

---

ETD Archive

---

Winter 1-1-2019

## **An Attention Based Deep Neural Network For Visual Question Answering System**

Labhesh Popli  
*Cleveland State University*

Follow this and additional works at: <https://engagedscholarship.csuohio.edu/etdarchive>  
**How does access to this work benefit you? Let us know!**

---

### **Recommended Citation**

Popli, Labhesh, "An Attention Based Deep Neural Network For Visual Question Answering System" (2019).  
*ETD Archive*. 1301.  
<https://engagedscholarship.csuohio.edu/etdarchive/1301>

This Dissertation is brought to you for free and open access by EngagedScholarship@CSU. It has been accepted for inclusion in ETD Archive by an authorized administrator of EngagedScholarship@CSU. For more information, please contact [library.es@csuohio.edu](mailto:library.es@csuohio.edu).

AN ATTENTION BASED DEEP NEURAL NETWORK FOR VISUAL QUESTION  
ANSWERING SYSTEM

LABHESH POPLI

Bachelor of Technology in Electronics and Communication

Punjab Technical University

June 2007

Submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE IN SOFTWARE ENGINEERING

at the

CLEVELAND STATE UNIVERSITY

December 2019

We hereby approve this thesis for

LABHESH POPLI

Candidate for the Master of Science in Software Engineering degree for the

Department of Electrical Engineering and Computer Science

and

CLEVELAND STATE UNIVERSITY'S

College of Graduate Studies by

---

Committee Chairperson, Dr. Sunnie Chung

---

Department & Date

---

Committee Member, Dr. Yongjian Fu

---

Department & Date

---

Committee Member, Dr. Wenbing Zhao

---

Department & Date

Student's Date of Defense: December 5, 2019

## **ACKNOWLEDGEMENTS**

I would like to sincerely thank Dr. Sunnie Chung for her continued guidance, expertise and advice throughout the year. This wouldn't have been possible without her in-depth knowledge on the subject. I would also like to thank my committee members Dr. Wenbing Zhao and Dr. Yongjian Fu, for their understanding and providing me all necessary support during my thesis research.

AN ATTENTION BASED DEEP NEURAL NETWORK FOR VISUAL QUESTION  
ANSWERING SYSTEM

LABHESH POPLI

**ABSTRACT**

With advances of internet computing and a great success of social media websites, internet is exploded with a huge number of digital images. Nowadays searching appropriate images directly through search engines and the web is trending. However, automatically finding images relevant to a textual query content remains a very challenging task. Visual Question Answering (VQA) system has emerged as a significant multidisciplinary research problem. The research combines methodologies from the different areas like natural language processing, image recognition and knowledge representation.

The main challenges for developing such a VQA system is to deal with the scalability of the solution and handling features of the objects in vision and questions in a natural language simultaneously. Prior works have been done to develop models for VQA by extracting and combining image features using Convolution Neural Network (CNN) and textual features using Recurrent Neural Network (RNN). This thesis explores methodologies to build a Visual Question Answering (VQA) system that can automatically identify and answer a question about the image presented to it. The VQA system uses methods of deep Residual Network (ResNet), an advanced Convolution Neural Network (CNN) model for image identification, and Long Short-Term Memory (LSTM) networks, which is advanced form of Recurring Neural Network (RNN) for Natural Language Processing (NLP) to analyze a user-provided question. Finally, the features from both an

image and a user question are combined to indicate an attention area to focus on to identify objects in the area of the image in deep residual network, to produce an answer in text.

When evaluated on the well-known challenging COCO data set and VQA 1.0 dataset, this system has produced an accuracy of 59%, with a 12% increase when compared with a baseline model without the attention-based technique and the results also show comparable performance to other existing state-of-the-art attention-based approaches in the literature. The quality and the accuracy of the method used in this research are compared and analyzed.

