Universität Passau
Fakultät für Informatik und Mathematik

CLASSIFICATION OF VISUALIZATIONS IN SCIENTIFIC LITERATURE

Masterarbeit zur Erlangung des akademischen Grades
Master of Science (M.Sc.)

am Lehrstuhl für ... oder an der Professur für ...
der Fakultät für Informatik und Mathematik
der Universität Passau

Name:	Arnold Azeem
Matrikelnummer:	79176
Fachbereich:	Informatik
Studiengang:	Master Informatik
Schwerpunkt:	TODO
Studienjahrgang:	TODO
Erstprüfer:	Prof. Dr. Christin Siefert
Zweitprüfer:	Prof. Dr. Michael Granitzer

Contents

List of Figures

List of Tables

Einleitung

Motivation

Figure 1.1: Describe this picture.

ABSTRACT

Distinct visualization techniques are used in scientific research publications to summarize large amount of data and also represent a variety of data. These visualizations help to communicate complex information and support the arguments presented in the paper in a easy to understand and follow way. These figures tend to reveal trends,patterns or relations that might otherwise be difficult to grasp using only text. In this context, classifying these visualizations is really relevant since there is a variety of visualizations and each one will have a different approach to processing it, example is extracting the raw data from it.

INTRODUCTION

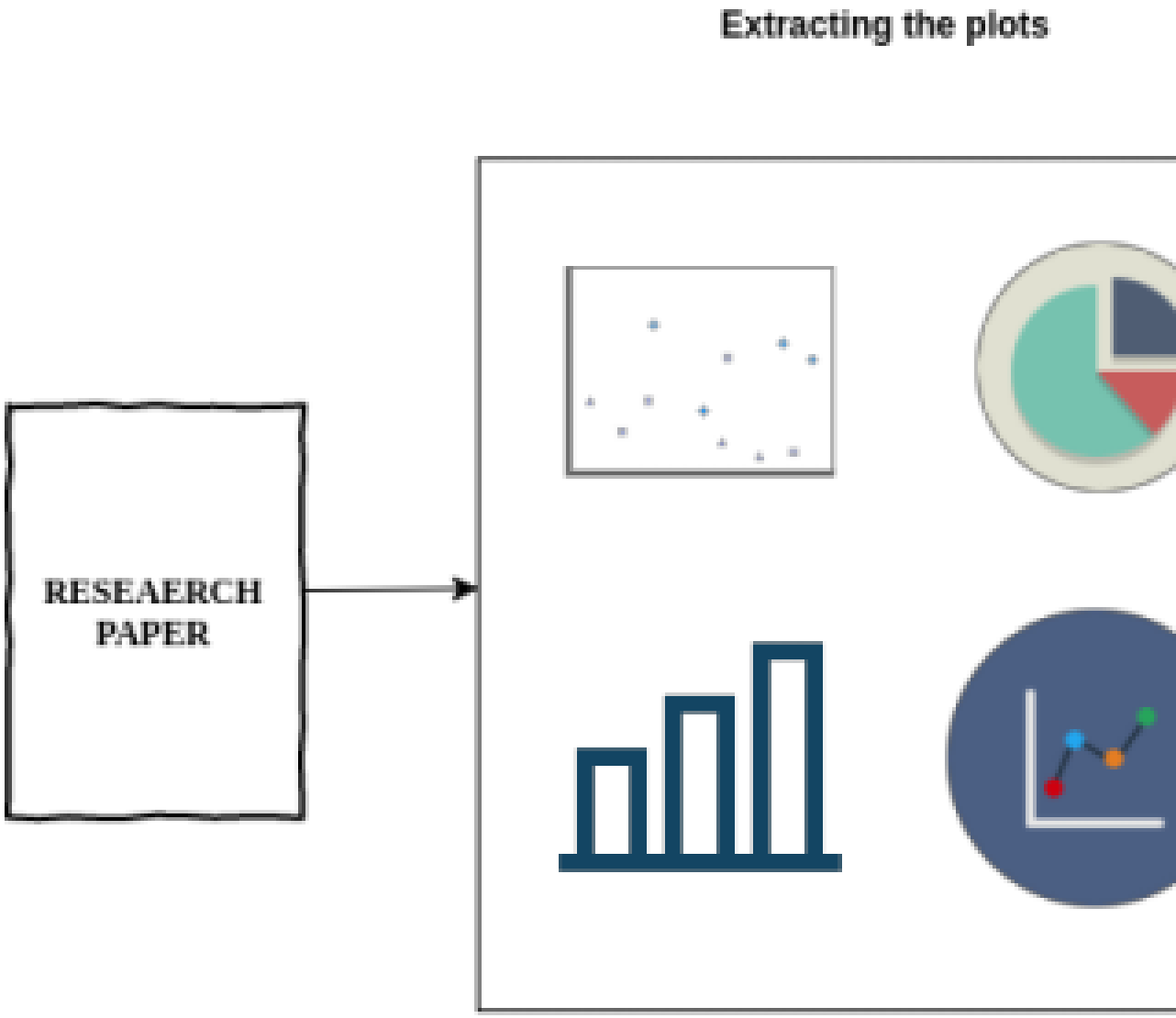
A picture tells a thousand words even though a cliché stands to be very true especially when it comes to presenting complex findings in scientific research publications. The importance of these figures in papers cannot be undermined since they provide a way to easily interpret,find patterns and relations in the data which would have otherwise been more complex relying on only textual data. All though extracting data from a plot manually is relatively easier, doing the same task automatically requires each type of plot to be processed specifically. In this work we present a way to classify each plot effectively, since that the first step before further processing of a plot is possible.

