Shahid Beheshti University

Cyberspace Research Center

Department of Information Technology Engineering – Multimedia Systems

MASTER OF SCIENCE THESIS

Solving the problem of visual question answering on images including diagrams (Figure Question Answering)

By:

Marzieh Malekzadeh Mahani

Supervisor:

Dr. Ali Nadian-Ghomsheh

September 2022

Table of content

Table of figures

Table of tables

Abstract

Figure Question Answering (FQA) is a multimodal task that aims to resolve a high-level image comprehension issue. The network is tasked with giving the right response to a set of scientific figure-question pairs, such as: Does the red Gaussian in the figure have the highest variance? Extraction of rich features for figure representation is one of the primary issues in FQA. Additionally, the provided datasets don't include any difficult numbers. In this study, we propose a FQA pipeline which combines the transformer architecture for enhanced figure representation. This contrasts with traditional techniques that employ CNNs. In addition, we offer the SBU-FQA dataset, which is more complex and diversified than earlier datasets. There are variable numbers of figures in each plot of the dataset's six main figure kinds. In order to make the task harder, 23 question templates were created. The proposed method was evaluated on SBU-FQA and showed significant improvements compared to the base-line and state-of-the-art FQA methods. Additionally, the results showed that when used with the SBU-FQA dataset, the prior technique performs less well. The maximum accuracy of the current models was only about 61, but this model was able to achieve a reasonably respectable accuracy of about 74.30 after being applied to the new dataset. The dataset is made freely available, and the code to regenerate the findings will also be made public.

Chapter 1

1. Introduction

Today, despite the growth of human-computer interaction and the feeling of need for this field, activities related to this field have gradually become very important. Visual question and answer, as one of the most important ways to communicate between humans and computers, is not an exception to this rule, and in recent years, creating a system that is able to answer the questions raised in connection with an image is the concern of many activists and enthusiasts in the field of artificial intelligence [1]. It is natural that ordinary people are able to answer many questions related to it by seeing an image, but for an artificial intelligence system, understanding the question in terms of meaning, processing and understanding the image, finding the connection between the two and generating an appropriate answer is very challenging. However, significant advances in machine learning in both computer vision and natural language processing have made it possible to design and build such a system, one that takes an image and question in natural language as input and outputs the answer in natural language. There is a special branch of VQA called figure question answering[1] [2]. The image that this system receives as input can contain information from a chart, plot or graph. Like real world images, these images are also composed of points, lines and different colors. Based on the nature of the graphs used, the content of the questions in this system is usually related to numerical values, how the graphs, points, lines are placed in relation to each other, the position of the elements in relation to the coordinate axes, common points of the graph, the surface area under the graph and many concepts. One of the most important differences between an FQA system and other

[1] FQA

VQA branches is the analysis of the exact placement of elements in the image; this issue may not be important in some other studies [3].

1 - 1 The overall structure of the video question and answer system:

After getting a general picture of the structure of the VQA system, it is necessary to briefly examine the parts of this system. As you can see below; after entering the image and question into the system, some processing is necessary to continue the work. The most important processing is feature extraction from the input data, which can be done in different ways. The feature algorithm is chosen specifically for the input image and query string, and then we reach the feature fusion stage. This step can also be done by many different methods that are chosen in each problem according to the level of necessity of the problem. In the VQA system, it is decided which features to pay attention to in order to produce a suitable response [4].

1 - 2 Application fields of vision questions and answers:

Due to the fact that in the VQA system, one of the inputs is in the form of an image, so wherever it is possible to capture images or create images, it is also possible to use this system, for this reason, we will have many applications for VQA:

- The field of medical images and disease diagnosis [1, 5, 6],
- Helping blind and visually impaired people to understand images [5],
- Obtaining information from unmanned vehicles [6],
- Helping users understand the meanings in specialized images (diagrams and flowcharts) [5],
- Helping children understand pictures and motivate them to learn [5].

1 - 3 Visual question and answer methods:

In feature extraction from images, there are several methods; choosing a pre-trained convolutional network or building a network from scratch. Choosing any of these methods can make a big difference in the quality of the output [7].

In the part of feature extraction from questions, first of all, how to tokenize questions, using combined or simple methods, each can have a high impact, but in general, if we divide the resulting methods into two categories; The first category is the methods based on the integration of features and the second category is the methods based on the attention mechanism [6].

We know that each part of the input is processed separately and the vector will have its own characteristic. In the methods of the first category, the focus is on how to integrate the feature. Integration can be done using different methods, simple and basic methods, such as; the successive joining of vectors, multiplying and adding together, or more complicated methods, such as; Combination of element-wise multiplication and addition [8, 9].

In the second category, for better results, it is necessary to consider the relational answer between the image and the question string. Each of the words in the input strings can represent an object or a part of the image, but to answer the question, this is not enough, and it is necessary to consider the relationship between the words and the important areas of the image. The attention mechanism [4] includes two different styles in terms of how to pay attention to different areas of the image; Soft attention and hard attention. In addition, the methods based on the attention mechanism can be divided in another way; Bottom-up attention and top-down attention [4], which we will explain in detail in future sections.

1 - 4 Problems and defects:

One of the most important problems in the field of FQA is the insufficient data set in terms of quantity and quality. In terms of quantity, it means the number of data sets, and in terms of quality, it means the variety of data sets. Since the collection of real data sets and the creation of artificial data sets are both time-consuming and expensive, it can be said that the number of FQA data sets has not grown much due to the new nature of this research. For this reason, active researchers in this field have gone less to produce new shapes and diagrams [10]. Another problem is in the models developed in FQA. Although a lot of knowledge and time has been spent in producing these models, most of them show high accuracy only in limited cases of inputs and will not perform significantly in the case of a new input data [3].

1 - 5 Introduction of the main idea:

In order to solve the mentioned problems, we first decided to produce a new data set that is not limited in terms of quantity. In this section, there is no ceiling in terms of number for generating artificial images from different graphs (horizontal, vertical, circular, linear, dotted line and Gaussian) and as many as necessary are generated from each data sample. Additionally, this dataset contains new graphs that were not present in other FQA datasets. To create more complexity and challenge, we added the Gaussian diagram, which is one of the most widely used diagrams in mathematics, statistics and computer science, to the dataset. Further, by implementing existing models and examining their strengths and weaknesses, we tried to develop models that have high accuracy in answering questions. Also, without being biased towards a particular graph or graphs, give appropriate answers to questions related to all graphs.

Chapter 2

2. Feature extraction in VQA

2 - 1 Feature extraction from image

One of the two preliminary operations performed in the VQA system is feature extraction from input images, which leads to the generation of feature vectors such as each image. The feature vector of each image is actually the numerical description of that image, which helps to make various computational operations easier. Proper training of deep learning models from the beginning requires large datasets and special computer resources, therefore, by using pre-trained neural networks, feature extraction can be done more easily. A pre-trained network is a network that has already been trained on a large dataset and its weights have been initialized. The large volume of the dataset indicates the training of the model as much as possible, and therefore the weights of this network can be trusted [7, 11].

The general structure of these networks will be as follows; First, the convolution part, which includes convolution and pooling layers, then densely connected classifier layers. The convolution layers themselves are divided into two parts; Initial convolutional layers that output more detailed features such as colors, edges, and texture, and final convolutional layers that output more general concepts such as cat ears, human hands, and balls. So, we noticed that as we go from the beginning of the convolutional layers, the obtained features are transformed from a more detailed state to a more general state. Next, in the densely connected classifier section, the obtained features are classified and some features are discarded according to the needs of the problem. For example, in one problem,

the location of objects may not be important, so this feature is discarded, but in another problem, the location of objects is considered important. For this reason, when we intend to use pre-trained networks, it is better to use the convolution part and write the classification part according to the needs of our problem, or if we want to use the classification part of these networks, according to the similarity of the problem that the network is designed for that and let's pay attention to our problem [7].

It is one of the most famous neural networks for feature extraction from CNN. Table 1 provides information about several different types of CNN networks, these networks are widely accepted and used. Most of the researches conducted in the field of VQA have used these networks to extract features from images. The number of layers, the dimensions of the input image, the dimensions of the last extracted feature vector and the reported error of each model can be seen in the Table 1 [7].

