HIGH LEVEL DESIGN DOCUMENT

VISUAL QUESTION ANSWERING ON STATISTICAL PLOTS

UE18CS390A – Capstone Project Phase – 1

Submitted by:

Sneha Jayaraman	**PES1201802825**
Sooryanath I T	**PES1201802827**
Himanshu Jain	**PES1201802828**

Under the guidance of

Prof. Mamatha H.R.
Professor
PES University

January - May 2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

TABLE OF CONTENTS

1. Introduction

The aim of the project is to build a Visual Question Answering system which accepts statistical plots along with questions on the plot with respect to the elements of the plot (such as intersection of the curves, area under the curve, median value and few other varieties of such relational queries) and provides answers to the questions posed.

The system should discover relationships between elements of a plot and provide relational reasoning to answer questions on the plot.

Given an image of a statistical plot and a corresponding question, the model must be able to generate a representation of the image, parse and understand the query, and generate a suitable reply. Therefore, it involves an understanding of image and the query language to be able to provide for visual reasoning.

High Level Design:

Question Answering System for Charts take in statistical charts and questions related to them as input. Visual features from the charts have to be extracted and preprocessed. This can be done using techniques like image processing and Optical Character Recognition (OCR). The questions provided as an input need to be pre-processed using NLP techniques. Important details from the image and the questions need to be mapped and the corresponding answer should be predicted. Deep learning methods and architectures will be employed to predict the output.

Inputs are validated by the system to recognize only statistical charts. Any other images will be rejected by the system. The system will be trained on huge amounts of data and the results will be validated against the true answers. The parameters for the model will be tuned to provide the most optimal answer as the output.

2. Current System

There have been attempts in the recent past to improve machine reasoning capabilities through visual question answering systems on graphical plots. The RN architecture and the CNN-LSTM architecture form baseline comparison models with an accuracy of 75%

and 60% respectively. This in comparison to human accuracy falls short by a large margin.

A recent paper publication introduced the FigureNet architecture that was able to achieve an accuracy of approximately 85% on an open-source dataset. This however, only gives a yes/no binary output to a question posed, and is limited to only bar and pie charts.

Another adaptation to this model, showed significant enhancements in terms of being able to answer open-ended questions on a different synthesised dataset.

This domain of visual question answering on statistical plots, however, has a lot of scope of improvement in terms of future enhancements of these models. There is a possibility of expanding the types of charts to those beyond bar and pie charts or even improving on accuracy through model adaptation.

3. High Level System Design

Component Diagram:

Input: There are two inputs to our model that the user needs to provide. The first input is an image – that depicts a statistical plot and the second input is a relational question on the image.

Our Model: The design consists of 3 primary components.

1. Image Encoding Module

This is a module that takes in the image as its input and produces image feature vectors as its output. Feature vectors are a vector representation of the image that encodes all of the relevant information in the image.

2. **Question Encoding Module**

   The input to this module is the question (English Language) and the output is a question embedding. The question embedding captures all of the relevant information in the question in a format that is suitable for further modelling.

3. **Result Estimation Module**

   This module takes in as input the output produced by both of the previous modules. It produces as an output the final answer to the question. Thereby, it finds the correlation between the image and the question based on previous training on a variety of image-question pairs.

Output: The output is an answer to the question that was posed on the image.