## TABLE OF CONTENTS

	Page
ABSTRACT.....	iv
LIST OF TABLES.....	viii
LIST OF FIGURES.....	ix
CHAPTER	
I. INTRODUCTION.....	1
1.1 Background.....	1
1.2 Motivation.....	3
1.3 Problem Statement.....	4
1.4 Overview of the Framework.....	5
II. LITERATURE REVIEW AND RELATED WORK.....	9
2.1 Literature Review.....	9
III. INTRODUCTION TO CONVOLUTION NEURAL NETWORKS.....	12
3.1 Convolution Layer.....	14
3.2 Non-Linearity: (ReLU).....	15
3.3 Pooling or Sub-sampling.....	17
3.4 Fully-Connected Layer.....	18
3.5 Softmax Function.....	19
3.6 Cost Function.....	20
3.7 Batch Normalization.....	21
3.8 Dropout.....	21
IV. INTRODUCTION TO RECURRENT NEURAL NETWORKS.....	24
4.1 Back Propagation Through Time (BPTT).....	26

4.2	The Problem of Long-Term Dependencies & Vanishing Gradient.....	27
4.3	Long Short-Term Memory (LSTM) networks.....	27
V.	TECHNICAL APPROACH AND METHODOLOGY.....	30
5.1	Architecture.....	30
5.1.1	Image Model .....	32
5.1.2	Question Model.....	32
5.1.3	Stacked Attention.....	32
5.2	Feature Extraction.....	34
5.2.1	Image Feature – ResNet.....	34
5.2.1.1	Residual Network.....	36
5.3	Comparing ResNet with AlexNet and VGG architectures.....	39
VI.	EXPERIMENTAL RESULTS AND EVALUATION.....	44
6.1	Datasets used for The Framework.....	44
6.2	Evaluation Metrics.....	45
6.3	Data Preprocessing.....	47
6.4	Model Training.....	47
6.5	Training Result.....	48
VII.	CONCLUSION AND FUTURE DIRECTION.....	53
7.1	Conclusion.....	53
7.2	Future Work.....	54
	BIBLIOGRAPHY.....	55
	APPENDIX.....	62



## LIST OF TABLES

Table	Page
I. Number of questions and images in VQA dataset .....	45
II. Percentage Accuracy for The Framework.....	48
III. Comparing accuracy with other ResNet+LSTM+attention models.....	50
IV. Comparing percentage accuracy of The Framework with prior work.....	51

## LIST OF FIGURES

Figure	Page
1. Information flow and models connectivity of The Framework.....	5
2. CNN architecture.....	6
3. LSTM with attention architecture.....	7
4. Convolution Neural Network Visualization.....	13
5. A simple ConvNet .....	13
6. Convolution layer.....	14
7. ReLU function.....	16
8. ReLU operation.....	16
9. Max pooling.....	17
10. Pooling.....	18
11. Recurrent Neural Network structure.....	25
12. Recurrent Neural Network representation.....	25
13. Equation for Recurrent Neural Networks.....	25
14. Long Short-Term Memory (LSTM).....	28
15. Model architecture and visualization.....	31
16. Plain layers and residual block in ResNet.....	36
17. ResNet architecture 1 .....	37
18. Conv1- Convolution.....	37
19. ResNet architecture 2 .....	38
20. Layer 1 .....	39
21. ILSVRC challenge error rates 2010-15.....	42

22.	Computational complexity of CNN models .....	43
23.	Comparison of different evaluation metrics for VQA .....	46
24.	Distribution of the length of the questions from the training set.....	47
25.	Comparison for ResNet + LSTM (with and without attention).....	48
26.	Comparison of The Framework to prior work done with attention model.....	50
27.	Comparing accuracy of The Framework with other models.....	52

# **CHAPTER I**

## **INTRODUCTION**

### **1.1. Background**

With advances of internet computing and a great success of social media websites, internet is exploded with a huge number of digital images. Nowadays searching appropriate images directly through search engines and the web is trending. The most common search engines today such as Google, Yahoo or Bing offer image search in a natural language query. However, the challenging task of automating the image retrieval relevant to a textual query content remains prevalent. Google image search engine which is used by numerous people around the world is reported to have a precision of only 39% [38].

Visual Question Answering (VQA) [2, 3, 4, 5, 8] has emerged as a significant multidisciplinary research problem and employs methods from areas such as image recognition, natural language processing and knowledge representation in both academia and industry. The machine needs to interpret both images and the questions to correctly answer any given questions about an image. Recently, visual attention-based models [41, 42, 43, 44] have shown great prospect for VQA, where the attention mechanism highlights

the image regions relevant to answers of the question by producing a spatial map manually [39].

As deep learning advances, VQA employs methods from image recognition, natural language processing and knowledge representation to obtain a better solution [30]. However, integrating the domain knowledges from multiple areas like object identification, object classification and text processing is a challenging AI task. For example, for a given image and a question “how many apples are in the basket?” the challenge is to recognize the “basket” in the image along with detecting and classifying the object “apples”. When such an integrated model is presented with an input image and a textual question, the model generates back a textual answer in the form of natural language [20].

