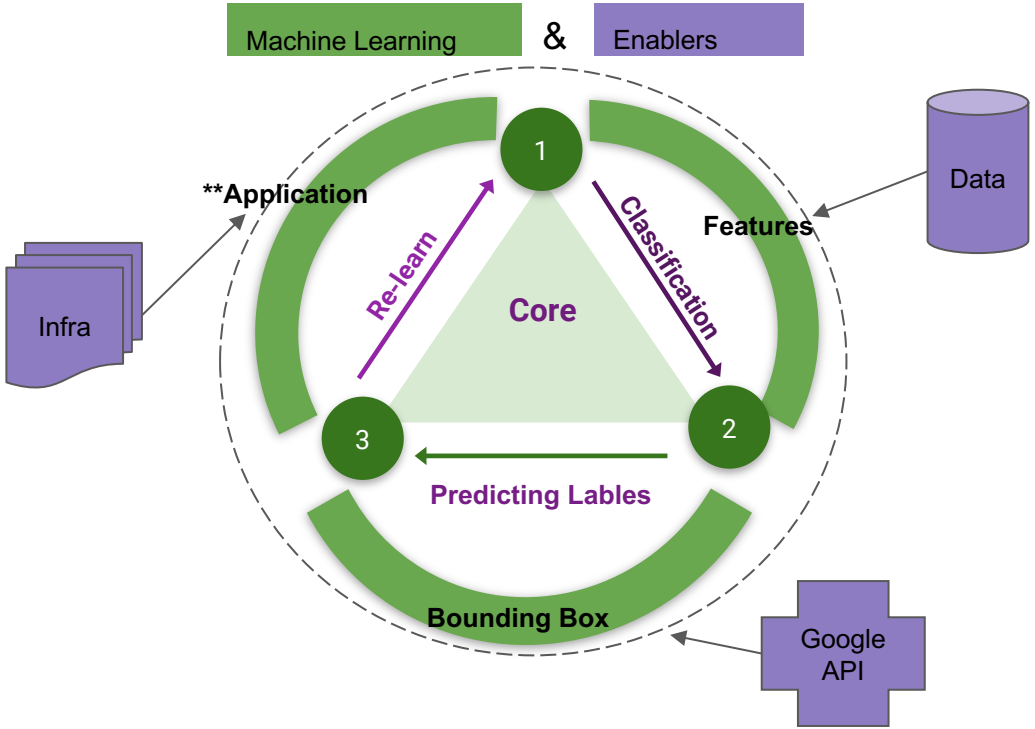
Extracting data from charts:

- Data-Ink Ratio from a graph. This is a helpful metric to determine the effectiveness of a visualization
- Bounding Box features to determine axis labels | axis title | legend label | title | subtitle

**Build Application:

- Different people have different perceptions and find different charts easy to grasp. Thus, showing end-users alternative visualization is a very interesting topic, and can have huge applications in areas such as computational journalism
- Develop voice reader for visually impaired people

Enabling Entities - How they impact Our Decision



Data: Data is the new currency. Bringing new insights from conventional charts (using ML) or using unexplored un-conventional charts is key in beating state of art.

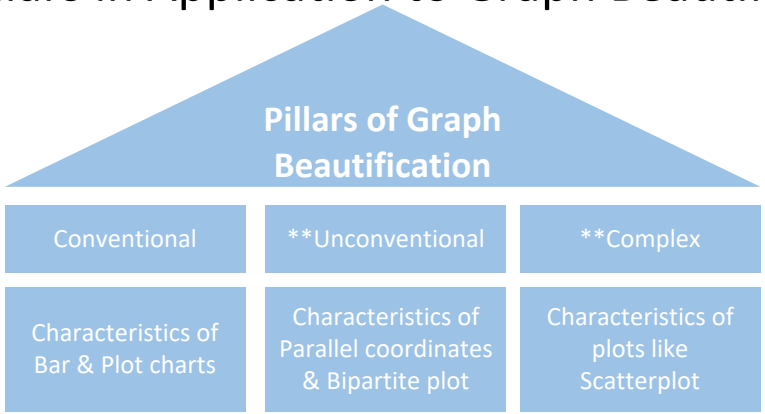
Tools: Tools help us make smarter and efficient decisions. Semi automated processes are good to start off. Google Cloud API is leveraged to extract bounding box features for any queries test image

