IIT JODHPUR

M.TECH. PROJECT

Chart2Table: A Neural Model for Converting Charts to a Latex Script of Table

Author:
Neelu Verma
Roll Number: MP19AI002

Supervisor:
Anand Mishra, PhD

A project submitted in partial fulfillment of the requirements for the degree of M.Tech-PhD in AI

in the

Department of Computer Science and Engineering

February 2, 2022

Declaration of Authorship

I, Neelu Verma
Roll Number: MP19AI002, declare that this thesis titled, "Chart2Table: A Neural Model for Converting Charts to a Latex Script of Table" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

IIT JODHPUR

Abstract

Department of Computer Science and Engineering

M.Tech-PhD in AI

Chart2Table: A Neural Model for Converting Charts to a Latex Script of Table

by Neelu Verma
Roll Number: MP19AI002

Scientific documents, business reports, magazines and newspapers often contain plots such as bar, pie and line charts. These plots provide many useful information about the content in a precise form. However, despite progress in developing robust OCR engines, these plots are only labelled as a graphic region, and ignored for further analysis. In this work, our goal is to develop methods to make chart images machine readable. We refer to this novel problem as *chart2table*. The goal of *chart2table* is as follows: given a chart image, we would like to automatically convert it to a table (or equivalently corresponding LaTex script) which encodes all the information present in the chart. As part of this MTP, we perform extensive literature survey and obtain preliminary results using image processing-based and seq2seq learning-based techniques. Furthermore, we also worked towards obtaining a fully-annotated dataset, namely ChartReal for evaluation of this task. In the future, our goal is to develop neural methods building on the top of recent advancements in vision and language techniques for this task, and perform rigorous experiments and benchmarking.

Acknowledgements

I would like to express my special thanks of gratitude to my guide, Dr. Anand Mishra for his valuable guidance and providing me proper resources to do this wonderful research project on the topic "Chart2Table: A Neural Model for Converting Charts to a Latex Script of Table". Secondly I would like to thanks my friends Arpit Gupta and Raghav Ranjan who help to motivate me and help me in implementation of this research project.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The optical character recognition (OCR) is one of the most successful applications of machine learning. Most OCR engines [Smith and Inc, 2007; Tafti et al., 2016] read the textual content with a reasonably high performance in document images, but they tend to ignore the graphics contents by just tagging them. In scientific documents, these graphic contents often contain information-rich charts such as bar, pie and line charts. Our goal is to fill this gap in modern OCR engines by developing a method to make charts more interpretable. To this end, we propose a new task in document image literature, namely *chart2table*. The goal of this task as illustrated in the Figure 1.1 is to convert unstructured chart images to semi-structured and machine-readable tables (or their equivalent Latex scripts).

1.2 Challenges

The *chart2table* is pathetically challenging task due to inherent complex structures of charts and diversity among them. Consider an example shown in Figure 1.1. It is non-trivial to precisely convert a bar chart shown in the left to a table which encodes all the information present in the chart. The heuristic or rule-based methods, e.g., based on image processing techniques may work well for a category of chart image, but does not generalize well to a novel chart with an entirely different visualization, e.g., a chart without legend and having horizontal bars may require a different rule. We see an analogy between *chart2table* and image captioning problem [Zhu et al., 2020; Rampal and Mohanty, 2020]. In image captioning problem, given a natural scene image the goal is to generate a natural language description. Whereas in *chart2table*, given a chart image, our aim is to generate LaTex script corresponding to a table which encodes all the information present in the chart image. This analogy motivates us to explore and adapt state-of-the-art image captioning techniques to our problem.

1.3 Proposed Solution

Sequence to sequence models have shown promising performance in image captioning tasks. We adapt them for *chart2table* and propose the following solution where model takes input as sequence of bounding boxes and text labels and produce latex script of table.