Most of literature of attention models for visual question answering systems talk about the problem of identifying and focusing attention on the image. In this study, the problem and the importance of identifying “which words to listen to” or the attention on the question text is stressed upon. Consider the questions “how many horses are in this image?” and “how many horses can you see in this image?”. The same meaning from these two sentences are being captured from the first three words itself. Motivated by this observation, this thesis studies approaches to address the problem of question attention as well as reasoning about visual attention.

One major approach in research in deep learning for QA system per given input image is the use of attention based on the semantic representation of a question as query. The attention-based method searches for the regions in an image that are closely related to the answer instead of searching the whole image. When an image is searched focusing only on the relevant regions of the image with respect the questions to generate answers

progressively, the results of the attention-based approaches are proven to outperform any previous state-of-the-art approaches without attention [4].

This thesis follows a similar approach to develop an attention-based image question answering system with an unsupervised attention generated by a user-provided textual question related to the image. The system extracts feature from images using a deep Convolutional Neural Network (CNN), combines the features from the question analyzer in Natural Language Processing (NLP) using Long Short-Term Memory (LSTM) networks, which are a special kind of Recurrent Neural Network (RNN) explicitly designed to remember information for long periods of time. This eventually generates an attention model to produce answers based on the inputs with improved accuracy. The following sections explain a deep learning architecture with neural networks in detail, how they are trained and how a deep convolutional neural network is so remarkably good at understanding images.

## **1.2. Motivation**

Visual question answering (VQA) and image captioning present the researchers with high complexity compared to traditional machine learning tasks, such as classification or segmentation since the task of generating an answer for a textual question related to the image is not straightforward, it introduces a number of new challenges [34].

One of the challenging tasks begins with generating data sets to train and test. These efforts include crowdsourcing generated curated datasets. A significant motivation for crafting good benchmark dataset is that the interpretation and evaluation of the generated answer becomes difficult as problem and the scope of the task grows. Establishing and evaluating methodology that assigns scores also becomes more complex. As people have

a greater access to social media, the human answers in different scenarios are becoming more inconsistent [35].

Malinowski and Fritz [36] identified three different set of challenges to deal with in VQA tasks. The first deals with scalability of the solution, dealing with inherent concept ambiguity, and handling attributes of the objects in vision and language. The second is how to use the commonsense knowledge in question answering. To define and craft a benchmark dataset and quantifying the performance of different methods are the third challenge [34].

There are many applications for VQA systems which can inspire the researchers to establish more advance systems to make the results better. Some of the examples are the disability aid systems along with being tremendously useful for blind and visually impaired users. VQA systems can also provide an image retrieval system, which could help the local authorities to solve cases from image retrieved from CCTV cameras to identify the criminals. It can also have an impact on e-commerce trends etc.

### **1.3. Problem Statement**

This thesis explores methodologies to build a framework (henceforth called “The Framework”) for a Visual Question Answering (VQA) system that can automatically identify answers for a question in a natural language about an image presented to it. The study explores the research in building a Visual Question Answering system using deep learning algorithms based on CNN for image recognition, and LSTM networks for NLP for question analysis for generating attention. The research categorizes the visual question answering technique as a classification problem [15]. For a given image and a textual question in the form of natural language, the system estimates the most likely answer from

a fixed set of answers based on the content of the image [15]. The model uses a deep residual network (ResNet), an advanced CNN model to compute the image features, and LSTM, a special type of RNN to compute question embeddings combined with an attention mechanism to focus on most relevant parts of the image, and the probabilities over an answer set is generated using a classifier.

#### 1.4. Overview of the Framework

In a nutshell, The Framework is a Visual Question Answering (VQA) system for image identification using the techniques and algorithms of deep learning with CNN and the Natural Language Processing (NLP) techniques which process the user provided question to combine the features learned from the user given image and question to predict an answer. The VQA system consists of three major segments- *Feature extraction*, *feature fusion* and *Classifier*, which will be explained in detail in the further sections. The overall flow of information and connection of various models used in The Framework with attention-based method is shown in figure 1 below [21].