Table1 Pre-trained CNN networks

Above models CNN	Age	Number of layers	Entrance dimensions	Output dimension (number of features)	Error reported
AlexNet	2012	8	227×227	4096	16.4
ZFNet	2013	8	227×227	4096	11.7
VGGNet	2014	19	224×224	4096	7.3

GoogleNet	2014	22	229×229	1024	6.7
ResNet	2015	152	224×224	20148	3.57

2 - 2 Feature extraction from text:

The term text embedding means converting phrases and words into numbers that are better understood by the computer, in such a way that a text is entered and a vector containing numbers to describe the text is output. This method is used to model language and train features in processing. Natural language is used. The main idea used in the design of all text embedding methods is to extract their correct meanings and concepts; Because words and expressions have different meanings in different texts and applications. Choosing the best method for embedding words requires trial and error; Because the existence of many algorithms in this field and the different structure of words and phrases can affect the results obtained.

In VQA, to process questions, we need to extract features from the text, because most deep learning models are not able to process string data, in the following, we will mention three methods to implement feature extraction from the text, in each of these methods, the text or the string as input and the results will be displayed as a feature vector [7].

• Methods based on counting

• Prediction-based methods

•Combined method

2 - 2 - 1 One-hot-encoding method

The simplest method of this category is called One_Hot_Encoding, the result of which is a vector of length |V| Is. In this method, we measure the occurrence of each word in relation to the places in the entire text, and we only answer the question of how many places and in which place each word has appeared. How this method works is shown in the Table 2.

Table 2 One-hot-encoding matrix

	1	2	3	4	5	6	7	8	9	10	11
visual	1	0	0	0	0	0	0	0	0	0	0
question	0	1	0	0	0	0	0	0	0	0	0
answering	0	0	1	0	0	0	0	0	0	0	0
is	0	0	0	1	0	0	0	0	0	0	0
a	0	0	0	0	1	0	0	0	0	0	0
task	0	0	0	0	0	1	0	0	0	0	0
of	0	0	0	0	0	0	1	0	0	0	0
based	0	0	0	0	0	0	0	1	0	0	0
on	0	0	0	0	0	0	0	0	1	0	0
given	0	0	0	0	0	0	0	0	0	1	0
image	0	0	0	0	0	0	0	0	0	0	1

The problem of the mentioned method is that it does not pay attention to the similarity between the words and the relationship between them, and only by measuring the Euclidean distance between the two words, it comes to the conclusion that there is a distance of √2 between each word and the other word and with its similar word. It has a distance of 0. To solve this problem, a new method was proposed, which is called the co-counting matrix method. This matrix shows the number of times that both words occur in the presence of each other. To measure this issue, a window of length K is used, and every time a word appears in the neighborhood of K

of another word, one item is included in the matrix. In this method, each row is defined for a specific word such as X, and it indicates the number of occurrences of another word in the K neighborhood of the word X. In this method, words that are not very important can be omitted, on the other hand, due to the large number of words and variables that mostly have zero value, we are faced with a large but empty matrix, and it is better to be able to reduce the number of words. How this method works on the string is shown in Table 3[12].

Table 3 Co-occurrence matrix

	1	2	3	4	5	6	7	8	9	10	11
visual	1	1	1	0	0	0	0	0	0	0	0
question	1	0	2	1	1	0	0	1	1	0	0
answering	1	2	0	1	1	1	1	0	0	0	0
is	0	1	1	0	1	1	0	0	0	0	0
a	0	1	2	1	0	1	2	1	0	0	0
task	0	0	1	1	1	0	1	0	0	0	0
of	0	0	1	0	2	1	0	0	0	0	0
based	0	1	0	0	2	0	0	0	1	0	0
on	0	1	0	0	1	0	0	1	0	1	0
given	0	0	0	0	0	0	0	1	1	0	1
image	0	0	0	0	1	0	0	0	0	0	0

2 - 2 - 2 Prediction-based methods:

In designing this type of methods, neural networks are used as the most basic components. The first model used in prediction-based methods was a 3-layer network, after which all the related works done in this field were inspired by this network.

From advanced word embedding models, both of which are designed based on prediction; They are called CBOW model and Skipgram model. In CBOW, the focus was on predicting the next words; In this way, this model could predict the use of n-1 previous words in a text or phrase from its previous learning. In the Figure 1 it is clear that the input of CBOW in the form of One_Hot representation is two words visual and answering and its output will be the probability of presence of each of the words in the text after the input words [13].

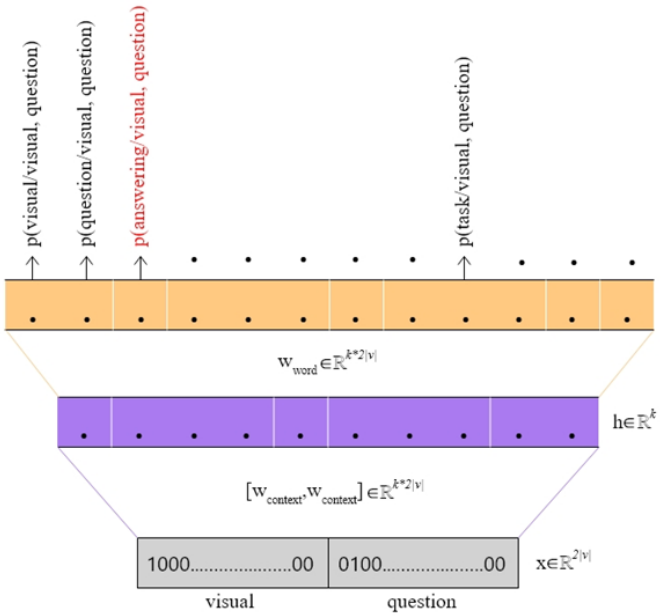


Figure 1 CBOW structure

In the Skipgram model, the prediction is done in a more advanced and complete way than the CBOW model; In this method, the model has the ability to predict the words before and after the nth word, so that the input of the model is the One_Hot representation of a word, and the output is the probability of the presence of each word in the position before and after the input word. Figure 2 shows Skipgram in action. Paying attention to the relationship between words in the string and extracting each feature vector based on one word, this method is called word2Vec, which means the correspondence between each word and a feature vector [14].

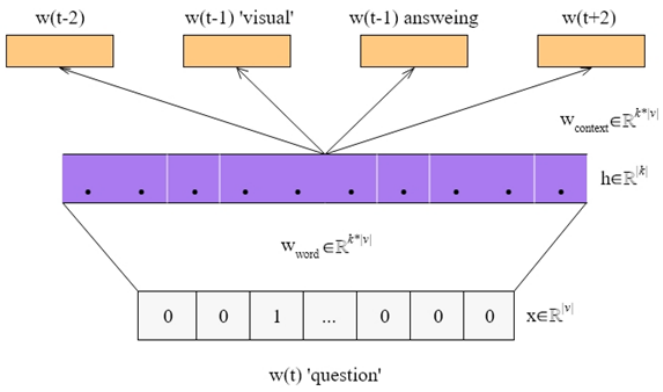


Figure 2 Skip-gram structure

2 - 2 - 3 Combined method:

This category of methods consists of the combination of two methods based on counting and based on prediction. One of the most famous hybrid methods is the GloVe (global vector for word representation) method, which uses information extracted from the co-counting matrix. In this method, unlike prediction-based methods, it does not focus on the meaning of a word and the relationships of a word with other words; Rather, the focus is on the entire meaning of a single text or phrase, and a feature vector is extracted for each sentence or phrase. As mentioned, this method tries to match the text with the feature vector, that's why it is also called text2Vec.

2 - 3 Commonly used methods in text embedding

In recent years, one of the methods of feature extraction from text that has attracted much attention from researchers is called long-term short-term memory. This method is a recurrent neural network that was created to solve the problem of gradient explosion and gradient fading. The LSTM layer

actually stores information in its own memory units and acts as a bridge between words in a sequence. This network has 3 more gates compared to the traditional RNN network structure in order to provide a better description of the relationship and dependence between data in the long term; Input, output, and forget gates. Input gate and output gate that provide the possibility of reading and writing information in the memory units, and the forget gate resets it when the information in a memory unit is out of date. In the Figure 3 can be seen that the basic structure of the LSTM network [7, 15].