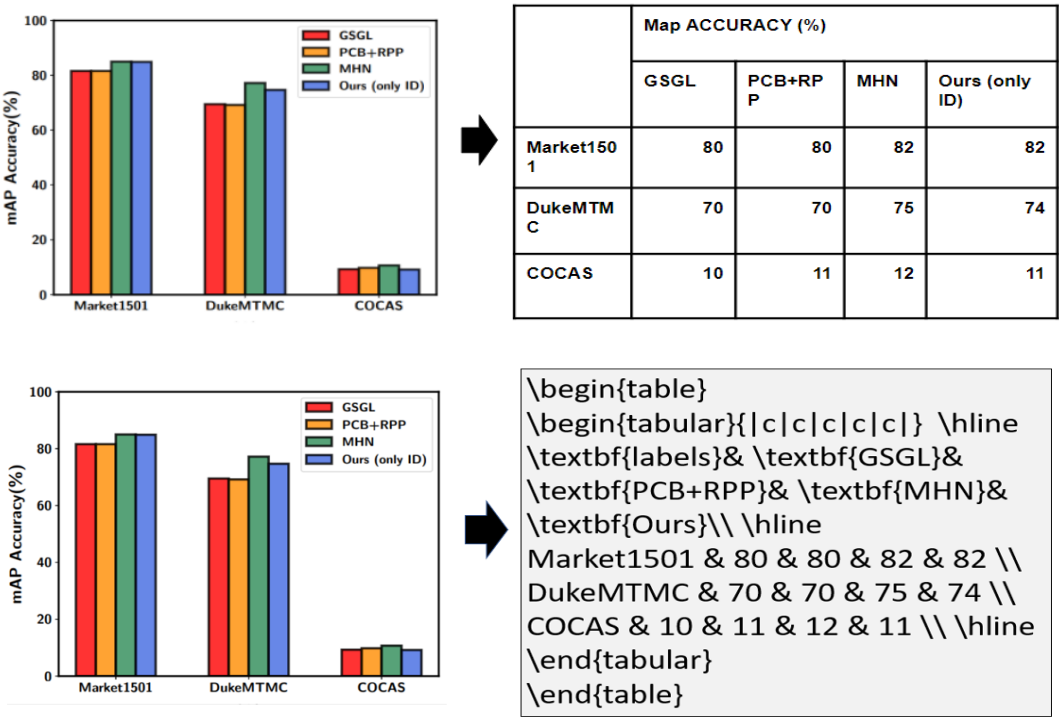


FIGURE 1.1: **Chart2Table**: The overall objective of our research to show input image is converting into latex script of table. Left side represent the image from ChartReal dataset and right side show the corresponding latex script

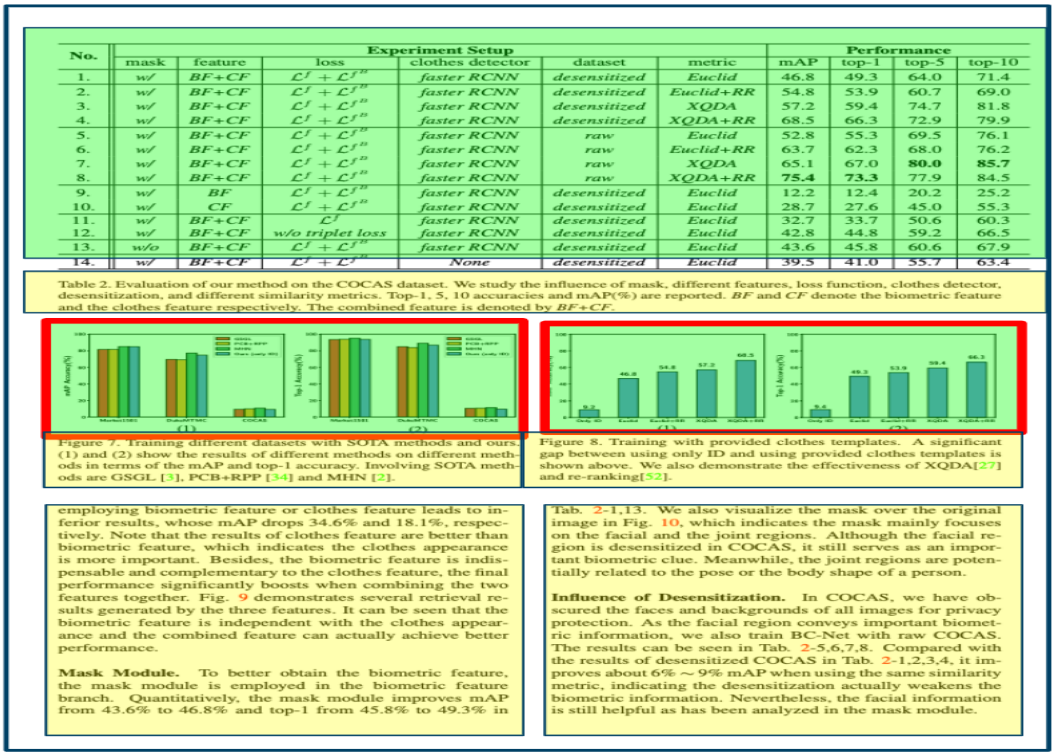
1.4 Our Contributions

Contributions of this work are three folds,

1. We draw attention of document image analysis community to the important but unexplored problem of interpreting chart images and making them better machine-readable by converting them to semi-structured table encoding all the information in the chart image. We refer this new task as *chart2table*.

2. We introduce *ChartReal*: a first fully annotated dataset for evaluating performance of *chart2table*. This dataset contains chart images collected from various such as technical reports, research papers, magazines and newspapers. Further, each of these chart image is annotated by a human annotator to get corresponding table.

3. We perform a systematic study of the problem and provide strong baseline methods. These proposed baseline methods are built on modern image captioning based techniques. Further, we evaluate the performance of the baseline methods in a principled way. We firmly believe the new task and associated dataset will enhance the interest in the area of understanding chart images.