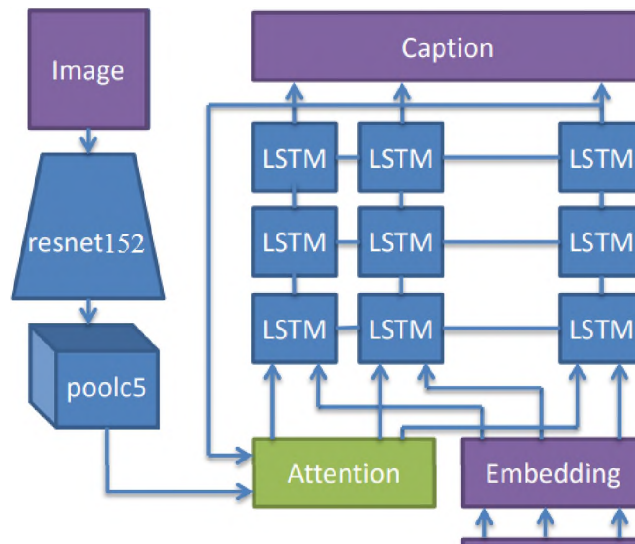


Figure 1. Information flow and models connectivity of The Framework



The Framework is divided into two components whose individual outputs are combined using attention-based technique to predict an answer. The first component is a question analyzer with Long Short-Term Memory (LSTM) that extracts features of user given question sentence using Natural Language Processing (NLP) methods. The second component is for image processing, object identification and action recognition using a Convolution Neural Network (CNN) model.

For sentence analysis using Natural Language Processing techniques, an analysis pipeline is generated using a sequence of annotators. The input to the annotator is the raw text itself.

The resulting annotation information containing all the information is then output in XML or plain text format. Part-of-Speech (POS) Labels are used in The Framework, which has tokens with their POS tag, using a maximum entropy POS tagger (Toutanova et al., 2003). Figure 2 below shows the CNN architecture and the various layers involved in image feature extraction and image classification. Figure 3 represents the LSTM network with attention model for question text processing.