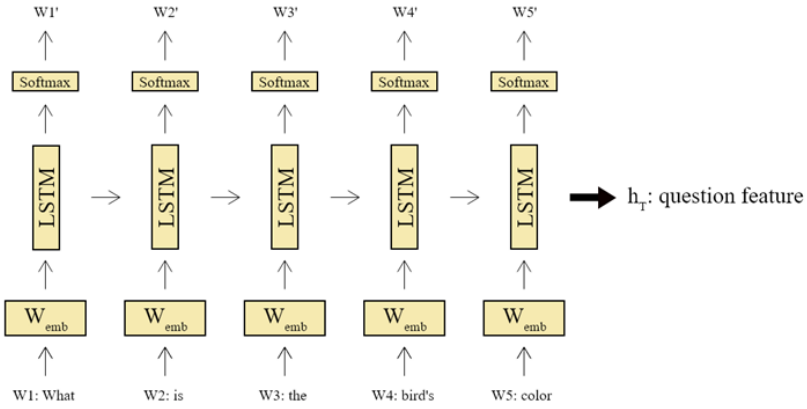


Figure 3 LSTM structure

2 - 4 Fusion of features:

What we said so far was related to how to extract features from input images and strings; We know that each part of the input is processed separately and each of them results in a separate feature vector.

In this section, we discuss how to combine features and explore how combining image and text features can help generate better answers to What we said so far was related to how to extract features

from input images and strings; We know that each part of the input is processed separately and each of them results in a separate feature vector.

In this section, we discuss how to combine features and explore how combining image and text features can help generate better answers to questions.

We examine the methods of combining features in the following 3 figures:

1. Simple and elementary methods of integration

2. MCB method

3. Methods based on attention mechanism

2 - 4 - 1 Simple and basic integration methods:

Sequential joining of vectors, multiplication and addition one by one are considered to be the most basic methods of integrating features, in the last two methods, the dimensions of the vectors must be equal, because these two operators are performed peer-to-peer between the elements of the vectors, and for each element in the first vector must have the same element in the second vector. In the joint research of Malinovsky and his colleagues [16], it was found that the multiplication of the elements by one by one has a better result than the other two methods; However, if the features extracted from the image are normalized, it will have a greater impact on all three methods, especially in the sequential joining of vectors and the addition of similar elements, and after normalization, they will achieve very good accuracy.

After examining addition and multiplication separately, we introduce a method that is referred to as a combination method. The way this method works is that it puts multiplication and addition together in such a way that they form polynomials [8].

$$(1 + x_1 + x_2 + x_3 + \cdots + x_d)(1 + y_1 + y_2 + y_3 + \cdots + y_d) \quad (1)$$

In the said polynomial, x_i is the vector of features related to the image and y_i is the vector of features related to the text.

In this method, the features created from addition and multiplication are taken advantage of; Because the idea behind this method was based on the fact that the features resulting from multiplication and addition are different from each other and a method should be provided that can use both of them together.

2 - 4 - 2 Multimodal Compact bilinear pooling method

In this method, after extracting the feature vector, an algorithm called Count sketch, which is based on counting, is used. sketch is a translation language for feature vectors based on the number of repetitions of letters in a sequence. To explain how this algorithm works, it is necessary to use a simple example in this case. Pay attention to the input string and hash function below.

Input string: [A, B, K, A, A, K, S, inf]

Table4 Hash function

	H1	H2	H3	H4
A	1	6	3	1
B	1	2	4	6
K	3	4	1	6
S	6	2	4	1

If we imagine the input string as a vector that contains various characters with different repetitions, we need a function that helps us compress the input data through hashing. This is a selective hashing

function; the number of its criteria is variable and the number of rows is equal to the number of unique elements in the input sequence. The table below is the initial hashing table of the input string, all of which are initially set to zero and will be filled in the future steps based on the values of the hashing function.

Table5 Count sketch step 1

	0	1	2	3	4	5	6
H1	0	0	0	0	0	0	0
H2	0	0	0	0	0	0	0
H3	0	0	0	0	0	0	0
H4	0	0	0	0	0	0	0

In this step, after reading each line of the hashing function, we put the value of one in the given address. For example, in the line related to the letter A, we see the addresses (1,6,3,1) respectively, so in the addresses (H1,1), (H2,6), (H3,3) and (H4, 1)) respectively we put the value 1 to reach the following table.

Table6 Count sketch step 2

	0	1	2	3	4	5	6
H1	0	1	0	0	0	0	0
H2	0	0	0	0	0	0	1
H3	0	0	0	1	0	0	0
H4	0	1	0	0	0	0	0

After continuing the said process and creating hashing tables, the last table we see is the following table. Now it is necessary to read this table, the reading method will be such that the values in the line corresponding to each input element in the hashing function will create a series related to that element. For example, by looking at the hashing function, we see that the line opposite the letter A gives us the addresses of houses (1, 6, 3, 1) respectively. We read the values in these houses from the last table, which will give us the number series (4,3,3,4), then among the values in this series, we choose the lowest one, and in all sequences instead of all A's We see that we put the number 3 only once. In the last step, the median value can be used instead of the lowest value, and this does not make a difference in the overall performance of the method.

Table7 Count sketch final step

	0	1	2	3	4	5	6
H1	0	4	0	2	0	0	1
H2	0	0	2	0	2	0	3
H3	0	2	0	0	2	0	0
H4	0	4	0	0	0	0	3

After obtaining the count sketch description of the vectors, convolution multiplication operation is used to merge them. To implement this method, the vectors are transferred to frequency space by Fourier transform, then the vectors are converted into a single vector by simple multiplication operation, it is necessary to remember that simple multiplication in frequency space is equal to convolution multiplication in space-time space. Next, the resulting vector is returned to the space-time space with the inverse Fourier transform operation, and finally, we have the product vector of the convolution of two image and text feature vectors. The general architecture of this method can be seen in the Figure 4 [17].

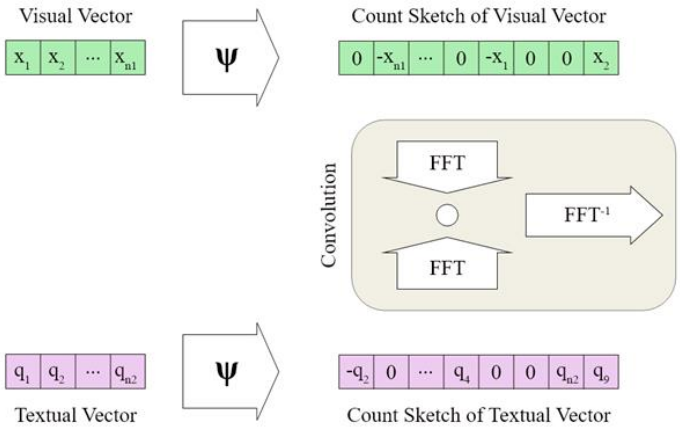


Figure 4 MCB structure

2 - 5 Methods based on attention mechanism

In order to achieve better results in visual question answering, it is necessary to consider a semantic relationship between the image and the question string in the production of the answer. Each of the words in the input string can represent an object or a part of the image, but to answer the question, this alone is not enough, and it is necessary to obtain the relationship between the words and the important areas of the image through inference.

In the visual question answering, we discuss one of the aspects of the attention mechanism, which is paying attention to the location of the object. In this mechanism, an attempt is made to pay attention to the correct places and points of the image by using the meanings available in the question, and other areas are left aside. The mechanism of attention in terms of how to pay attention to different areas of the image includes two different styles; Soft attention and hard attention. Soft attention means selecting areas from the image that help answer questions with a high probability, and hard attention means sending only a specific area to the output and removing areas and extra information.

In the following, we will examine and compare two different ways of implementing the attention mechanism; Attention from top to bottom and attention from bottom to top.

Bottom-up attention: In this type of attention mechanism, important areas and objects in the input image are extracted without considering the meanings and relationships of the words in the input string. Certainly, the regions obtained from this method are not all important to answer the question and there is additional information.

In the implementation of this part, Faster R-CNN is used, which is a kind of object detection method. This method, which was designed by Shaoqing Ren et al. [18], was able to help detect the location of objects in images and created the necessary complexity for this task. This was while most object detection methods relied on a hypothesis about the location of the object to be detected. This issue over time led to the development of location detection algorithms, and for this reason, the accuracy of most object detection methods relied to a large extent on the power of detecting the location of the object in them. Faster R-CNN was developed to share all the convolutional features of the image with the object detection network by using a region recommender network. To better understand this network, we explain it in two separate parts.

In the first part, we have a CNN network that provides hypotheses related to the location of each object. This part is called RPN (Region Proposal Network) and its task is to tell the second part of the Faster R-CNN network which part of the image to look at to find a specific object. RPN takes an image as input and outputs a rectangular object, which is actually a suggestion for the location of that object. In addition, this network also returns a numerical value that actually indicates the possibility of that object being a member of a certain class. In this part, a search function is used to select parts that have similar texture and color and places them in separate boxes. In the following, the SoftMax

activation function is used to classify these parts and linear regression to create bounding boxes. In this section, feature extraction is generally performed.

