Visual Question Answering – Implementation using Keras

Devangi P. Bhuva[1,] Riddhi N. Nisar[2], Prof. Pramila M. Chawan[3]

[1]B.Tech Student, Dept of Computer Engineering and IT, VJTI College, Mumbai, Maharashtra, India
[2]B. Tech Student, Dept. of Computer Engineering and IT, VJTI College, Mumbai, Maharashtra, India
[3]Associate Professor, Dept. of Computer Engineering and IT, VJTI College, Mumbai, Maharashtra, India

---***---

Abstract – *This project aims to perform the task of Visual Question Answering using Keras deep learning frameworks that combine the image features and the question vector to produce answers for the questions posed by making use of a pre-processed image dataset along with spaCy word embeddings. In this article, we have explained how we can merge the CNN and RNN models with a dense multi-layer perceptron to produce categorical classes that correspond to our answer. The underlying problem here is to merge the two different models, as these are two different domains of the machine-learning regime. We have solved it by merging the image vector and question vector in a multi-layer perceptron with the number of layers and class of activation mentioned further in the article. In the implementation, we use the Keras python package with the backend of Tensorflow and the pre-processed VGG-16 weights for extracting image features using CNN and the question vector using the spaCy word embeddings, and finally we use the multi-layer perceptron to combine the results from the image and question.*

Key Words: convolutional neural network, recurrent neural networks, word vector, Keras, statistical bias, and multilayer perceptron

1. INTRODUCTION

The issue of solving visual question answering goes past the ordinary issues of image captioning and natural language processing, as it is a blend of both the strategies which makes it a perplexing system. Since language has a complex compositional structure, the issue of taking care of vision and language becomes a tough task.

It is very simple to get a decent superficial exhibition of accuracy when the model disregards the visual content on account of the predisposition that is available in the dataset. For this situation, the model doesn't genuinely comprehend the information embedded in the picture and just focuses on the language semantics, which is not what we need. For example, in the VQA dataset, the most widely recognized game answer "cricket" is the right response for 41% of the inquiries beginning with "What game is", and "white" is the right response for half of the inquiries beginning with "What shading is". These dialects can make a bogus impression of accomplishing precision. It is very conceivable to get cutting edge results with a moderately low comprehension of the picture. This should be possible by misusing the factual inclinations as well, which are present in the datasets. They are commonly obvious in standard language models as well. Presently, we need language to posture difficulties including the visual comprehension of rich semantics. The frameworks should not have the option to get rid of ignoring the visual data.

In this work, we implement a visual question answering system aiming to use a balanced bias free dataset constructed specially to counter these language biases and make the role of image understanding in VQA more impactful by utilizing the underlying image features and the corresponding semantics of language.

2. LITERATURE REVIEW

2.1 Convolutional Neural Networks – VGG-16

In neural networks, convolutional neural network (ConvNets or CNNs) are one of the major categories to do the recognition and classification of various types of images. Convolutional neural networks are now capable of outperforming humans on some computer vision tasks, such as classifying images. Imagenet was a research project to develop a large database of images with annotations, e.g. images and their descriptions. For the classification task, images must be classified into one of 1,000 different categories.

Researchers from the Oxford Visual Geometry Group, or VGG for short, released two different CNN models, specifically a 16-layer model and a 19-layer model. In our implementation, we use the pre-trained weights of the VGG-16 model for creating the image vector.

2.2 Word embedding using spaCy:

In our implementation, we use a pre-existing model for the word embeddings. With word embeddings, we were able to capture the context of the word and then find semantic and syntactic similarities. We have used the spaCy word embedding model. While spaCy was only recently developed, the algorithm already has a reputation for being the fastest word embedding in the world.

2.3 Multilayer Perceptron (MLP)

A multilayer perceptron (MLP) is portrayed fundamentally as a neural system which is profound, which has shrouded layers present in it. As intended by the name, it has various perceptions. A multilayer perceptron contains info layer, managed to transfer the essential information highlights and a yield layer, which give out the conclusive outcome of the counterfeit neural system.