Infra: Infra helps scale-up, where speed of decisions without errors is a must have. We have exposed an API for end user to upload Images that provides “goodness of a chart”

Section 2

Data Selection

Choice of data are pillars in Application to Graph Beautification



Training Data:

- Research Paper: We plan to utilize existing data resources from the research paper @**UW Interactive Data Lab** here
- Training Data We plan to use **Academic & Vega** data to get the combination of automatically generated chart features
- Collections → subject to charts variations

Test Data:

1. We have used 20% of data from **Academic repository** as test data to test the accuracy against which the true labels are known
2. For Un-seen end user we are using **Google Cloud API** to extract features and we are storing as a test Image and plan to fine tune the Model using the incoming Image

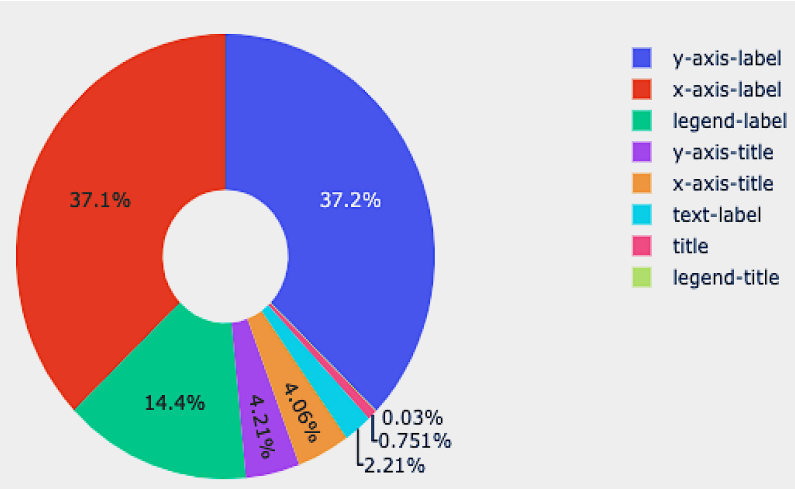
Section 3

Feature Engineering

Current stage >> Feature Enhancement stage by Sep'20

Current Stage

- normalized top-left coordinate
- normalized right-bottom coordinate
- normalized center coordinate
- normalized box size



Bounding Box	% Occurence in Training Images
y-axis-label	37.19%
x-axis-label	37.14%
legend-label	14.42%
y-axis-title	4.21%
x-axis-title	4.06%
text-label	2.21%
title	0.75%
legend-title	0.03%

**Sep '20

- angle from actual center
- radius from normalized center
- normalized top-left coordinate in container box
- normalized bottom-right coordinate in container box