In the second part, there is the object recognition part of Faster R-CNN, which receives the feature tables that are the output of the previous part and performs ROI pooling on it. ROI actually extracts the important regions of the image according to the application from all the regions that are the output of the previous CNN and then gives it to the Fully-connected layer so that the features there are first flattened and classified to be demarcated with boxes. The Figure 5 shows how Faster R-CNN works.

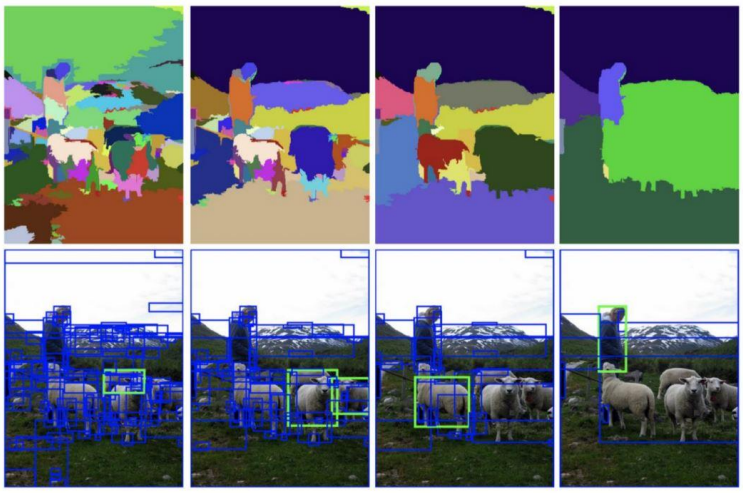


Figure 5 Faster R-CNN performance

Note from top to bottom: In this method, the concepts and connections in the question determine the importance of image areas. After dividing the image into smaller areas, there is a feature vector for each area. In parallel, the attention mechanism applied to the input string recognizes specific words as important, and each of those words refers to a specific region of the image. In this way, a kind of weighting happens to the image areas, and then only the feature vectors of those areas will be effective in producing the answer.

The details of this method are specified in the following equation:

$$\hat{V} = f(h, v) \quad (2)$$

In the above equation, v represents the feature vector related to the candidate areas to be in the focus of attention; h is the feature vector corresponding to the text or form of the question, and f will be the function of applying the attention mechanism to the two categories of said features. Therefore, $\hat{V}$ will be the feature vector corresponding to the regions selected by the attention function. Figure... is a simple representation of how the model proposed in this article works, which was obtained from the study of the documents available in connection with the presentation of the above article by its author, Peter Anderson, at the CVPR 2018 conference, which shows well how the image regions are selected by the mechanism Note and the vector represents the features of the question [19].

2 - 6 How to implement the top-down attention mechanism:

In the research conducted in 2017 by Tenny et.al about how to implement the attention mechanism. The steps of the mechanism are as follows [20].

In connection with each of the image areas that have a number between i=1...k; y_i is the feature vector specific to that region, which is connected with the text feature vector by equation 3:

$$a_i = w_a f_a([v_i, q]) \quad (3)$$

These features are first of a non-linear layer named f_a; are passed and then both are passed through a linear layer to obtain the linear weights of attention. w_a in this relation is a vector containing the trained parameters that is multiplied by f_a.

Next, it is necessary to normalize the obtained attention weights in some way. As seen in equation 4, the SoftMax function is used for this purpose. Then, according to equation 5, the normalized weights are multiplied by the feature vector of the relevant area, and the sum of the feature vector of the areas will be equal to the feature vector of the whole image.

$$\alpha = softmax(a) \tag{4}$$

$$\hat{v} = \sum_{i=1}^{K} \alpha_i v_i \tag{5}$$

After weighting the image areas, we merge the text and image features. Here, as we said before, we can use one of the simple methods of integration, such as multiplying the elements by the same elements. You can see this operation in equation 6.

$$h = f_q(q) \circ f_v(\hat{v}) \tag{6}$$

Since the question answering problem has been introduced in the researches, it is a multi-answer problem. That is, there can be several correct answers for a question that is asked in connection with an image; In this way, at the last stage, when the h vector is created in relation 6, we multiply this vector once by the weights trained by the textual data and once again by the weights trained by the image features, and the areas obtained from these two operations are combined.

Currently, we have N number of regions, which are selected using the sigmoid activation function, and the possible answers are sent to the output with their corresponding probability. The sigmoid function normalizes the probabilities of the regions in such a way that they are in the interval (0,1). You can see the explanations given in equation 7 and 8:

$$\hat{s} = \sigma(w_o f_o(h)) \tag{7}$$

$$\hat{s} = \sigma(w_o^{text} f_o^{text}(h) + w_o^{img} f_o^{img}(h)) \tag{8}$$

Chapter 3

3. Introduction of datasets:

In this section, we review the datasets available in the field of visual question and answer, focusing on the datasets related to the FQA field. In this regard, we must first explain that each dataset is necessary;

- Be large enough,
- Be diverse in their images, questions and concepts,
- Include separate sections for fair evaluation of models,
- Have the least amount of bias [7].

Most of the existing datasets contain a triad of an image, a question, and the correct answer to that question. General capabilities required to answer questions include the following;

- Ability to recognize objects, features and spatial relationships,
- The possibility of counting, making logical inferences and making comparisons,
- Appropriate and intelligent use of real-world information [7].

The existing datasets for visual question answering can be classified based on 3 factors;

- Image type
- Question and answer format
- How to use real information [7].

The images in the dataset are divided into 3 categories:

- Natural images
- Simple graphical image
- Artificial images [7].

Now we will introduce the relevant datasets in the field of FQA and explain the characteristics, strengths, and weaknesses of each one;

3 - 1 CLEVR dataset

This collection includes artificial images of three-dimensional objects, which are divided into 3 shapes: square, cylinder and sphere. They are made of plastic or metal and come in two sizes, large and small. There are objects in this collection in gray, purple, turquoise, blue, green, yellow, red and brown colors, and in total, 96 unique modes are obtained from the combination of the aforementioned features, which creates an acceptable variety in the images of this dataset [21].

In the educational part of this collection, there are 70,000 images and 700,000 questions in different formats. In the validation section, 15,000 images and 150,000 questions, and in the test section, 15,000 images and 150,000 questions were asked. The questions of this series are presented in 5 formats; Questions are related to the presence or absence of objects, counting objects, comparing numbers, properties of objects, and comparing properties [21].

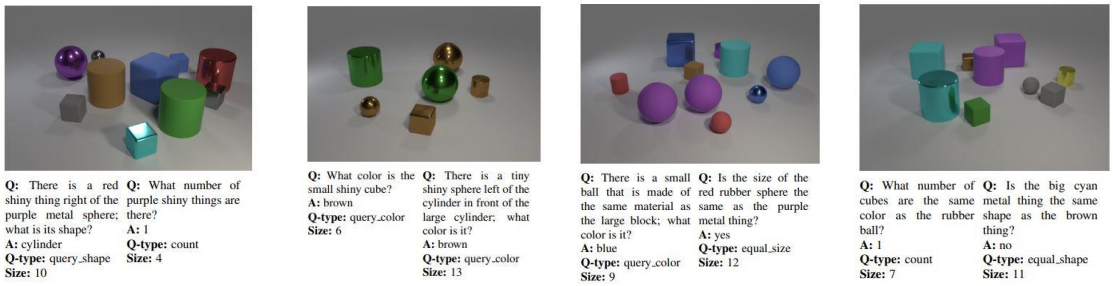


Figure 6 Sample images of the CLEVR dataset[22]

3 - 2 FigureQA dataset:

One of the largest collections of FQA model tutorials, including artificial images of various graphs. This collection contains 180,000 images and more than 2 million pairs of questions and answers. The images of this collection are divided into 5 classes; horizontal column, vertical column, circle, line, dotted line charts. The charts in these images are randomly composed of 100 different colors. The questions of this collection are presented in 15 different formats and are from topics such as: the largest, the smallest, the median, the area under the graph line and the commonality between the graphs. All these questions are answered in binary and in the form of yes or no questions [3, 10].