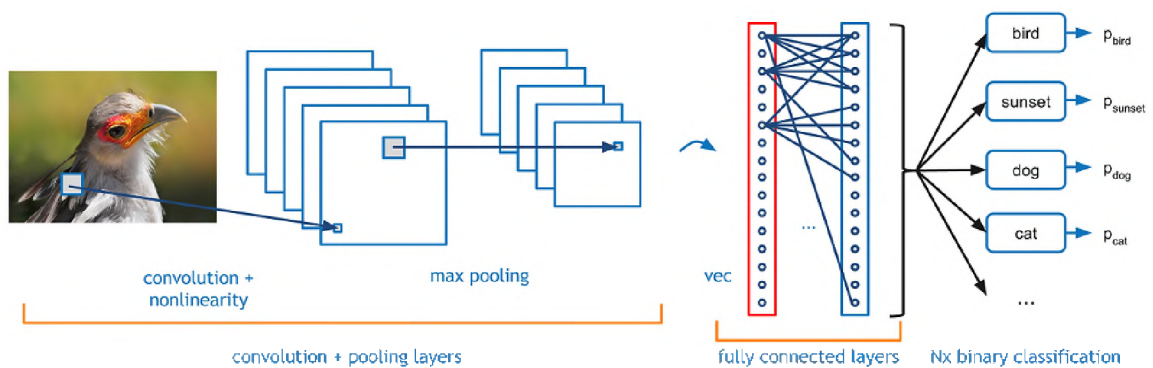


Figure 2. CNN architecture

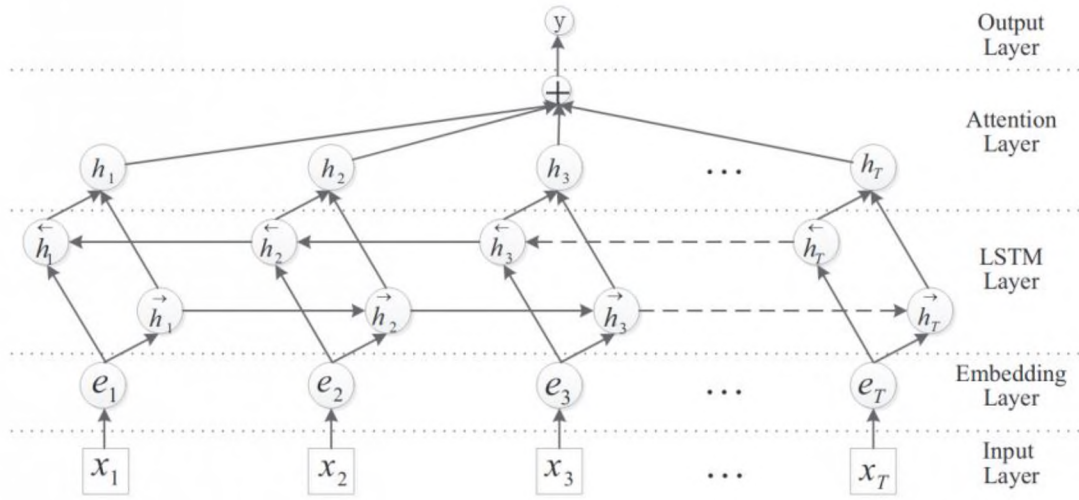


Figure 3. LSTM with attention architecture

The QA system was built, trained and validated with an attention mechanism. The comprehensive evaluations were performed on an image question answering benchmark. The results demonstrate that the multi-layered attention mechanism outperforms previous approaches in the literature with significant margins and shows the results at par with existing attention-based models in the literature. A detailed analysis with visualizations was done to show that the outputs of various attention layers of the model indicate the progressive focus of the attention model on the relevant visual parts, omitting the irrelevant part of the image to lead to the answer.

The questions are based on facts (e.g. what is the average life expectancy in Ohio?) or complex narrative questions (e.g. what do analysts think about Bill Gates new foundation dealing with clean water crisis in Asia?). Visual question answering problems are unique and very different from the previous object detection systems in the sense that solutions of VQA systems are varied and unpredictable. Besides predicting an answer for an object in the image [32] ("What is there in the image?"), they also recognize the exact object to

produce an answer ("Is the detected object a cat?") [32]. Then, VQA systems also classify the objects based on its attributes ("What is the color of the cat?"), scenario ("Is it day or night?") or produce a count the number of times the object has occurred in an image ("How many cats are there in the image?") [32]. Another challenging task is the proposed natural language questions that are only known to the system during run-time.

The final challenge was to combine the result of natural language processing with the image data focusing on most relevant areas of the given image (attention) to produce a set of possible answers with a cutoff accuracy. The approach used in The Framework addresses the problem to employ deep residual network (ResNet) and LSTM. The experiments were done using the well-known COCO data set [14]. The results showed that the approach used with ResNet and attention-based LSTM networks improved up to 5% in accuracy when compared to the existing literature using other well-known models- AlexNet [6] and VGG16 [37]. The Framework, despite a simple architecture, achieves an accuracy of 59% on VQA 1.0 [53] and COCO datasets.

The rest of the thesis is organized in the following manner. Chapter 2 discusses the literature review and related work done to build a VQA system using other deep learning-based models. Chapter 3 explains the CNN architecture and the main steps involved in detail. Chapter 4 gives an overview of the RNNs and explains the LSTM network with attention based RNN in detail. Chapter 5 discusses the technical approach and the methodology used in The Framework, including the image and question models, explanation of ResNet architecture and feature extraction. Chapter 6 shows the experimental results and evaluation done on The Framework. Finally, chapter 7 concludes the thesis also mentioning about the future scope of work.

## **CHAPTER II**

### **LITERATURE REVIEW AND RELATED WORK**

#### **2.1 Literature Review**

Visual Question Answering (VQA) system and image captioning systems are closely linked as both need a reasoning about the visual contents and output either a word or a full sentence. Some of the work has been accomplished using attention-based configurable convolutional neural network (ABC-CNN) to locate the question-guided attention based on input queries [3]. The technique used here is based on question guided attention maps (QAM) where QAM is achieved by convolving the image feature maps and the convolution kernels obtained by different dense question embeddings.

Some of the other similar work done uses the power of backpropagation techniques to train the model deterministically by maximizing a variational lower bound [28]. This is achieved by using a stacked attention networks (SAN) as in [4]. This is closely related to The Framework in the sense that it uses SANs to represent a question semantically and search for only specific regions of the image related to the question. This is achieved in multiple steps of reasoning. Thus, a multiple-layer SAN infers the answer progressively in this case. When the answer only relates to a smaller region of the whole image, it is valid to say that using the one global image feature vector introduces noise from the irrelevant

regions with respect to the answer, hence producing suboptimal results [4]. Instead, using SANs the image feature and question vector are first fed through a single layer neural network and then an attention distribution is generated over the specific regions of the image using a Softmax function [4].

There are several other recent papers to address VQA [2, 3, 4, 5, 8]; most of them are based on deep learning except [2]. Seo et al. [2] also mentions about a methodology proposing a Bayesian framework, which exploits current advances in natural language processing and image processing even though this method depends on a pre-defined set of predicates. This eventually creates difficulties in representing complex models which are required to understand input images.