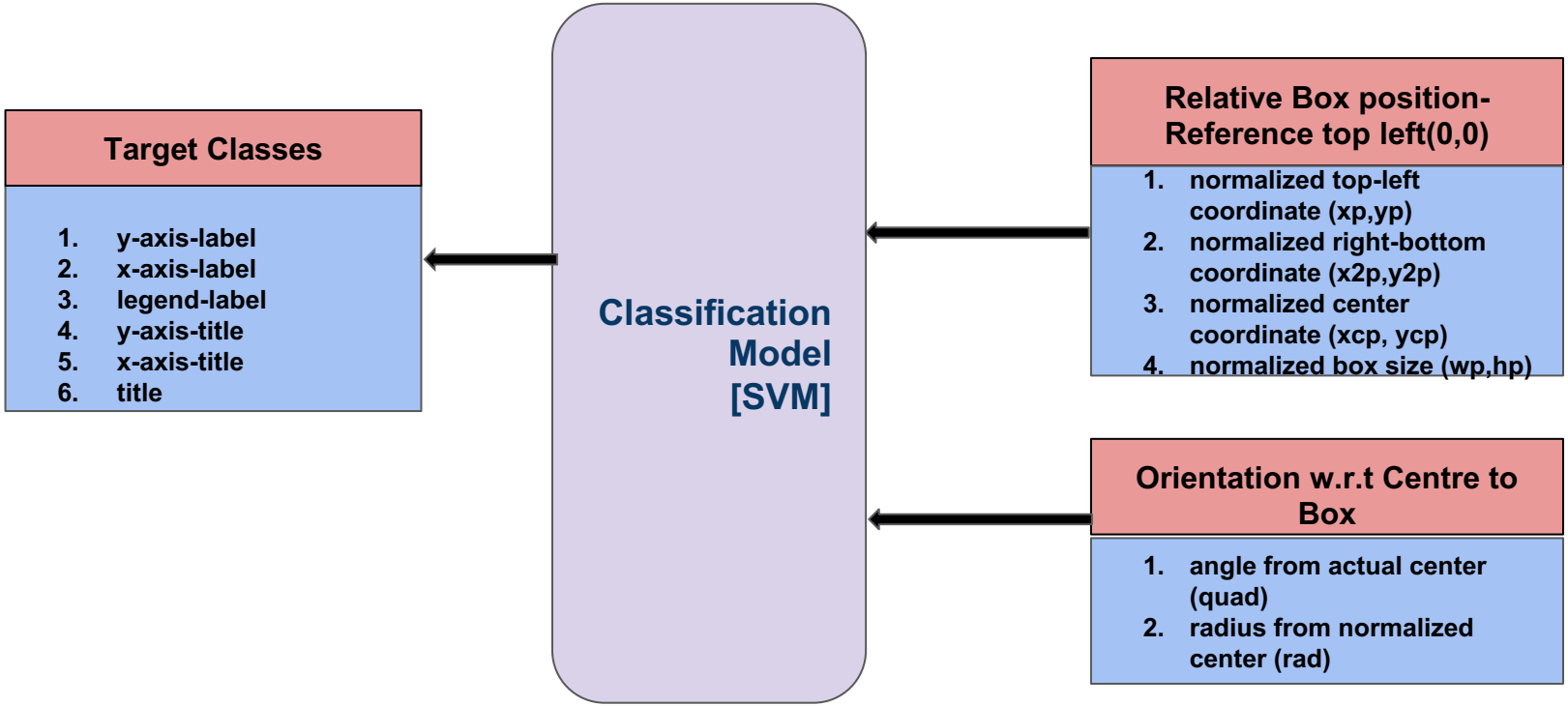
as they are very Has no dependency on external tools/applications. It is driven by Data Collection storage and trusted features extracted, we have not picked up legend-title & text-label as our class lable because very few images were found to contain

** Strategy requires many A/B testing to mature. Also, strategy could change due to changing chart complexity. Therefore, for evolving features, automating is very difficult

Section 4

Training Classifier

Model Features



Prediction Model

Model V0
Support Vector Machine (SVM)

Training & Validation Set

	precision	recall	f1-score	support
legend-label	0.90	0.91	0.91	184
title	0.90	0.90	0.90	10
x-axis-label	0.95	0.96	0.96	521
x-axis-title	0.86	0.83	0.84	52
y-axis-label	0.96	0.97	0.96	478
y-axis-title	0.96	0.88	0.92	57
accuracy			0.95	1302
macro avg	0.92	0.91	0.91	1302
weighted avg	0.95	0.95	0.95	1302

Training set score for SVM: 0.956722
Validation set score for SVM: 0.945469

Test Set

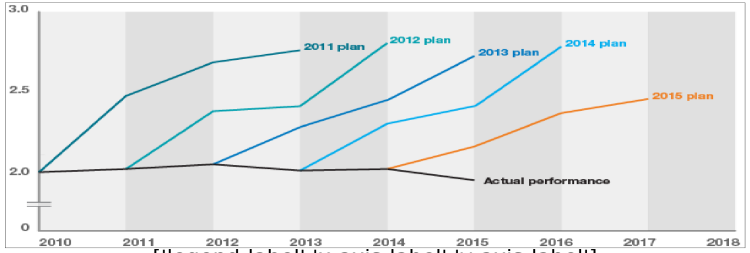
	precision	recall	f1-score	support
legend-label	0.92	0.92	0.92	202
title	1.00	0.69	0.82	13
x-axis-label	0.95	0.96	0.95	503
x-axis-title	0.84	0.82	0.83	50
y-axis-label	0.97	0.98	0.98	483
y-axis-title	0.98	0.90	0.94	51
accuracy			0.95	1302
macro avg	0.94	0.88	0.90	1302
weighted avg	0.95	0.95	0.95	1302