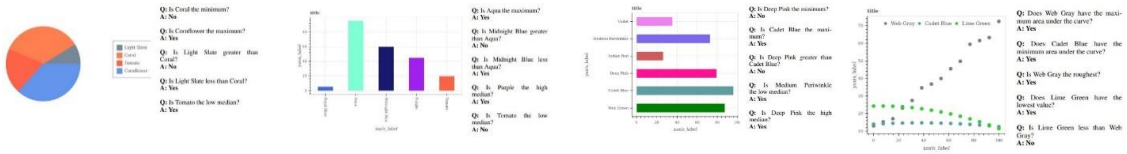


Figure 7 Sample images of the FigureQA dataset

3 - 3 DVQA dataset:

This collection contains 300,000 artificial images of bar graphs and more than 3 million questions. The questions are created in the form of short answers and in 26 different formats. One of the unique

features of this collection is the use of OOV[2] technique. In the experimental part of this series, there are a number of words in both the questions and the answers that have not been seen in the educational part; So, the system under test must dynamically consider a set of words to replace unseen words. In addition, because in the images of this collection, labels are used for writing and in the questions, to address different parts, labels are placed in each part of the address, after using the OCR[3] is necessary to process labels. The need to use different algorithms during the training and testing of a model on a specific set increases the number of models and this leads to additional difficulty and complexity [23].

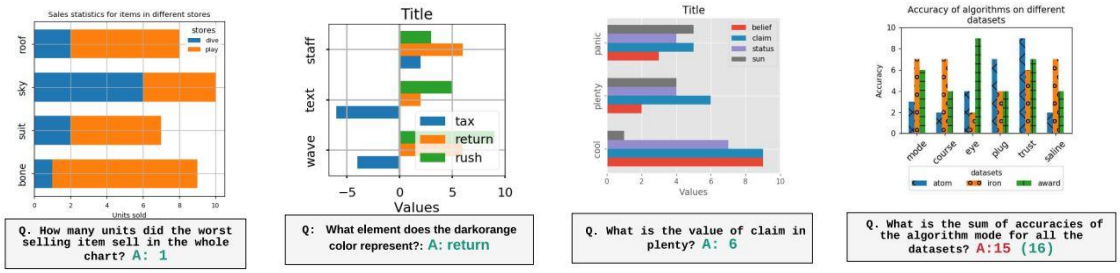


Figure 8 Sample images of DVQA dataset

3 - 4 SHAPES dataset:

This collection contains 244 artificial images of different types of geometric shapes (squares, triangles, circles, etc.) and 15,000 questions have been asked for these images, which are binary and the answers are in the range of yes and no. The subject in question is generally the location of shapes relative to each other, and since shapes are present in various colors, each shape is addressed by its

[2] Out-Of-Vocabulary
[3] Optical-Character- Recognition

color. Questions include 2 to 4 different features, a number of shape names, and relationships between shapes [2].

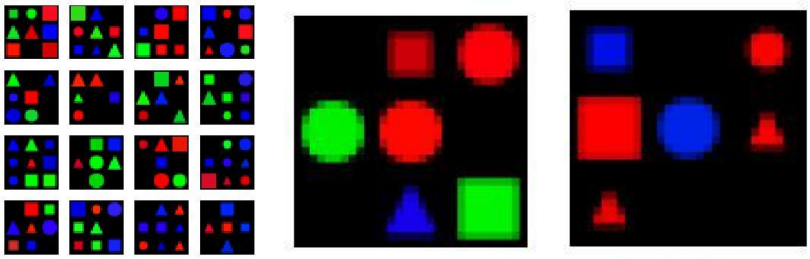


Figure 9 Sample images of SHAPES dataset

3 - 5 Diagrams dataset:

This collection contains 5,000 images of real cycles and diagrams that are drawn in connection with various topics in science lessons, such as: the water cycle and the digestive system. 15,000 questions have been asked for these charts, which are in the form of multiple options. The important point is that the OCR technique is needed to process the texts present in the graph images [2].

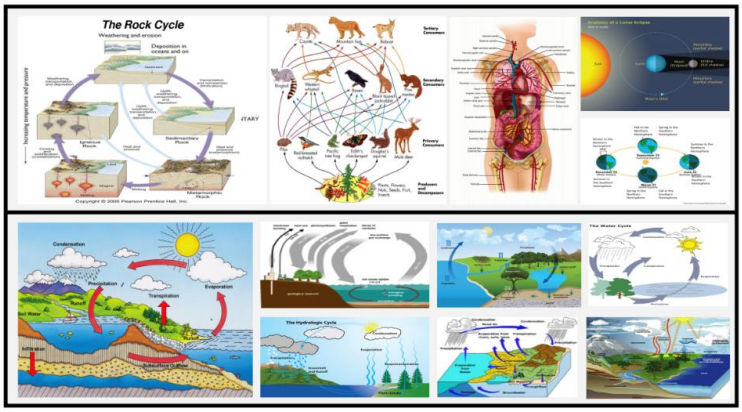


Figure10 Sample images of the Diagrams dataset

3 - 6 PlotQA dataset:

This dataset, which is prepared from real graph images, contains 224 thousand images. These images have been prepared from various sources such as: data available from the World Bank, data published by the government and global terrorism data. In this set of graphs, there are 3 types of columns (horizontal and vertical), linear and scatter graphs, which examine indicators such as: rainfall, coal production, fertility and pollution by carbon dioxide.

Since these images were real and collected from available sources, generating the related question was a big challenge. Due to the high cost of producing questions for all images, they selected only 1400 images with unique diagrams and assigned each image to 5 separate employees, resulting in 7,000 pairs of questions and answers.

The questions raised can be conceptually divided into 3 categories; Structural, data retrieval and inference. Structural questions are more about the overall structure of the graphs and detailed information is not needed to answer them. For example: How many columns of different colors are there in the chart? Data retrieval questions extract numbers related to concepts, such as: How many

tax payers were there in Myanmar in 2015? And in deductive questions for which there is no direct answer in the picture, more analysis and comparison is necessary, such as: Which country has the least number of endangered bird species [26]?

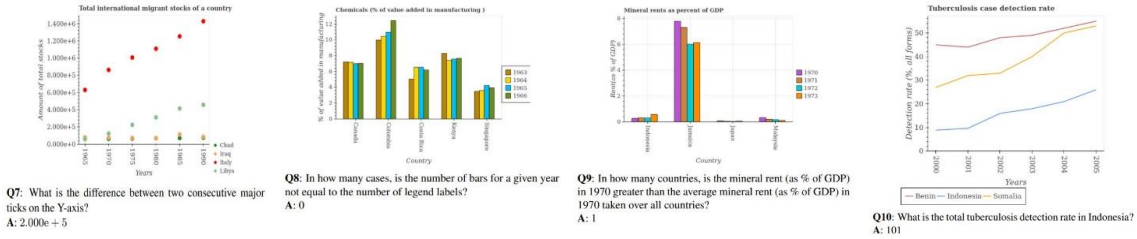


Figure 11 Sample images of the PlotQA dataset

3 - 7 Proposed dataset:

3 - 7 - 1 Introducing SBU-FQA dataset:

By examining the characteristics and types of graphs and shapes in the existing dataset, we decided to produce a set that was as diverse as possible in terms of graph types and question formats. In this regard, the dataset includes 6 categories of graphs. We created a horizontal column, a vertical column, a circle, a line, a dotted line, and a Gaussian. Among the charts mentioned, the Gaussian chart is present for the first time in an FQA dataset. Gaussian diagrams are widely used in statistics, to describe normal distributions, in signal processing, to define Gaussian filters, in image processing, to create Gaussian blurs, and in mathematics and engineering to solve heat equations and diffusion equations and many other applications are considered and the lack of presence of this type of graph in the images in the FQA dataset was felt. The SBU-FQA dataset is a dataset including 120,000 images of 6 types of graphs and more than 1300,000 questions related to these images. Currently, the images in this collection are equally divided between 6 types of graphs and there are 20,000 images of each graph in the collection, but the code developed to generate this dataset is written in a dynamic

way that does not include restrictions and from each graph type. It produces as many as the user needs. The questions in this series are binary questions and are answered with yes or no. These questions have been asked in 23 different formats, which are generally considered as deductive questions. In the following, we will check what topics are asked in the questions related to each type of diagram.