MOTIVATION

Complex data is better explained in scientific papers with the aid of visualizations. These plots present complex data in an easy to understand way compared to textual representation. The data which these visualizations contain when extracted play an important role in events where another researcher wants to verify the work of the publisher, this data can also be used to develop other visualizations in situations where the paper needs to be presented to a different audience with a different background as opposed to the audience which the visualizations were created for, Also when comparing two plots the raw data helps make a better decision than just the figures. Since each plot will be processed differently to extract the raw data, it very relevant that we can distinguish one plot from another and this is the main aim of this thesis.

OBJECTIVE

The purpose of this thesis is to answer the question **HOW WELL CAN WE CLASSIFY THE DIFFERENT TYPES OF PLOTS IN SCIENTIFIC LITERATURE.**. In this work we focus on only four plots. These plots are scatter plots,bar charts,line charts and Box plots. The diagram below shows the vision of this work, The first part of the diagram involves extracting or obtaining the four different types of plots mentioned earlier,after which we then label our plots and train a neural network model to be able to classify with high accuracy any of the four plots if shown to our model, then finally the raw data can be extracted from the detected plot. But this work mainly focuses on the red dotted lines shown below in the diagram which which is getting the plots, labeling them, training the model and classifying the plots.


Extracting the plots
RESEAERCH
PAPER

LITERATURE REVIEW

Our work is inspired by Architecture proposal for data extraction of chart images using Convolutional Neural Network(2017), in this work a simple LeNet-based CNN model trained with four classes. The model was implemented using Tensorflow, and the model architecture was made of 3 convolutional layers, followed by a fully connected layer. The training process is based on the dataset divided into mini-batches, random samples of fixed sizes(100) are selected and used as input for the CNN, as different batches are used as the input the model keeps learning to generalize the classes of the dataset and therefore becomes more robust.Each image is converted to JPG and resized to 224x224x3 (using crop or pad), that is, 224 pixels of height, 224 of width and 3 layers of output (one for each RGB color channel). Additionally, the process was executed in 1000 epochs and the learning rate was set to 0.003. The accuracy of the final model was 70%.

Dataset

Dataset for Matlab

The dummy data for creating the plots in Matlab were downloaded from Project Datasets [1] a site where datasets are provided in CSV formats for reasons of teaching people how to load datasets. The datasets are multidimensional and complied from different fields. On the average the datasets used contain about 500 instance and 5 different columns. The biggest dataset is called Speed dating data with over 8,000 observations of matches and non-matches, with answers to survey questions about how people rate themselves and how they rate others on several dimensions. The smallest dataset used is the Happiness dataset, it contains information about European quality of life survey with questions related to income, life satisfaction or perceived quality of society.