3. PROPOSED SYSTEM

3.1 Problem statement

“To find the correct answer, in an implementation, to a question posed based on an image using the technique of combination of language and vision using VGG-16 pre-trained weights, spaCy word embedding and a multi-layer perceptron combining the two with the Keras framework”

3.2 Problem Elaboration

This has been observed that finding the answers to questions based on an image correctly without inherent statistical bias on the dataset is a bit difficult. This leads to answers based on the dataset bias, which give quite accurate results, but without considering the features of the image.

To carry out the process of finding the correct answer to a question posed based on an image, we implement a python script using Keras, that encodes the question and image into vectors and then concatenates the two using MLP. We use a balanced bias free dataset so that the inherent statistical biases leading to constant answers can be overcome.

3.3 Proposed Methodology

There are different methods in the language+vision domain to find the answer to the question posed based on the input image. But each of the methodologies has their pros and cons. To work effectively with the proposed framework and after an exhaustive comprehension of the given writing review, the proposed approach appears to be a reasonable fit for accomplishing best in class exactness.

The following steps are proposed:

1. The first step will be the word transformation. For the question, we will convert each word to its word vector, and for that we will use the spaCy word embeddings model.

2. Coming to the image, it is sent through a Deep Convolutional Neural Network (from the well-known VGG Architecture), and the image features are extracted from the activation of the second last layer (that is, the layer before the softmax function).

3. To combine the features from the image and the word vector, we use a multilayer perceptron consisting of fully connected layers. The layers are mentioned further in the article. We get a probability distribution over all the possible outputs. The output with the highest probability is our answer to the question posed based on the image.

3.4 Proposed System Architecture

The proposed system workflow is as given as shown in diagram:

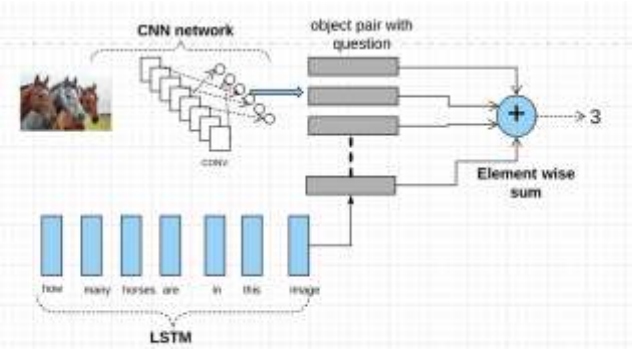


Fig - 1: Workflow

The above block diagram illustrates the architecture that we propose for the Visual Question Answering system. Here we import the **Keras** library to create a Convolutional Network layer for particular image and extract the required image features. spaCy word embeddings are used as a part of RNN for natural language processing to convert the question into word vector, understanding its semantics. We merge the extracted image features and word vector and using MLP we create an (object, question) pair.

4. IMPLEMENTATION

4.1 Dataset

Libraries: Flask, os, werkzeug_utils.

Dataset: Natural images

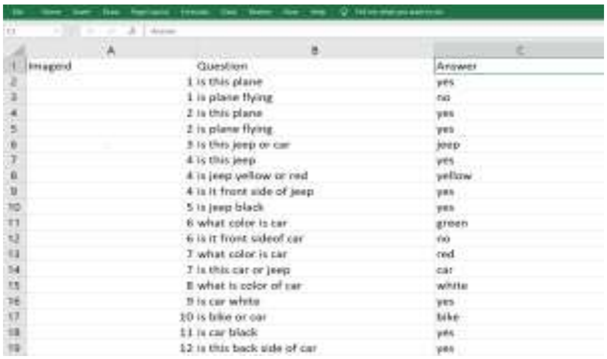
List of the steps required getting the dataset:

1. Download and collect set of natural images from various repositories.

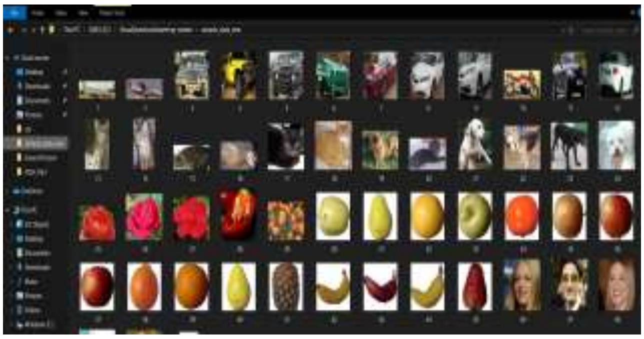
2. Make sure images have same extension (.jpg or .png)