Table8 Items in question for each chart model

Chart model	Items in question
Vertical bar chart	1.The amount of a bar 2. Comparing the value of two bars
Horizontal bar chart	1 .The amount of a bar 2. Comparing the value of two bars
Linear graph	1. The amount of area under a line graph 2.Amount of data 3.Comparing the data value of two lines 4.The possibility of crossing two lines in the diagram 5. The slope of the lines
Dotted line diagram	1.The amount of area under a line graph 2.Amount of data .3Comparing the data value of two lines 4.The slope of the lines

Pie chart	1 .The amount of a sector 2. Comparing the value of two sectors
Gaussian diagram	1. Comparison of population charts 2.Average comparison 3. Comparison of standard deviation 4.Variance comparison

3 - 7 - 2 Gaussian diagrams:

As can be seen from Table 8, the Gaussian diagram contains many parameters, each of which can be questioned. In the image below, we can see the statistical characteristics of this graph;

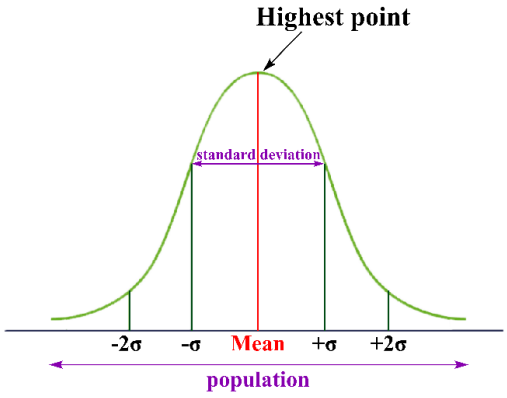


Figure 12 Gaussian distribution

If we consider the Gaussian diagram as a description of the normal distribution, this distribution includes: population, variance, standard deviation and mean. In the following, we define the mentioned statistical concepts:

Population: The result of subtraction of the largest sample from the smallest.

Standard deviation: Deviation means the distance from the average, and the standard means the standard of this value. The smaller the standard deviation of a set of samples, the closer those samples are to the mean.

Variance: The square of the standard deviation is called variance, which describes how the samples are scattered.

Middle: It is a sample in which 50% of the samples are located before it and 50% after it, and this sample itself is in the first 50%.

Average: the sum of all samples divided by the total number of samples.

Mode: The example that has the most repetition.

In the SBU-FQA dataset, all generated Gaussian charts are of normal type, when the Gaussian chart is normal, the three concepts of median, mean, and mode will be equal. In the following, we see the formula for drawing the Gaussian diagram;

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}}\, exp\left(-\frac{1}{2}\,\frac{(x-\mu)^2}{\sigma^2}\right) \tag{9}$$

In the formula above, x represents each sample, μ represents the average value and σ is the standard deviation, and as we explained, its square represents the variance value.

3 - 7 - 3 Generating images for SBU-FQA dataset:

As we said, this set is a synthetic dataset that is drawn based on the nature of each graph using its own points, lines, coordinate axes and guides. In drawing each diagram by randomly selecting numerical values and used colors, the effort was to produce diagrams closer to reality. The colors

used in each chart were selected from a set of 100 colors, and since there is more than one element in most of the images, many unique color combinations have been produced. Numerical ranges such as: maximum bar length, number of columns in a graph, maximum coordinates of the last point on lines, number of line elements in an image, coordinates of the center point of a circle, maximum length for circle radius, mean, standard deviation and number of Gaussian elements in the image was determined in such a way that we have the most diverse images. After determining the numerical ranges and the number of elements in each image, we started to generate graphs using Bokeh library functions in Python. Another important issue is the presence or absence of a diagram guide, which we considered to occur randomly with a 50% chance, and thus half of the produced images include a guide and the other half do not. This difference itself causes more challenges in the dataset. The background of all the images is white and its design is simple and checkered. For each image, the probability of its occurrence is random and with a 50% chance between the two states. In the Figure 13, you can see examples of images in the SBU-FQA collection.

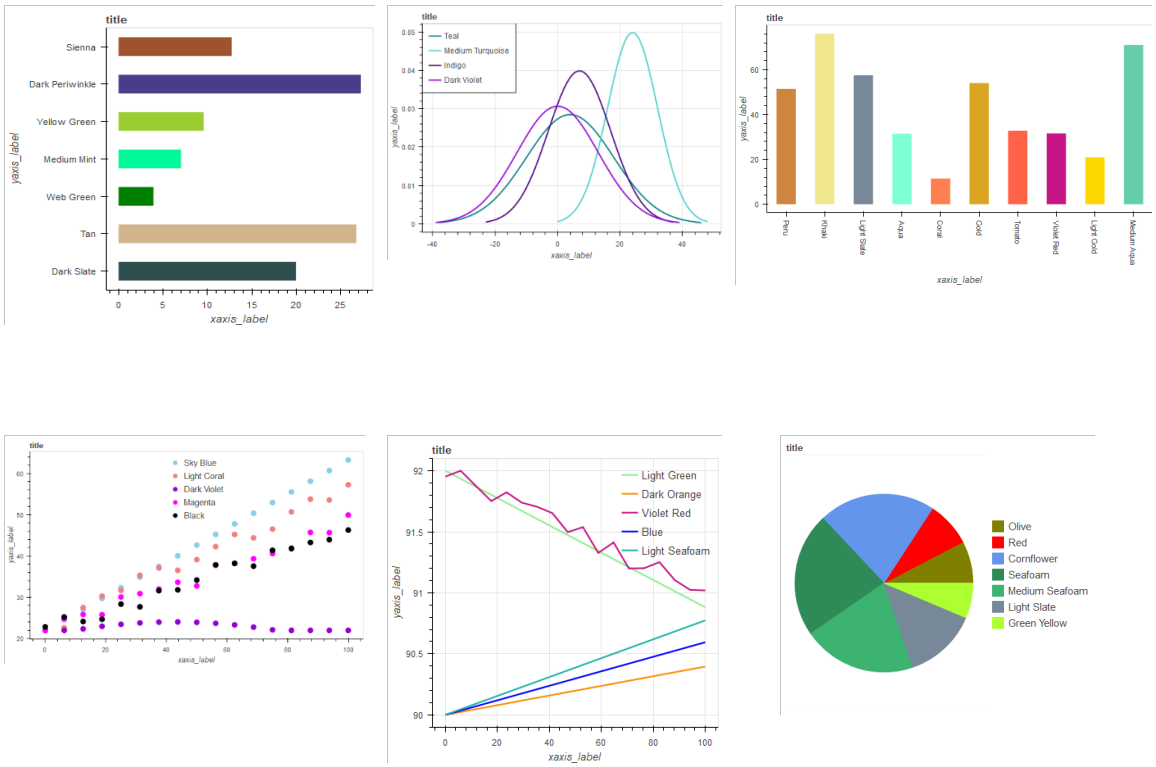


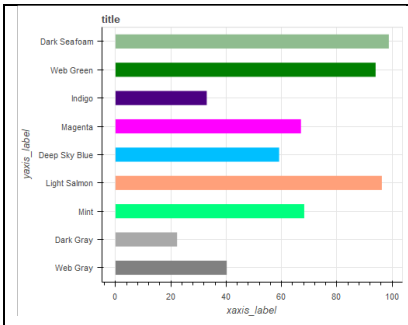
Figure 13 Sample images of the SBU-FQA dataset

3 - 7 - 4 Generating question-answer pairs for SBU-FQA dataset:

During the image generation process, the data related to the generation of each image, such as: the number of elements, the colors used, the value of each numerical parameter, etc., are stored separately. After generating the images, by specifying 23 specific formats for the form of questions and using the stored information for each image, several questions are generated. After generating the questions and saving the answer to each question, the act of filtering the questions is done based on their answers. The purpose of doing this is that in the entire set of SBU-FQA questions, the number of questions with a yes answer and the number of questions with a no answer are the same. This

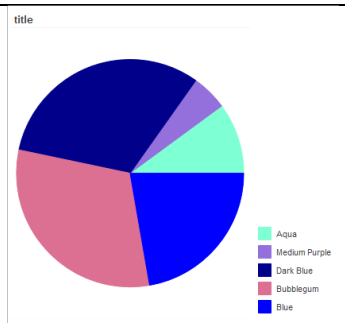
causes the same distribution of answers in the dataset and prevents bias towards a particular answer.

In the Figure 14, can be seen examples of questions and answers raised for different images.



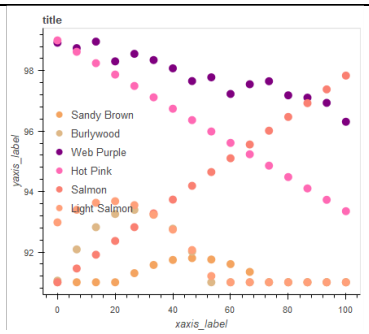
Q: Is Dark Gray greater than Dark Seafoam?