Dataset for R

For the plots in R, 13 random dummy csv files where download from an archive of datasets distributed with R [2] called Rdatasets. Rdatasets is a collection of 1147 datasets that were originally distributed alongside the statistical software environment R and some of its add-on packages, the main aim is to make these data more broadly accessible for teaching and statistical software development. On the average their are 80 instances and 5 columns. The biggest CSV file is the Australian athletes data set, these data were collected in a study of how data on various characteristics of the bloood varied with sport body size and sex of the athlete. It contains attributes like sex,height,weight and sports. The smallest dataset is the Canadian Women's Labour-Force Participation saved as the Bfox.csv file. The Bfox data frame has 30 rows and 7 columns, and contains Time-series data on Canadian women's labor-force participation, 1946–1975. It contains information like average wages of women, percent of adult women in the workforce etc.

Dataset for PYTHON

The dummy data used for creating the plots in Python were 15 randomly seleted csv files also from Rdatasets [2]. The biggest dataset among the 15 is the Monoclonal gammapothy data, contains natural history of 1341 sequential patients with monoclonal gammapothy of undetermined significance (MGUS). The CSV is with 1384 observations with 10 columns with attributes like age,sex,time of death and last contact in months. On the average each dataset contains about 200 instances and 7 columns of multi-dimensional data. The smallest dataset however contains only 33 instances with 11 columns and is called the Nuclear Power Station Construction Data.The data relate to the construction of 32 light water reactor (LWR) plants constructed in the U.S.A in the late 1960's and early 1970's. The data was collected with the aim of predicting the cost of construction of further LWR plants.

Dataset for Java

For the java plots I used the dataset made available by Plotly [3].They are CSV datasets used in the Plotly API examples.14 random CSV files were downloaded, the biggest file 3d-line-plot has 1002 instances and 9 columns, and on the average each file contains about 100 instances and 9 columns. The smallest file however is made of 33 instances and 12 columns called the mtcars file. It contain information about a variety of different car models like the number of gears, speed etc.

DUMMY DATA			
PYTHON	MATLAB	R LANGUAGE	JAVA
3d_line_sample_data.csv LightFordwardFlapStall.csv line_3d_dataset.csv longley.csv loti.csv lung.csv nuclear.csv timeseries.csv USJudgeRatings WVSCulturalMap.csv wind_rose.csv volcano.csv uspop2.csvm	Camera.csv Cars.csv CausesOfDeath- France.csv Cereal.csv film.csv happiness.csv TestData1.csv TestData2.csv mpg.csv okcupid- compatibility- religion.csv spectral.csv stockdata.csv subplots.csv	ais.csv Angell.csv Baumann.csv Bfox.csv cane.csv carprice.csv Chirot.csv Davis.csv Ericksen.csv Florida.csv Highway1.csv Pottery.csv Prestige.csv salin- ity.csv urine.csv	3d-line-plot.csv 3d-scatter.csv 2011_flight_paths.csv 2011_us_exports.csv auto-mpg.csv candlestick_dataset.csv finance-charts-apple.csv globe_contours.csv hobbs-pearson- trials.csv motor_trend_tests.csv nz_weather.csv volcano.csv iris.csv mtcars.csv

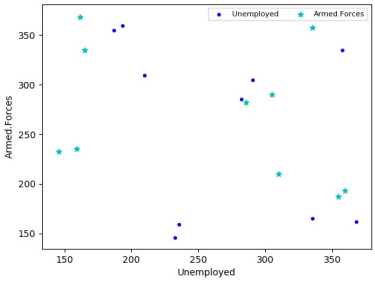
Table 6.1: Names of CSV files used in each language