No.	Experiment Setup						Performance			
	mask	feature	loss	clothes detector	dataset	metric	mAP	top-1	top-5	top-10
1.	*w/*	*BF+CF*	$\mathcal{L}^I + \mathcal{L}^{T^m}$	*faster RCNN*	*desensitized*	*Euclid*	46.8	49.3	64.0	71.4
2.	*w/*	*BF+CF*	$\mathcal{L}^I + \mathcal{L}^{T^m}$	*faster RCNN*	*desensitized*	*Euclid+RR*	54.8	53.9	60.7	69.0
3.	*w/*	*BF+CF*	$\mathcal{L}^I + \mathcal{L}^{T^m}$	*faster RCNN*	*desensitized*	*XQDA*	57.2	59.4	74.7	81.8
4.	*w/*	*BF+CF*	$\mathcal{L}^I + \mathcal{L}^{T^m}$	*faster RCNN*	*desensitized*	*XQDA+RR*	68.5	66.3	72.9	79.9
5.	*w/*	*BF+CF*	$\mathcal{L}^I + \mathcal{L}^{T^m}$	*faster RCNN*	*raw*	*Euclid*	52.8	55.3	69.5	76.1
6.	*w/*	*BF+CF*	$\mathcal{L}^I + \mathcal{L}^{T^m}$	*faster RCNN*	*raw*	*Euclid+RR*	63.7	62.3	68.0	76.2
7.	*w/*	*BF+CF*	$\mathcal{L}^I + \mathcal{L}^{T^m}$	*faster RCNN*	*raw*	*XQDA*	65.1	67.0	**80.0**	**85.7**
8.	*w/*	*BF+CF*	$\mathcal{L}^I + \mathcal{L}^{T^m}$	*faster RCNN*	*raw*	*XQDA+RR*	**75.4**	**73.3**	77.9	84.5
9.	*w/*	*BF*	$\mathcal{L}^I + \mathcal{L}^{T^m}$	*faster RCNN*	*desensitized*	*Euclid*	12.2	12.4	20.2	25.2
10.	*w/*	*CF*	$\mathcal{L}^I + \mathcal{L}^{T^m}$	*faster RCNN*	*desensitized*	*Euclid*	28.7	27.6	45.0	55.3
11.	*w/*	*BF+CF*	$\mathcal{L}^I$	*faster RCNN*	*desensitized*	*Euclid*	32.7	33.7	50.6	60.3
12.	*w/*	*BF+CF*	*w/o triplet loss*	*faster RCNN*	*desensitized*	*Euclid*	42.8	44.8	59.2	66.5
13.	*w/o*	*BF+CF*	$\mathcal{L}^I + \mathcal{L}^{T^m}$	*faster RCNN*	*desensitized*	*Euclid*	43.6	45.8	60.6	67.9
14.	*w/*	*BF+CF*	$\mathcal{L}^I + \mathcal{L}^{T^m}$	*None*	*desensitized*	*Euclid*	39.5	41.0	55.7	63.4

Table 2. Evaluation of our method on the COCAS dataset. We study the influence of mask, different features, loss function, clothes detector, desensitization, and different similarity metrics. Top-1, 5, 10 accuracies and mAP(%) are reported. *BF* and *CF* denote the biometric feature and the clothes feature respectively. The combined feature is denoted by *BF+CF*.

Figure 7. Training different datasets with SOTA methods and ours. (1) and (2) show the results of different methods on different methods in terms of the mAP and top-1 accuracy. Involving SOTA methods are GSGL [3], PCB+RPP [34] and MHN [2].

Figure 8. Training with provided clothes templates. A significant gap between using only ID and using provided clothes templates is shown above. We also demonstrate the effectiveness of XQDA[27] and re-ranking[52].

employing biometric feature or clothes feature leads to inferior results, whose mAP drops 34.6% and 18.1%, respectively. Note that the results of clothes feature are better than biometric feature, which indicates the clothes appearance is more important. Besides, the biometric feature is indispensable and complementary to the clothes feature, the final performance significantly boosts when combining the two features together. Fig. 9 demonstrates several retrieval results generated by the three features. It can be seen that the biometric feature is independent with the clothes appearance and the combined feature can actually achieve better performance.

Mask Module. To better obtain the biometric feature, the mask module is employed in the biometric feature branch. Quantitatively, the mask module improves mAP from 43.6% to 46.8% and top-1 from 45.8% to 49.3% in Tab. 2-1,13. We also visualize the mask over the original image in Fig. 10, which indicates the mask mainly focuses on the facial and the joint regions. Although the facial region is desensitized in COCAS, it still serves as an important biometric clue. Meanwhile, the joint regions are potentially related to the pose or the body shape of a person.

Influence of Desensitization. In COCAS, we have obscured the faces and backgrounds of all images for privacy protection. As the facial region conveys important biometric information, we also train BC-Net with raw COCAS. The results can be seen in Tab. 2-5,6,7,8. Compared with the results of desensitized COCAS in Tab. 2-1,2,3,4, it improves about 6% ∼ 9% mAP when using the same similarity metric, indicating the desensitization actually weakens the biometric information. Nevertheless, the facial information is still helpful as has been analyzed in the mask module.

FIGURE 1.2: Modern OCR engines detect text and graphic regions with a reasonably good performance. The plots in the images (shown in red box) are often labelled as graphics and ignored for the further analysis. Our goal is to make these plots machine readable.