A: No



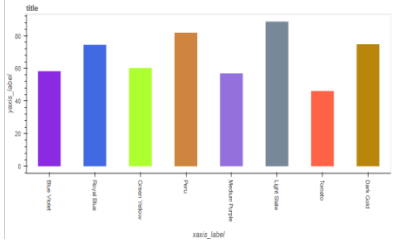
Q: Is Bubblegum less than Medium Purple?

A: No



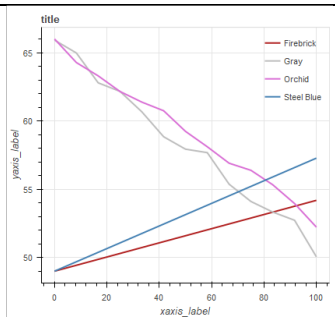
Q: Does Web Purple have the largest area under the curve?

A: Yes



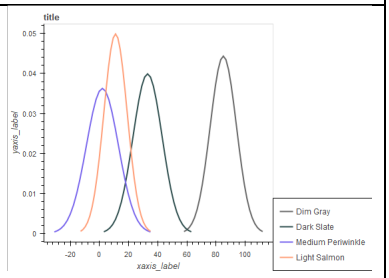
Q: Is Light Slate greater than Tomato?

A: No



Q: Does Gray have the lowest value?

A: Yes



Q: Does Light Salmon have the smallest variance?

A: Yes

Figure 14 Sample Question and Answers in the SBU-FQA dataset

Chapter 4

4. Proposed Method:

4 - 1 Architecture of ViTs

Transformers naturally dealt with word sequences since their primary training was in the area of natural language processing. But in computer vision, we only have one input image. The authors of the original ViTs paper suggested a solution in the form of splitting the input image into a number of fixed-sized image patches. Since sets are position-invariant, this did actually make it possible to employ the transformer architecture in computer vision, but critical positional information was lost. By incorporating a learnt positional encoding into the image patches, this issue was resolved.

4 - 2 ViT+LSTM method

The ViT architecture itself is quite straightforward and all the computations inside it can be summarized as:

From the input image, the first layer of ViTs extracts a predetermined number of patches. A special class token vector is then added to the series of embedding vectors after the patches are projected to linear embeddings. Then, vectors containing positional information are added to the embeddings and the class token and the sequence is passed into the transformer blocks. The final classification is produced by an MLP head using the class token vector that was retrieved from the last transformer block's output.

We used Visual Transformer in this model, ViT+LSTM, in order to extract visual features for the provided model, as described in [27].

A transformer is a deep learning model used in machine learning that uses attentional processes to weight the importance of each input data element differently. Transformers offer enormous potential as a general-purpose learning technique that can be used to a variety of data modalities, including recent developments in computer vision that have attained state-of-the-art standard accuracy while requiring fewer parameters.

We will first examine the overall architecture before going more in-depth on each of the following steps: splitting input images into patches, obtaining linear representations from each patch (referred to as Patch Embeddings), adding position embeddings, and classifying each Patch Embedding with the [cls] token.

Preprocessing the photos is necessary before creating patches. FQA photos come in a variety of sizes, but we reduce them to 224 × 224 so that image patching is easier. Making patches all over a three-channel (RGB) input image of a gaussian shape is the first step towards identifying it. With the first patch coming from the top-left and the last patch coming from the bottom-right of the source image, we are able to generate 14 ×14 or 196 patches. Last but not least, there are 196 patches in total, each measuring 3 ×16 ×16. The number of channels is represented by 3 in each patch (RGB). To get the Patch Embeddings matrix, we pass these patches through a trainable linear projection that is used in the main research by Dosovitskiy, A. et al. [28]. The patch Embeddings matrix of size 196 x 768 is acquired then.

The [cls] token, which is a vector with a size of 1 × 768, will be inserted in the following phase, and a matrix with position embeddings will be added to the patch embedding matrix, which has a size of 196 × 768. Both patch and position embeddings are combined to enter the transformer encoder, as can be seen in Figure 15.

When the image feature vector acquired, it is concatenated with the question's extracted features, the question encoding is generated using an LSTM network. The structure of this model is such that, first the input question is tokenized into tokens using glove [29], it puts a specific token for each word. Tokens will then enter the LSTM layer with size 128, followed by a dense layer.

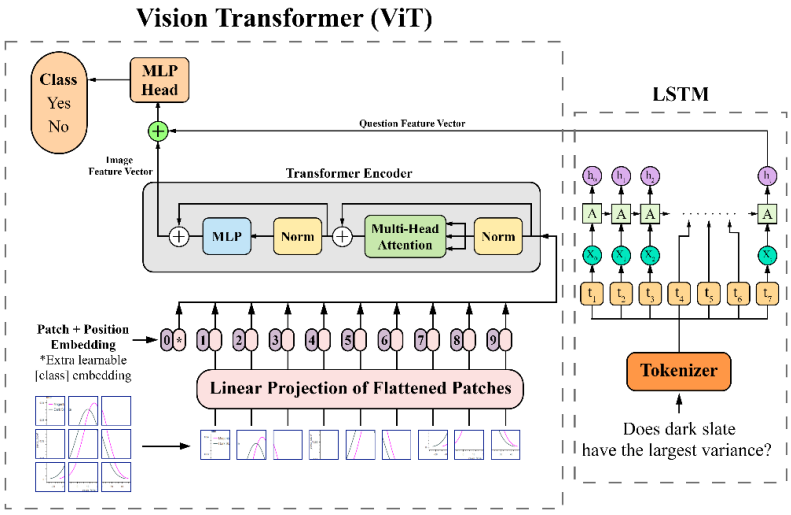


Figure 15 ViT+LSTM structure

Chapter 5

5. Result and discussion

This section serves as an evaluation of our suggested paradigm for FQA. We contrast our model with the prior State in order to evaluate it. We will also demonstrate how well our model works with each sort of figure. Each model's "performance" on the validation set of datasets is used to describe its accuracy. We also discuss how earlier techniques that produced promising results on simpler datasets fall short when dealing with more complex data, i.e., given a more complex dataset, earlier models aren't performing as well as they should, and despite their reported performance, more precise models like the one proposed in this research are needed. Although the proposed dataset can produce any number of images and question combinations, we created 20,000 images for this research's dataset. They are equally split into six main categories of diagrams: gaussian, pie, line, dot-line, vertical bar charts, and horizontal bar charts.

5 - 1 Experiments

The performance of the models trained for FQA is shown in Table.1. The accuracy obtained from the PReFIL[3] and the RN[10] model on SBU-FQA are 57 and 61, respectively. Our proposed model achieved a high accuracy of 74% on the dataset, improving previous results by more than 10%. This shows that the self-attention mechanism in the ViT[28] has significantly improved in comparison with both the RN[10] model and PReFIL[3]. Our model in nature is more similar to RN, in spirit that both models try to find correspondence between the question and image regions However, our model has the advantage of considering correspondence between image regions. As shown in Figure 15, attention layer in transformer encoder makes it feasible to globally embed information throughout the

entire image. In order to recreate the structure of the image, the model also learns from training data to encode the relative locations of the image patches.

Table 9 Performance of baseline models on FigureQA and SBU-FQA validation sets

models	SBU-FQA
Text-Only[10]	48.18
CNN+ LSTM[27]	53.70
RN[10]	57.08
PReFIL[3]	61.00
ViT+LSTM	74.30

Depending on the type of figures, each model's performance is displayed in Table.9. All prior models were outperformed by ViT+LSTM, which is indicated in each row of Table.2 along with the accuracy attained after applying the models to a particular type of figure. In general, our suggested model performs best in vertical and horizontal bars, with accuracies of 75 and 76, respectively; however, after comparing the various columns of Table.2, we find that the ViT+LSTM model has accomplished noteworthy accuracies despite the double complexity in Line, Dot Line, and Gaussian images.

Multiple models' deployment on the SBU-FQA dataset produced useful results, demonstrating that training models for the FQA job does not require enormous amounts of datasets and can be completed with a sufficient number of images with high diversity. Despite the fact that our dataset is smaller than earlier datasets (such as FigureQA and DVQA), the findings show that it is more complex. More so than the quantity of photographs, the diversity of the dataset's images has a substantial impact on

complexity. Because Gaussian images have a more complex structure than current images, baseline models that use the SBU-FQA are less accurate.