CREATING PLOTS

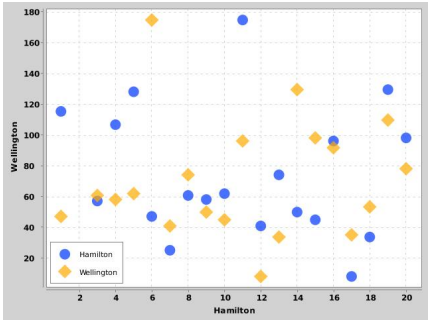
Scripts in various languages were written to handle the plotting and labeling process automatically. All datasets for a particular plot(example scatter plot for python) are put into one folder. The scripts reads each CSV file column by column. The first column is considered as the x-axis and its header the x-label and the next column will be the y-axis and its header the y-label, The y-axis then becomes the x-axis for the next plot and this process continues till all columns are used, while doing this all non numeric columns are ignored. Next the libraries used in creating the plots in various languages. For reading the values, the scripts written are done in a way to automatically create the plots. The inspiration for creating a variety of plots to capture all type of plots used in scientific papers was gotten by inspecting the dataset of this [4] and [5]. The tables below describe how the plots where created in each language, the plotting libraries used, the variants of a particular plot come under the type column, the parameter column describes parameters that were changed and finally the number of plots created were also added in the last column.

SCATTER PLOTS				
Language	Library	Parameters	Types	Number of plots
Python	Matplotlib v2.1.2 Plotly v2.5.1 Seaborn v0.8.1	MarkerStyle ['o', '*', '.', '+','x']	Normal scatter	100
MATlAB	Default	MarkerStyle ['o', '*', '.', '+','x','s']		644
R LANGUAGE	Default,Plotly R Library ggplot2	MarkerStyle ['o', '*', '.', '+','x','s']		644
JAVA	XChart 3.5.1,	setMarkerSize (16) LegendPosition		644