Test set score for SVM: 0.949309

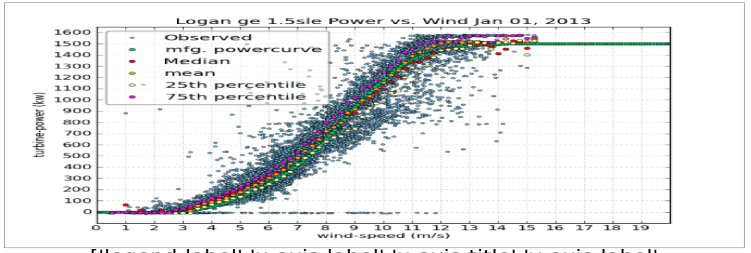
To remove high class imbalance we have under sampled so that we get high and low class in 1:1

Prediction Results on Some Random Test Images

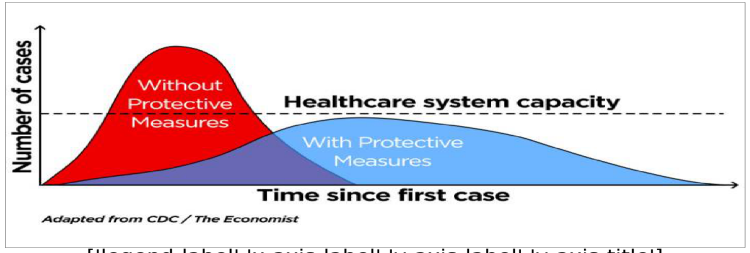
We have six target labels that could be predicted for any given test image



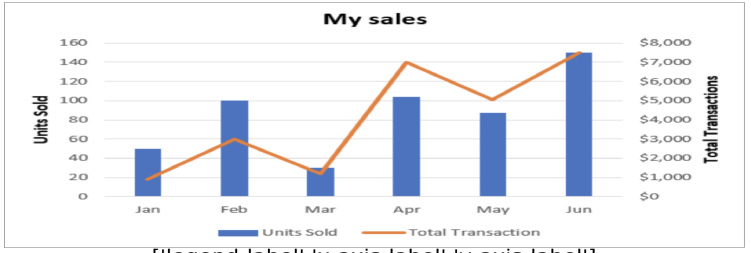
['legend-label' 'x-axis-label' 'y-axis-label']



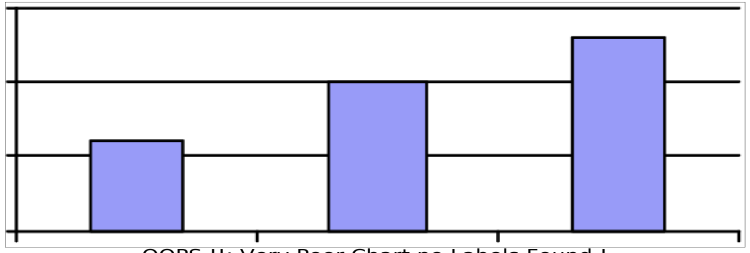
['legend-label' 'x-axis-label' 'x-axis-title' 'y-axis-label' 'y-axis-title']



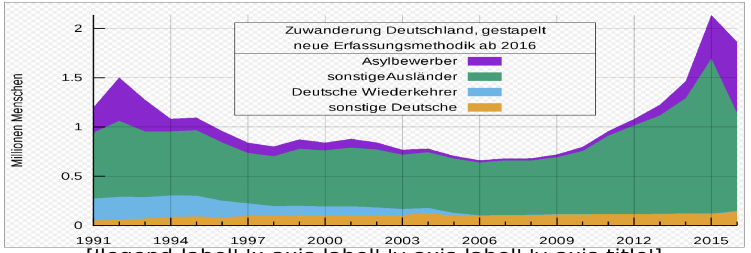
['legend-label' 'x-axis-label' 'y-axis-label' 'y-axis-title']



['legend-label' 'x-axis-label' 'y-axis-label']



OOPS !!: Very Poor Chart no Labels Found !