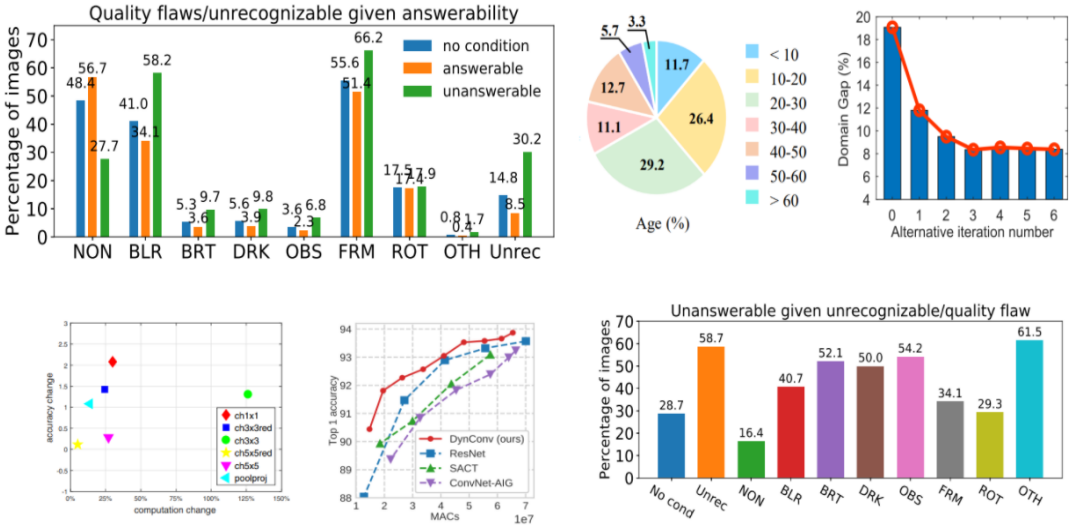


FIGURE 1.3: Our proposed dataset image gallery which shows our dataset contains more realistic, complex and variety of charts

Chapter 2

Literature Review

2.1 Literature Review

Any statistical charts contain complex, extensive information and a set of structural text and data. Charts/figures are impressive, useful, and extensively used representations of data in any documents. It can contain various text information like title, axis-title, legend, axis-tick, and data encoded into bar's width or height. For this purpose, one must first detect the bounding boxes of texts and bars. The system that automatically extracts information from charts would greatly benefit knowledge management because no one wants to look at formidable data. Such knowledge can be combined with other data sets to enhance business value further.

It is progressively hard to find robust technologies that can understand and predict the documents and charts and provide text labels. Large images synthetic dataset is publicly available, but it is hard to find charts with annotations in an appropriate form. The reason can be extraction of charts can face issues of legal and copyright. Another reason might be missing relevant annotations on scrapping the charts from the document. It can lose the necessary information. In the direction of converting extracted data into a semi-structured table, [Pasupat and Liang, 2015] present a compositional semantic parsing approach for the ease of questions answering task.

[Liu, Klabjan, and NBless, 2019] proposed a single deep neural classification framework that extracts this textual and graphical information from bar and pie charts with the accuracy of 79.4 and 88.0% resp. While for other charts performance is very low. [Savva et al., 2011] proposed a Revision method to redesign charts for classification, visualization, and data extraction. This approach extracted low-level image features using pixel shape and color information for image classification using SVMs. Bar charts text can be axis-tick, axis-title, legends, title, and chart data are available in bar's height and width. Various object detection methods like RCNN, Fast-RCNN, Faster-RCNN, YOLO, SSD, and some specially designed ways have been proposed for chart data detection and extraction in the form of bounding box information.

[Cliche et al., 2017] presented an automatic data extraction from the scatter plot with 89% test accuracy. FigureQA is a synthetic chart dataset proposed by [Kahou et al., 2017] of five class dataset. Most of these proposed datasets either have a question answering pair annotations or have no annotations. No corpus is available for the chart to table and chart to latex pair annotations. When we see out of question answering task, we found that image to latex creation is an important task to solve as a sequence to sequence problem. Long sort term memory LSTM [Hochreiter and Schmidhuber, 1997] a variant of recurrent neural network (RNN) [Sherstinsky, 2018] is a widely used model to sequence to sequence problem for machine translation.

There are lot of work based on Classification, Data Extraction and Question Answering related to charts. But no paper is reported to create table structure from charts images. Nice work in the field of the image to LaTex field is proposed by [Deng, Kanervisto, and Rush, 2016] to decompile an image to latex markup and web pages to HTML snippets using attention based enocder-decoder model. This paper introduces a real-world dataset of mathematical expressions with LaTex markup pair. He also proposed a synthetic dataset that includes web pages and corresponding HTML snippets with higher performance with rendered images.

2.2 Challenges form Literature Review

1. In [Kafle et al., 2018], the DVQA dataset promoted the dynamic question encoding. However, this scheme shows more complexity, including pie-charts, plots, and other visualizations since the dataset only contains the bar charts.

2. Although chart question answering pipeline modeled in [Kim and Agrawala, 2020], achieved reasonable accuracy, it failed to detect synonyms for visual features.

3. For single deep neural network developed in [Liu, Klabjan, and NBless, 2019], augmenting the training data set employing a more comprehensive simulation was an easy way to improve the performance consistently. However, it is more challenging to find a way to cope with small objects.