Table 7.1: How Scatter plots were created

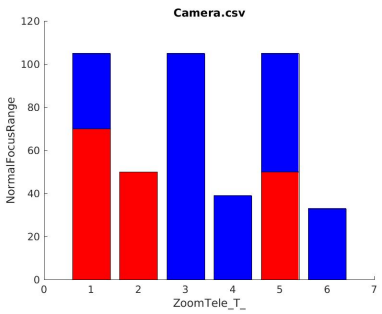


(a) created with Matplotlib

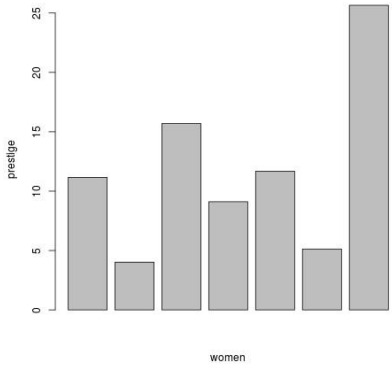


(b) created with Jfreechart library

Figure 7.1: Example Scatter plots

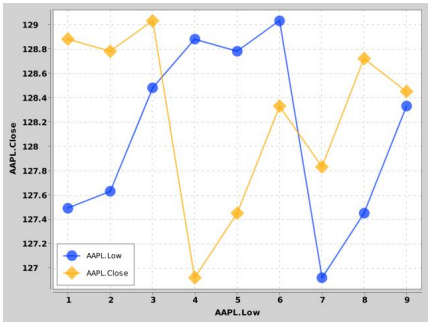


(a) created in Matlab

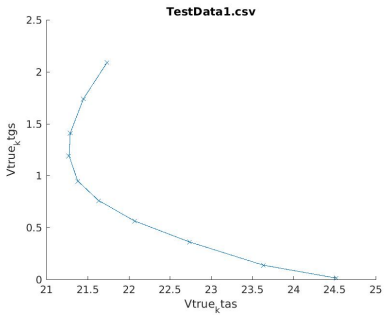


(b) created in R

Figure 7.2: Example Bar Charts



(a) created in Java

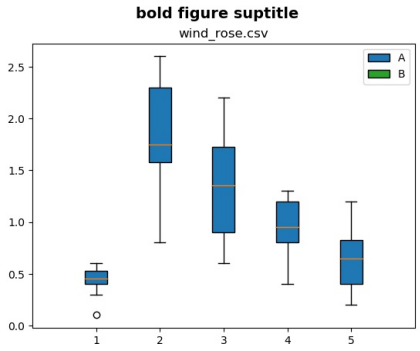


(b) created in Matlab

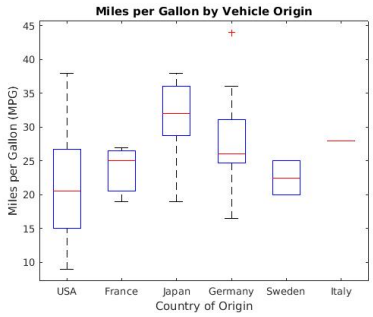
Figure 7.3: Example Line Charts

BAR CHARTS				
Language	Library	Parameters	Types	Number of plots
Python	Matplotlib v2.1.2 Plotly v2.5.1 Seaborn v0.8.1		Horizontal and Vertical Stacked bar charts Grouped bar charts	100
MATlAB	Default	Width of bar		644
R LANGUAGE	Default,Plotly R Library ggplot2			644
JAVA	XChart 3.5.1,	setMarkerSize (16) LegendPosition		644

Table 7.2: How Bar Charts were created



(a) created in python



(b) created in Matlab

Figure 7.4: Example Box Plots

Scatter plot

For constructing the plots for the scatter plots I looked through a dataset from this paper[]. The dataset contained 555 scatter plots and looking through them I realized the markers used to plot where mainly zeros, triangles,squares,dots and x's. So instead of just plotting all the markers randomly i plotted those mentioned above with a higher probability than the rest of the markers. Also the scatter plots consisted of positive and negative correlation and just a few zero correlation. I also noticed that some markers were filled and other were not filled. So for creating my dataset I plotted with some markers being filled and other unfilled and this was done randomly.

LINE CHARTS				
Language	Library	Parameters	Types	Number of plots
Python	Matplotlib v2.1.2 Plotly v2.5.1 Seaborn v0.8.1		Horizontal and Vertical Normal Line Lines with markers multiple lines	100
MATlAB	Default	Width of bar		644
R LANGUAGE	Default,Plotly R Library ggplot2			644
JAVA	XChart 3.5.1,			644

Table 7.3: How Line Charts were created

BOX PLOTS				
Language	Library	Parameters	Types	Number of plots
Python	Matplotlib v2.1.2 Plotly v2.5.1 Seaborn v0.8.1		Horizontal and Vertical Lines with markers multiple box plots	1000
MATlAB	Default			644
R LANGUAGE	Default,Plotly R Library ggplot2			644
JAVA	XChart 3.5.1,			644

Table 7.4: How Box Plots were created

Box Plots

Bibliography

[1] Wikipedia contributors. Latex — Wikipedia, the free encyclopedia. https://perso.telecom-paristech.fr/eagan/class/igr204/datasets, 2011. [Online; accessed 17-April-2018].

[2] Franklin Allen and Risto Karjalainen. Using genetic algorithms to find technical trading rules. *Journal of Financial Economics*, 51:245–271, 1999.

[3] Wikipedia contributors. Latex — Wikipedia, the free encyclopedia. https://github.com/plotly/datasets, 2011. [Online; accessed 17-April-2018].

[4] Paulo Roberto Silva Chagas Junior, Alexandre Abreu De Freitas, Rafael Daisuke Akiyama, Brunelli Pinto Miranda, Tiago Davi Oliveira De Araújo, Carlos Gustavo Resque Dos Santos, Bianchi Serique Meiguins, and Jefferson Magalhães De Morais. Architecture proposal for data extraction of chart images using convolutional neural network. In *Information Visualisation (IV), 2017 21st International Conference*, pages 318–323. IEEE, 2017.

[5] Po-shen Lee, Jevin D West, and Bill Howe. Viziometrics: Analyzing visual information in the scientific literature. *IEEE Transactions on Big Data*, 4(1):117–129, 2018.