3. Create training.csv for all types of natural images you want to select (E.g.: Cars, Fruits, Flowers, etc.)

4. Attributes such as IMG_ID, Question and answer mentioned in excel or .CSV file.



	A	B	C
1	ImageId	Question	Answer
2		1 is this plane	yes
3		1 is plane flying	no
4		2 is this plane	yes
5		2 is plane flying	yes
6		3 is this jeep or car	jeep
7		4 is this jeep	yes
8		4 is jeep yellow or red	yellow
9		4 is it front side of jeep	yes
10		5 is jeep black	yes
11		6 what color is car	green
12		6 is it front sideof car	no
13		7 what color is car	red
14		7 is this car or jeep	car
15		8 what is color of car	white
16		9 is car white	yes
17		10 is bike or car	bike
18		11 is car black	yes
19		12 is this back side of car	yes



4.2 Algorithm

Libraries: Keras=2.1.0, Tensorflow=1.15, OS, sklearn, cv2, spaCy, numpy, vgg16

Algorithm:

1. Take pre-processed VGG-16 dataset weights, which categorize objects into 1000 categories. It has 16 layers and we pop the last 2 layers for use in our image object classification.
2. The image is converted to a corresponding (1, 4096) dimension vector by the VGG-16 Model
3. We use the spaCy dataset for word-embeddings of our question tokens and convert it to a question tensor.
4. We have {{22}} trainY labels, so we convert it to categorical variables using to_categorical function.
5. Our final model has the following layers:
6. We train the data by creating image features and question features by passing it through model.fit() function.
7. Later, we save the model architecture in a JSON file and the model weights in a H5 file.
8. In our working application, we load the architecture and weights using these saved files, and input image and corresponding question to get result.
9. The result is probabilistic values of the categories and we output the category with the maximum probability.

4.3 Implementation

VGG16

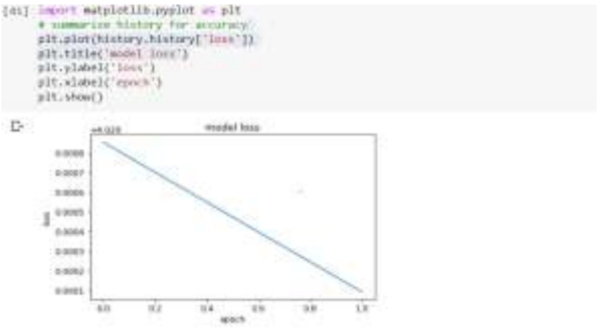
VGG16 is a convolution neural net (CNN) architecture. It is considered to be one of the excellent vision model architecture till date. Most unique thing about VGG16 is that instead of having a large number of hyper-parameter they focused on having convolution layers of 3x3 filter with a stride 1 and always used same padding and maxpool layer of 2x2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. In the end it has 2 FC (fully connected layers) followed by a softmax for output. The 16 in VGG16 refers to it has 16 layers that have weights. This network is a pretty large network and it has about 138 million (approx.) parameters.

SpaCy

SpaCy is an open-source software library for advanced natural language processing, written in the programming languages Python and Cython. It has the facilities for pre-trained word vector generation and integration with deep learning networks. SpaCy supports deep learning workflows that allow connecting statistical models trained by machine learning libraries like TensorFlow, Keras, Scikit-learn or PyTorch, few of which we are using in our implementation.

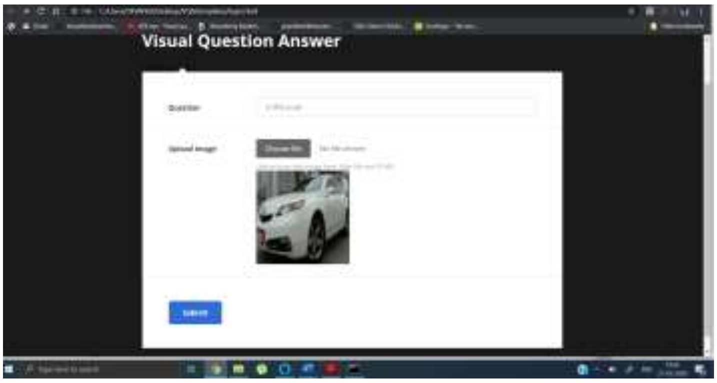
Screenshots:

Error rate:

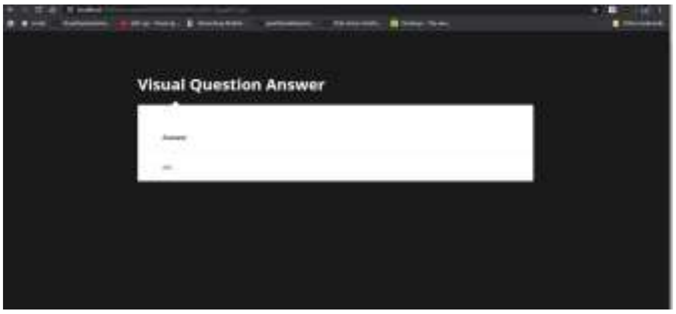


```
[41] import matplotlib.pyplot as plt
     # summarize history for accuracy
     plt.plot(history.history['loss'])
     plt.title('model loss')
     plt.ylabel('loss')
     plt.xlabel('epoch')
     plt.show()
```

Ask Question with selected image



Answer using trained model:



5. CONCLUSION

In this paper, we have implemented methodologies for visual question answering using the pre-processed VGG-16 weights for image feature extraction and the spaCy word embeddings for question vector creation. We have implemented the techniques using Python Keras framework using Tensorflow as the backend. For the application part of the project, we have used the light-weight Flask web-framework. The validation set of the dataset exhibits a peak accuracy of 94 percent. This is better than the traditional models not using the convolutional and recurrent neural network deep learning techniques. However, there is a good amount of future scope to improve the accuracy on real-time data, using larger datasets and a wide range of questions along with attention modelling.

REFERENCES

[1] Yin and Yang: Balancing and Answering Binary Visual Questions (CVPR 2016)

[2] VQA: Visual Question Answering by Aishwarya Agrawal , Jiasen Lu , Stanislaw Antol ,Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, Devi Parikh

[3] Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, Devi Parikh, Virginia Tech, Army Research Laboratory, Georgia Institute of Technology.

[4]https://towardsdatascience.com/deep-learning-and-visual-question-answering-c8c8093941bc

[5]https://visualqa.org/

[6] https://tryolabs.com/blog/2018/03/01/introduction-to-visual-question-answering/

[7] Image Captioning and Visual Question Answering Based on Attributes and External Knowledge Qi Wu, Chunhua Shen, Peng Wang, Anthony Dick, Anton van den Henge

[8] Learning to Reason: End-to-End Module Networks for Visual Question Answering ICCV 2017 Ronghang Hu Jacob Andreas Marcus Rohrbach Trevor Darrell Kate Saenko

[9] Graph-Structured Representations for Visual Question Answering CVPR 2017 Damien Teney, Lingqiao Liu, Anton van den Hengel

AUTHOR'S PROFILES

Devangi P. Bhuva, Final Year B. Tech Student, Department of Computer Engineering and IT, VJTI, Mumbai, Maharashtra, India.

Riddhi N. Nisar, Final Year B. Tech Student, Department of Computer Engineering and IT, VJTI, Mumbai, Maharashtra, India.

Prof. Pramila M. Chawan is working as an Associate Professor in the Computer Engineering Department of VJTI, Mumbai. She has done her B. E. (Computer Engineering) and M.E (Computer Engineering) from VJTI COE, Mumbai University. She has 27 years of teaching experience and has guided 75+ M. Tech. projects and 100+ B. Tech. projects. She has published 99 papers in the International Journals, 21 papers in the National and International conferences/symposiums. She has worked as an Organizing Committee member for 13 International Conferences, one National Conference and 4 AICTE workshops. She has worked as NBA coordinator of Computer Engineering Department of VJTI for 5 years. She had written proposal for VJTI under TEQIP-I in June 2004 for creating Central Computing Facility at VJTI. Rs. Eight Crore (Rs. 8,00,00,000/-) were sanctioned by the World Bank on this proposal.