4. The method developed in [Wang et al., 2018], enhanced the performance of recalling of table selection model. But providing an accurate decision on choice selection is still a tedious process.

5. In [Reddy et al., 2019], deep learning model obtained significant improvements in training time compared to relation networks. Moreover, on real-life scientific figures, QA tasks were not easy to accomplish.

2.3 Advantages and Disadvantages

Advantages and disadvantages from existing papers			
Authors	Methods	Advantages	Disadvantages
[Deng, Kanervisto, and Rush, 2016]	visual attention-based model to decompile an image into presentational markup	Novel work in the direction of Image to Markup (Latex Script) generation.	work for image of formulas and fails for chart images. Also dataset is restricted to images of formulas only, not other figure like charts.
[Wang et al., 2018]	Semi-structured tabular data	Achieved better performance due to the effect of attention layer and efficient model.	Failed to add MCQs with multiple correct choices to the dataset.
[Liu, Klabjan, and NBless, 2019]	Single deep neural network	It achieved successful results for the simulated bar charts	Achieved poor performance on the images downloaded from internet.
[Kim and Agrawala, 2020]	Automatic chart question answering pipeline	Achieved more transparency than the answers generated by the humans.	Failed to obtain better fluency and also offered little variations.
[Kafle et al., 2020]	Parallel recurrent fusion of image and language (PReFIL)	Provided better potential to improve retrieval of information from charts.	Failed to provide better performance using dynamic encoding method.

TABLE 2.1

Chapter 3

The ChartReal Dataset (Proposed Dataset)

3.1 The ChartReal Dataset

Our newly created dataset is more realistic than existing datasets. We captured images from a recently published paper in various fields and annotate that images. This dataset contains much more realistic and complex images, and some are hard to annotate. This dataset includes almost all types of pictures from papers as they are not in the same format.

3.1.1 Existing Datasets

Some of the existing synthetic datasets like FigureQA and DVQA are more inclined toward reasoning over charts that do not have variable data labels. They do not have real-valued data and not have annotated complex reasoning questions. PlotQA is a more realistic dataset than FigureQA and DVQA, but the problem with PlotQA is followed the same pattern of images. There is a lack of variability of labels in this dataset. Its annotation is also inclined to reasoning over charts. DocFigure [Jobin, Mondal, and Jawahar, 2019] is a dataset that contains real word document images of 28 categories mixed dataset and quality of these images are useful. Still, this dataset also does not have annotations figures and adequate to classify images.

3.2 Dataset Annotations

As we are working on sequence to sequence machine translation, we require annotated table format of these images to generate the latex pair as ground truth. No dataset contains chart annotations in table format. To provide table annotation of chart images, We have created and tested our model on these datasets.

- First and foremost, we captured and collected the images of 2k from the various standard research papers publicly available online. We annotated these charts into tables. In the future, we are targeting to make this dataset very large for further research.
- Secondly, we created a synthetic dataset of 25k images to train our model. 2000 images are generated randomly for testing purposes. For creating synthetic images, we used python script using matplotlib plotting libraries. This randomizes various variables like font, size, color, legends, x-tick, y-tick, number of images.

- we annotated the 40k images of bar chart out of 1 million datasets of PlotQA in table and latex pair out of which 40k images are bar chart images and ready to use as we are mainly focusing on the bar chart in this paper.

Chapter 4

Methodology

4.1 Methodology

4.1.1 Problem Statement

The primary intention of this research will be to provide a machine translation system for converting input string of bounding boxes and text labels from bar chart images to the sequence of LaTax form.

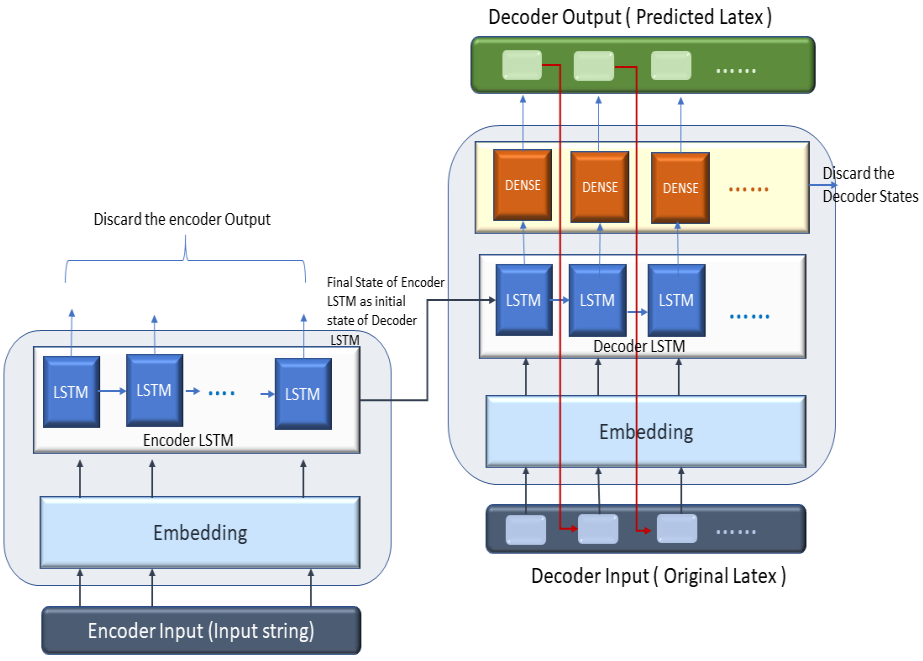


FIGURE 4.1: Methodology: block diagram of our proposed architecture which is encoder decoder base model. Left block showing encoder working and right block showing decoder working