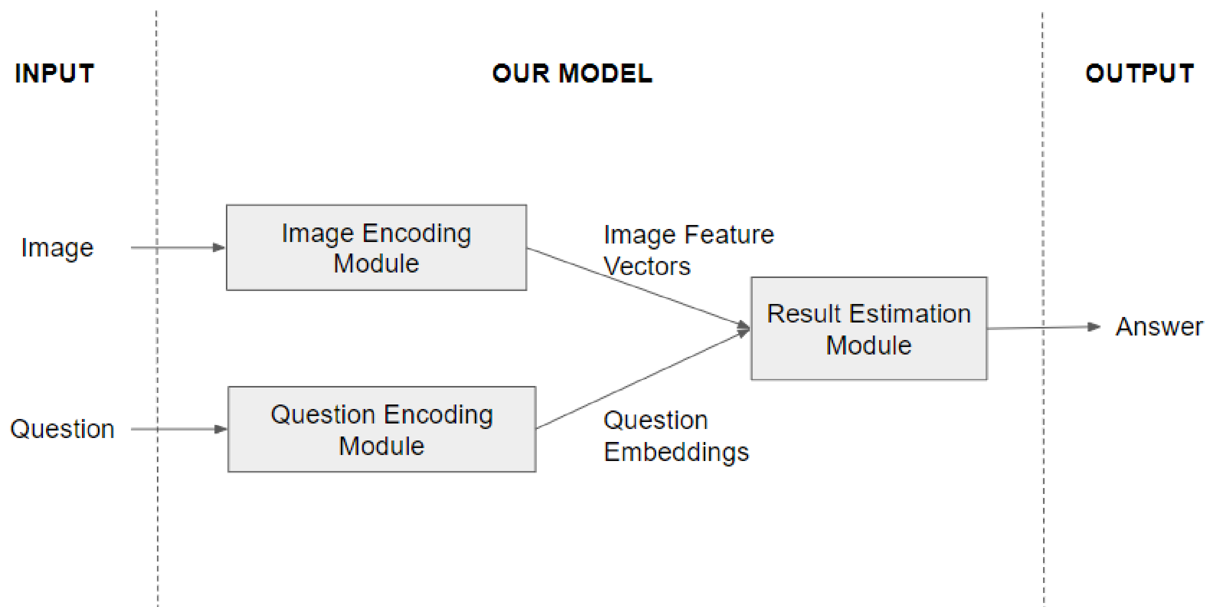
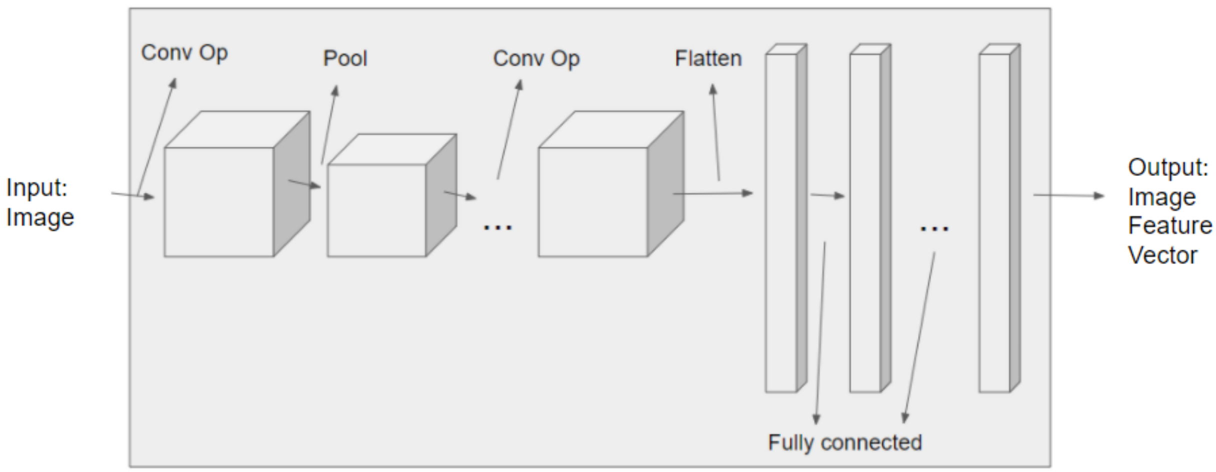
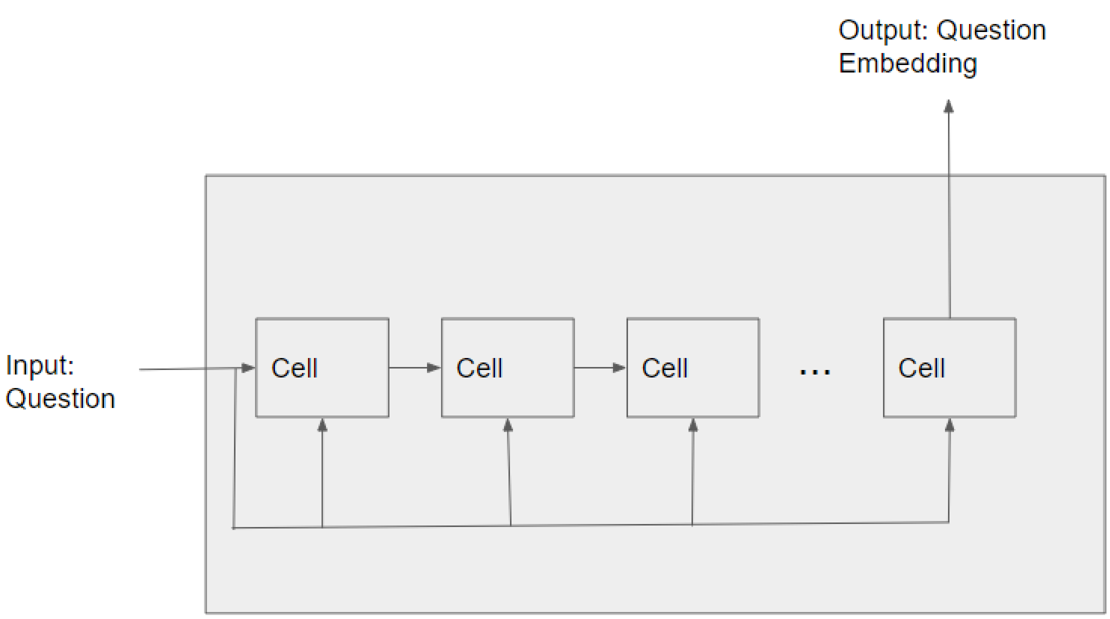


Image Encoding Module:
This consists of a sequence of convolution operations and pooling operations, followed by fully connected layers that would output a flattened image feature vector.