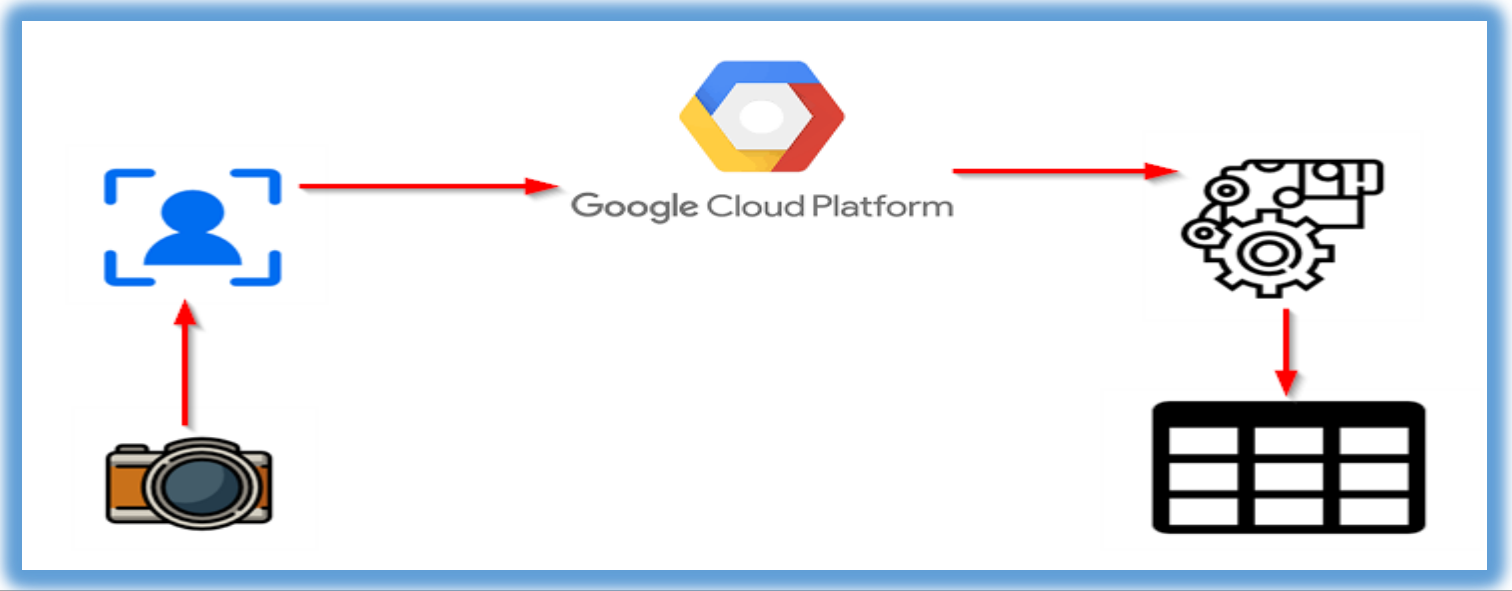
['legend-label' 'x-axis-label' 'y-axis-label' 'y-axis-title']

Section 5

Google Cloud API (Extracting real time bounding box features)

Why Google Cloud Vision & how it works?

- Sequential technique which works on the principle of ensemble. It combines a set of **weak learners** and delivers improved prediction accuracy.



Goodness of a Chart Explained: Features Extracted & Data Ink Ratio

- …
- …
-

Section 6

Exposing API to output attributes in real time from User Uploaded Image

Glimpse of the Production API

- …
- …
-

Next Steps

- Build robust model that could work even for complex charts like (like Scatterlpot, parallel co-ordinates, bipartite plots)
- Productionalize this model
  - Allow cross-functional services like “goodness of a chart”, voice reader from a graph, simple form of the complex graph
- Add more features (orientation specific, text speicifc)