4.1.2 Feature Extraction

The gray scaling and thresholding image segmentation method in digital image processing provides values between 0 (black) and 255 (white) to detect the bar's text and position. After binaries this value to black and white, tesseract is used to extract the text from the image, along with their position, and appended into a single string, two spaces separate each reading. The image is binarized and then inverted, following which the opening (erosion followed by dilation) is performed on the image to

eliminate all the noise and textual data, leaving us with only the chart's bars.
The bar charts' images need to traverse along its axis to detect and calculate the height and position (w.r.t. to the leftmost corner) of the bar. Step's readings are converts to a string and again append at the back of the string obtained from tesseract. Four spaces separate the strings obtained in the two different steps.
This step gives us a single string that contains the features of the graph and serves as an input to the sequence to sequence model in the next step.

4.1.3 Data Preprocessing

The string that we obtain in the previous step is first passed through data cleaning and prepossessing, where all the unwanted special characters, starting and ending white spaces were removed. The string obtained after extracting the features from the graph image is passed on to the model as input after it is prepossessed. The model works on this string and predicts the output LaTax table script.

4.2 Models

4.2.1 Baseline 1: Pre-trained VGG16

We used the pre-trained VGG16 model by replacing the top layer with a task-specific flatten layer as our first baseline model over 4000 synthetic training images and 1000 randomly generated testing images for each bar category to classify the charts into one of four categories of bar chart and setting hyperparameters with batch sizes of 25, 50 epoch, SGD optimizer to train and validate the model. After the charts' classification, we used our written script for creating the table and LaTax script of the table on the extracted bar. We found with this method that we should have an

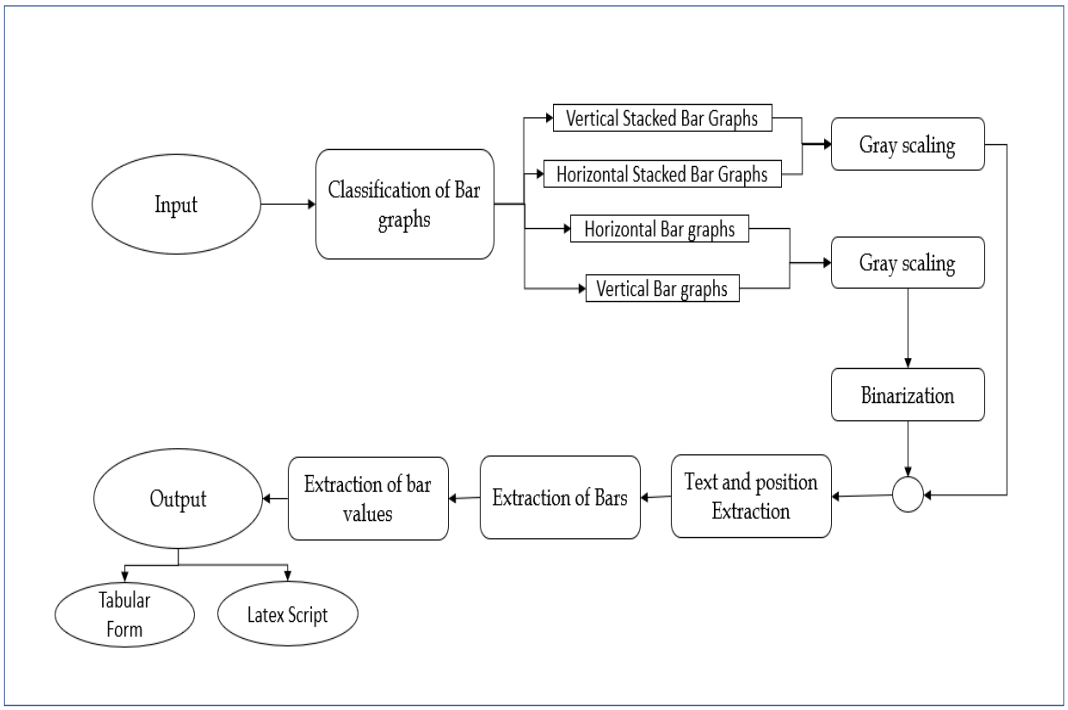


FIGURE 4.2: Baseline 1: Flow-Chart of Image Processing based approach showing that input image passing through various stages and converting into output table and LaTax script

automatic model to convert images to LaTax script.