Question Encoding Module:
This consists of a series of recurrent cells, that would take each word of the input question as its input, to finally output a question embedding.



Result Estimation Module:
This module concatenates the image feature vector and the question embedding, and passes that on to fully connected layers. The output is the answer to the question.

4. Design Details

4.1. Novelty

Visual question answering on statistical plots is one step towards better machine reasoning capabilities. This domain has been explored before, but there is scope for further research. This project will help all business/ data analysts easily explore and analyse plots via question-answering. We aim to develop a comprehensive automated model with all the research and learnings done so far, so as to improve accuracy of existing models, with focus on specific charts.

4.2. Innovativeness

There exist a lot of models for object detection in images. Statistical plots, though images, have different properties that make the use of simple object detection models insufficient for this use case. In recent years, there has been a focus on improving machine reasoning capabilities, and the use of scientific deep learning models to do so. This project will explore the possibilities of further enhancement to this relatively new field/ idea, that has wide use cases in the real world.

4.3. Interoperability

The model created (saved) can be run on any host irrespective of the operating system provided the desired modules are made available on the platform making it compatible. The requirements in terms of modules can be installed from a normal requirements.txt file.

4.4. Performance

The model is expected to draw statistical inferences based on the question and the input passed with a good and acceptable accuracy level. Accuracy is the metric to evaluate the performance of this model and comparison parameter would possibly be with respect to human responses/inferences from the graph.

4.5. Security

The model trained needs to be protected on a local machine or online if deployed onto the cloud . Modification of firewall rules or enabling regular scans may help in securing the build model.

4.6. Reliability

As the product is developed by following practices in deep-learning , machine learning and imaging , there is no certain fixed level of reliability that can be set for the product. Reliability is not completely independent of the inputs to the model , hence reliability varies with respect to the context and type of inputs passed in.

4.7. Maintainability

As the product is just a model , maintaining the model isn't a difficult task , it only requires a machine to reside on and a browser / interface to access. Alongside we have the project documentation and upfront planning documents if in case the project needs to be reworked in the future.

4.8. Portability

The developed model is implemented / coded up using python and python rooted modules along with standard open source tools offered by google such as TensorFlow and Keras. The code units can be packaged and run on any host irrespective of the operating systems provided they have python and the necessary modules in addition.

4.9. Reusability

The trained model must be available at ease to use it ,just by choosing an image input from the local drive or file-system . Similar to drag and drop or attaching files. The user must be able to navigate through the interface even with minimal exposure towards computing technologies.

4.10. Application compatibility

The model developed is compatible and interoperable based on the conditions provided in interoperability and portability sections . However , the input data must be an image which is concerned with the scope of the project.

4.11. Resource utilization

Local Disk Storage for input dataset storage and CPU / GPU for running the training code (or simply for training the data model).

4.12. Data Sharing

As there are lots of graphical images to be collected , the data and statistics storage will be done to maintain the correct functioning of the model and to reconstruct what went wrong in case of any system - failures by constructing checkpoints. The datasets if artificially generated need to be secured locally and not be made available for commercial usages.

Appendix A: Definitions, Acronyms and Abbreviations

ACRONYMS/TERMS	EXPANSIONS / DESCRIPTION
CPU	Central Processing Unit
GPU	Graphics Processing Unit

TPU	Tensor Processing Unit
OCR	Optical Character Recognition
VED	Visual Edge Detection
NLP	Natural Language Processing
CNN	Convolutional Neural Network
LSTM	Long Short Term Memory
GRU	Gated Recurrent Unit
Module	A Software Component

Appendix B: References

Title	Version Number	Date	Publishers	Reference
CNN and LSTM architectures	1.0	10/04/2021	Moses Soh	[1]
Combining LSTM and CNN architectures	1.0	12/04/2021	Jason Brownlee	[2]
Plot QA	1.0	12/04/2020	Mitesh Khapra Nitesh Methani , Pritha Ganguly and Pratyush Kumar	[3]

Appendix C: Record of Change History

#	Date	Document Version No.	Change Description	Reason for Change
1.	16/04/2021	1	Added all of the detailed design and appendix	More detailed in design

Appendix D: Traceability Matrix

Project Requirement Specification Reference Section No. and Name.	DESIGN / HLD Reference Section No. and Name.
3 : Functional Requirements	3 : High Level System Design
5 : Non Functional Requirements	4 : Design Details