Table 10 The accuracy of baseline models per figure type on SBU-FQA validation set

Figure Type	Text-only[10]	CNN+LSTM[27]	RN[10]	PReFIL[3]	ViT+LSTM
Vertical Bar	48.53	53.05	59.12	63.84	75.30
Horizontal Bar	47.15	51.26	57.85	63.27	76.37
Line	47.42	50.93	56.39	60.05	73.15
Dot Line	46.73	52.50	56.03	60.79	74.75
Pie	49.12	54.86	58.49	61.52	74.88
Gaussian	49.71	54.20	55.36	58.18	74.15

Furthermore, as anticipated, due to variations in the features of SBU-FQA compared to prior figures, it was difficult for the models to comprehend the structure of Gaussian figures and to respond to pertinent inquiries. However, we made an effort to solve this issue utilizing the transformer structure, which applied the question embeddings using an LSTM and extracted positional embeddings using a 2D convolutional network.

The structure of the Gaussian diagram was such that we were able to make questions for each Gaussian image in different templates. As a result, the dataset we developed had 8 question templates for gaussian figures. The accuracy of models for different question templates is shown in Table 11. Gaussian question templates are contained in the first eight rows. As can be seen from the table, ViT+LSTM performs much better on questions of this nature than the other models, with results on questions in lines 1, 2, 9, 10, 13, 14, 19, and 20 being more accurate than those of the other models.

All models, including ours, were unable to answer the challenging questions in lines 5, 6, 7, 8, 15, 16, and 23, and our model was no different. It's crucial to remember that the accuracy found in questions about Gaussian figures is typically lower than the accuracy found in other questions. Not surprisingly, given that a Gaussian distribution's characteristics have extremely nonlinear impacts on the distribution's form.

Table 11 The accuracy of baseline models per question type on SBU-FQA validation set

Templates	Text_Only[10]	CNN+LSTM[27]	RN[10]	PReFIL[3]	ViT+LSTM
Does X have the smallest population?	49.25	55.40	58.53	63.18	77.47
Does X have the largest population?	49.62	56.27	59.61	64.22	78.35
Does X have the smallest mean value?	47.13	54.12	57.82	59.70	70.68
Does X have the largest mean value?	48.35	54.15	58.73	60.24	72.10
Does X have the smallest variance?	46.88	51.31	55.72	57.32	70.29
Does X have the largest variance?	45.75	51.19	56.09	57.05	70.00
Does X have the smallest standard deviation?	46.20	52.34	54.11	58.30	70.42
Does X have the largest standard deviation?	46.20	51.04	55.45	58.42	69.71
Is X the minimum?	50.76	56.42	60.13	64.75	76.52
Is X the maximum?	50.38	57.75	59.40	63.50	76.00

Is X the low median?	48.41	53.74	59.39	60.37	70.59
Is X the high median?	48.76	54.00	57.30	61.11	73.25
Is X less than Y?	50.64	55.10	60.27	64.62	76.29
Is X greater than Y?	50.15	56.23	60.75	64.15	76.88
Does X have the minimum area under the curve?	45.65	51.50	54.05	58.36	69.26
Does X have the maximum area under the curve?	45.62	51.08	55.39	57.22	70.85
Is X the smoothers?	48.70	53.60	56.20	59.57	73.13
Is X the roughest?	47.86	55.18	57.24	60.43	73.24
Does X have the lowest value?	48.32	56.05	58.15	63.56	77.38
Does X have the highest value?	49.47	57.22	59.08	63.48	75.16
Is X less than Y?	48.38	53.40	57.31	59.10	72.67
Is X greater than Y?	47.16	54.31	58.65	60.41	71.55
Does X intersect Y?	46.26	50.17	54.25	57.70	69.38

Chapter 6

6. Conclusion

An FQA model built on the ViT architecture was suggested in this study. With the consideration of the relationship with other regions, this method offers rich picture attributes to describe each image region. Local features were then combined with LSTM network question embeddings to provide significant results on our dataset. The production of Patch Embeddings, which considers the interaction between different patches of the image while fusing with the main embedding, underlies the higher performance of the ViT+LSTM model over earlier models. The outcomes from this method were 10% better than the earlier ones. Additionally, we introduced a sizable FQA dataset in terms of information diversity, investigated the effects of a figure dataset on a FQA system, and offered recommendations for creating a suitable FQA model in accordance with the existing data.

7. References

[1] R. Li and J. Jia, "Visual question answering with question representation update (qru)," *Advances in Neural Information Processing Systems,* vol. 29, 2016.

[2] Q. Wu, D. Teney, P. Wang, C. Shen, A. Dick, and A. Van Den Hengel, "Visual question answering: A survey of methods and datasets," *Computer Vision and Image Understanding,* vol. 163, pp. 21-40, 2017.

[3] K. Kafle, R. Shrestha, S. Cohen, B. Price, and C. Kanan, "Answering questions about data visualizations using efficient bimodal fusion," in *Proceedings of the IEEE/CVF Winter conference on applications of computer vision*, 2020, pp. 1498-1507.

[4] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, 2015: PMLR, pp. 2048-2057.

[5] S. Barra, C. Bisogni, M. De Marsico, and S. Ricciardi, "Visual question answering: Which investigated applications?," *Pattern Recognition Letters,* vol. 151, pp. 325-331, 2021.

[6] D. Zhang, R. Cao, and S. Wu, "Information fusion in visual question answering: A survey," *Information Fusion,* vol. 52, pp. 268-280, 2019.

[7] S. Manmadhan and B. C. Kovoor, "Visual question answering: a state-of-the-art review," *Artificial Intelligence Review,* vol. 53, no. 8, pp. 5705-5745, 2020.

[8] K. Saito, A. Shin, Y. Ushiku, and T. Harada, "Dualnet: Domain-invariant network for visual question answering," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, 2017: IEEE, pp. 829-834.

[9] K. J. Shih, S. Singh, and D. Hoiem, "Where to look: Focus regions for visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4613-4621.

[10] S. E. Kahou, V. Michalski, A. Atkinson, Á. Kádár, A. Trischler, and Y. Bengio, "Figureqa: An annotated figure dataset for visual reasoning," *arXiv preprint arXiv:1710.07300,* 2017.

[11] S. Antol *et al.*, "Vqa: Visual question answering," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2425-2433.

[12] G. A. Miller and W. G. Charles, "Contextual correlates of semantic similarity," *Language and cognitive processes,* vol. 6, no. 1, pp. 1-28, 1991.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781,* 2013.

[14] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems,* vol. 26, 2013.

[15] J. L. Elman, "Finding structure in time," *Cognitive science,* vol. 14, no. 2, pp. 179-211, 1990.

[16] M. Malinowski, M. Rohrbach, and M. Fritz, "Ask your neurons: A deep learning approach to visual question answering," *International Journal of Computer Vision,* vol. 125, no. 1, pp. 110-135, 2017.

[17] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, "Multimodal compact bilinear pooling for visual question answering and visual grounding," *arXiv preprint arXiv:1606.01847,* 2016.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems,* vol. 28, 2015.

[19] P. Anderson *et al.*, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6077-6086.

[20] D. Teney, P. Anderson, X. He, and A. Van Den Hengel, "Tips and tricks for visual question answering: Learnings from the 2017 challenge," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4223-4232.

[21] V. Marois, T. Jayram, V. Albouy, T. Kornuta, Y. Bouhadjar, and A. S. Ozcan, "On transfer learning using a MAC model variant," *arXiv preprint arXiv:1811.06529,* 2018.

[22] J. Johnson, B. Hariharan, L. Van Der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2901-2910.

[23] K. Kafle, B. Price, S. Cohen, and C. Kanan, "Dvqa: Understanding data visualizations via question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5648-5656.

[24] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Neural module networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 39-48.

[25] A. Kembhavi, M. Salvato, E. Kolve, M. Seo, H. Hajishirzi, and A. Farhadi, "A diagram is worth a dozen images," in *European conference on computer vision*, 2016: Springer, pp. 235-251.

[26] N. Methani, P. Ganguly, M. M. Khapra, and P. Kumar, "Plotqa: Reasoning over scientific plots," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 1527-1536.

[27] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko, "Learning to reason: End-to-end module networks for visual question answering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 804-813.

[28] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929,* 2020.

[29] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532-1543.