4.2.2 Baseline 2: Sequence to Sequence model (sequence to sequence)

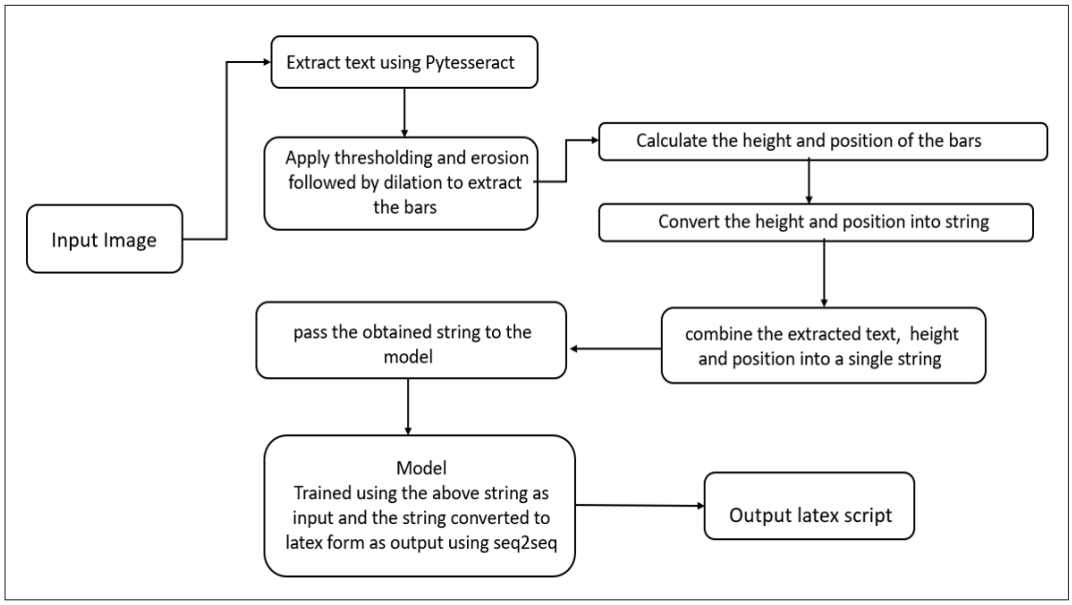


FIGURE 4.3: Baseline 2: Flow-Chart of proposed method showing that input image passing through various stages and converting into output LaTax script

This baseline uses the sequence to sequence model, an encoder-decoder based machine translation that takes an input text and converts it into another text sequence. We used sequence to sequence encoder-decoder LSTM based RNN architecture. We used word-level NMT (Neural Machine Translation). We used inbuilt embedding layers available in keras API so that each word can be mapped into a fixed-length vector.

Model architecture generated using the utility of keras. The model was trained for 50 epochs. Furthermore, a batch size of 128 over a dataset generated using matplotlib in python over 25000 samples.

4.2.3 Baseline 3: Image to LaTax (Im2LaTax)

We used publicly available tensorflow code of image to LaTax generation. This model uses Deep CNN Encoder with LSTM Decoder with attention for image to laTax. After successfully run the code on Im2LaTax dataset, we employ this model for our synthetic chart images for training and than ChartReal images for testing.

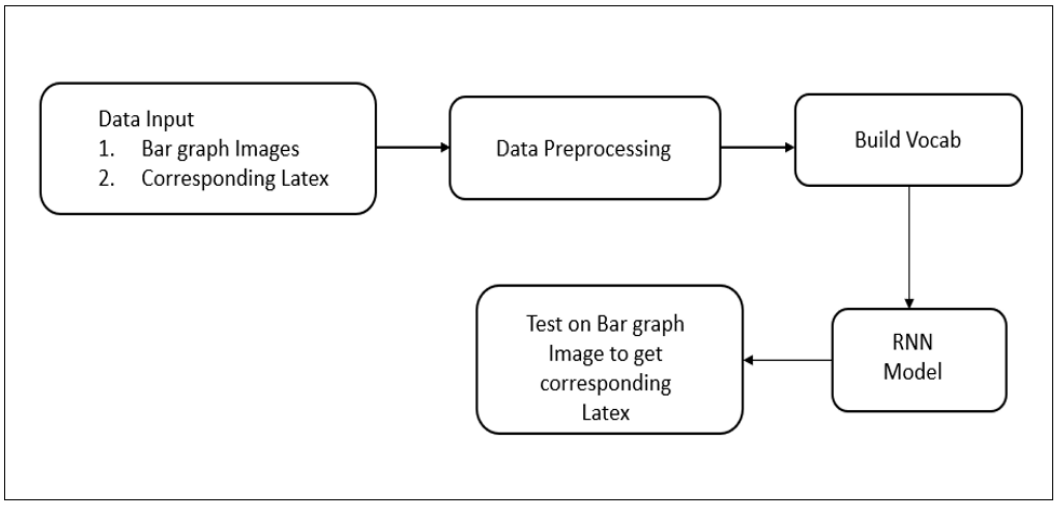


FIGURE 4.4: Baseline 3: Flow-Chart of proposed method showing that input image passing through various stages and converting into output LaTax script

Chapter 5

Performance Evaluation

5.1 Performance Evaluation

We analyzed that our proposed solution performs very well for the synthetic images dataset with the BLEU score of 0.48 as these images follow a similar pattern and are less variable. It contains horizontal bar charts, vertical bar charts, stacked horizontal and stacked vertical bar charts. Nevertheless, when testing the accuracy over another set of the dataset we annotated, accuracy is not much satisfactory as it fells considerably. For the randomly generated dataset, accuracy is low because of if various graphical effects to looks that images attractive enough. To improve the accuracy, one should improve the tesseract engine exactitude because the engine's efficiency depends on the tesseract engine. Another way to improve accuracy is using recurrent neural network to extract text regions and using more accurate OCR. The loss is calculated as:

$$Error = \frac{|Value_{pred} - Value_{GT}|}{|Value_{GT}|}$$

5.2 Experimental Results

We used BLEU score [Papineni et al., 2002] for the evaluation of quality of our translation. We analyzed that the model predicts the correct table structure and labels in most cases on training data because, as in the training data, there were fewer labels than 25k data. However, it is not able to do numeric manipulations. Nevertheless, for new dataset with random string as labels, it fails to detect even the same table structure.

5.3 Summary and Future Work

We introduced the more realistic and complex ChartReal dataset and analyzed the sequence to sequence models on our synthetic dataset and ChartReal dataset. In the future, our object will be to get more accurate results and include all types of images from other books and newspapers also. We are working on the shortcoming of our baselines to improve the results. We are exploring and experimenting im2latex architecture currently. We are making our dataset large so that we can train any model on our data itself.

Bibliography

Cliche, Mathieu et al. (2017). "Scatteract: Automated extraction of data from scatter plots". In: *CoRR* abs/1704.06687. arXiv: 1704.06687. URL: http://arxiv.org/abs/1704.06687.

Deng, Yuntian, Anssi Kanervisto, and Alexander M. Rush (2016). "What You Get Is What You See: A Visual Markup Decompiler." In: abs/1609.04938. URL: http://dblp.uni-trier.de/db/journals/corr/corr1609.html#DengKR16.

Hochreiter, S. and J. Schmidhuber (1997). "Long Short-Term Memory". In: *Neural Computation* 9.8, pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

Jobin, K. V., A. Mondal, and C. V. Jawahar (2019). "DocFigure: A Dataset for Scientific Document Figure Classification". In: *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*. Vol. 1, pp. 74–79. DOI: 10.1109/ICDARW.2019.00018.

Kafle, Kushal et al. (2018). "DVQA: Understanding Data Visualizations via Question Answering". In: *CVPR*.

Kafle, Kushal et al. (2020). *Answering Questions about Data Visualizations using Efficient Bimodal Fusion*. arXiv: 1908.01801 [cs.CV].

Kahou, Samira Ebrahimi et al. (2017). "FigureQA: An Annotated Figure Dataset for Visual Reasoning". In: *CoRR* abs/1710.07300. arXiv: 1710.07300. URL: http://arxiv.org/abs/1710.07300.

Kim D.H., Hoque E. and pp. 1-13 April 2020 Agrawala M. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (2020). *Answering Questions about Charts and Generating Visual Explanations*.

Liu, Xiaoyi, Diego Klabjan, and Patrick NBless (2019). *Data Extraction from Charts via Single Deep Neural Network*. arXiv: 1906.11906 [cs.CV].

Papineni, Kishore et al. (Oct. 2002). "BLEU: a Method for Automatic Evaluation of Machine Translation". In: DOI: 10.3115/1073083.1073135.

Pasupat, Panupong and Percy Liang (July 2015). "Compositional Semantic Parsing on Semi-Structured Tables". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, pp. 1470–1480. DOI: 10.3115/v1/P15-1142. URL: https://www.aclweb.org/anthology/P15-1142.

Rampal, Harshit and Aman Mohanty (2020). *Efficient CNN-LSTM based Image Captioning using Neural Network Compression*. arXiv: 2012.09708 [cs.CV].

Reddy, Revanth et al. (2019). *FigureNet: A Deep Learning model for Question-Answering on Scientific Plots*. arXiv: 1806.04655 [cs.LG].

Savva, Manolis et al. (2011). "ReVision: Automated Classification, Analysis and Redesign of Chart Images". In: *ACM User Interface Software & Technology (UIST)*. URL: http://vis.stanford.edu/papers/revision.

Sherstinsky, Alex (2018). "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network". In: *CoRR* abs/1808.03314. arXiv: 1808.03314. URL: http://arxiv.org/abs/1808.03314.

Smith, Ray and Google Inc (2007). "An overview of the Tesseract OCR Engine". In: *Proc. 9th IEEE Intl. Conf. on Document Analysis and Recognition (ICDAR*, pp. 629–633.

Tafti, Ahmad Pahlavan et al. (2016). "OCR as a Service: An Experimental Evaluation of Google Docs OCR, Tesseract, ABBYY FineReader, and Transym". In:

Wang, Hao et al. (Aug. 2018). "A Neural Question Answering Model Based on Semi-Structured Tables". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 1941–1951. URL: https://www.aclweb.org/anthology/C18-1165.

Zhu, Xinxin et al. (2020). *AutoCaption: Image Captioning with Neural Architecture Search*. arXiv: 2012.09742 [cs.CV].