Deep learning-based approaches demonstrate competitive performances in VQA [4, 6, 8]. Most common VQA approaches use CNN for image feature extraction and use various different techniques to handle question sentences. Some algorithms [52] employ embedding of joint features based on an image and a question. A currently used approach uses pre-trained neural networks such as ResNet [21] and VGG [37], pre-trained on the ImageNet corpus to extract the visual information of the input image. This image feature extraction method, has proven to achieve state of the art results in many systems, including VQA.

Stack based coarse-to-fine multistage prediction approach [29] propose a prediction framework for image captioning where multiple decoders operate on the output of the previous stage, resulting in improved image descriptions. They optimize the attention-based model with a reinforcement learning approach which normalizes the words by using the output of the inference algorithm of intermediate decoder along with its preceeding

decoder. It is claimed that this model solves the exposure bias and loss-evaluation mismatch problem [29].

Since the creation of AlexNet architecture in 2012 [6], there have been progressive researches in the field of CNNs for image recognition and the percentage accuracy for object identification has significantly increased. Over these years, there have been various other CNN based models such as VGG, GoogleNet and their enhanced versions due to which the task of image recognition has become even more innovative. In 2015, ResNet model won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) challenge on ImageNet classification, detection and localization and COCO detection and segmentation with only 3.6% top 5 error, which is considered the best so far.

The Framework has taken into consideration all the previous researches done in the same field and based the experiments and results to reproduce and improve the accuracy of results that have been achieved.

## **CHAPTER III**

### **INTRODUCTION TO CONVOLUTION NEURAL NETWORKS**

#### **Overview**

It was Fukushima who first proposed the idea of Convolution Neural Network (CNN) in 1988. Since then, many researches and improvement have been going on to improve the image recognition, face recognition and object identification tasks in the field of CNNs. CNNs offer many advantages over its previous architectures of Deep Neural Network such as providing human like visual processing system which is effective for processing 2D and 3D images, its learning and abstractions. CNN demonstrates to be better as it exploits the way that the input images sensibly constrain the architecture [5]. The layers of a CNN have neurons organized in 3 dimensions: width, height, depth in contrast to a customary Neural Network. Depth here alludes to the third dimension of an activation volume. For instance, the images in CIFAR-10 are a volume of activations of the dimensions  $32 \times 32 \times 3$  (width, height, depth respectively). The neurons in a layer are associated with a smaller layer before it, rather than the all the neurons connected in a fully connected way. The last output layer for CIFAR-10 have dimensions  $1 \times 1 \times 10$ , as the complete image is reduced into a single vector class score by the end of the CNN based model [5]. This layer is arranged along the depth dimensions. Figure 4 below shows a visualization of the CNN architecture:

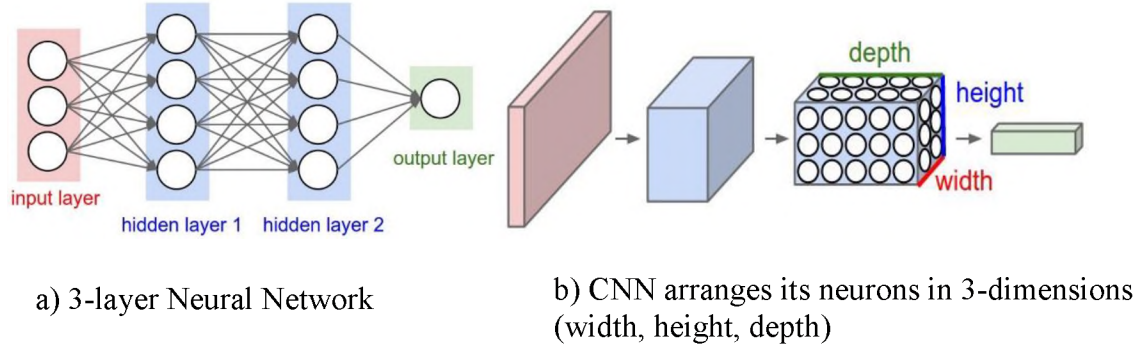


Figure 4: Convolution Neural Network Visualization [5]

The CNN in Figure 5 below shows four classes (cat, bird, dog and boat) of the input image. From the below image, the network correctly assigns the highest probability of 0.94 for boat amongst the four classes, when a boat image is input to it [47]. The sum of all probabilities in the output layer is always equal to one [47].

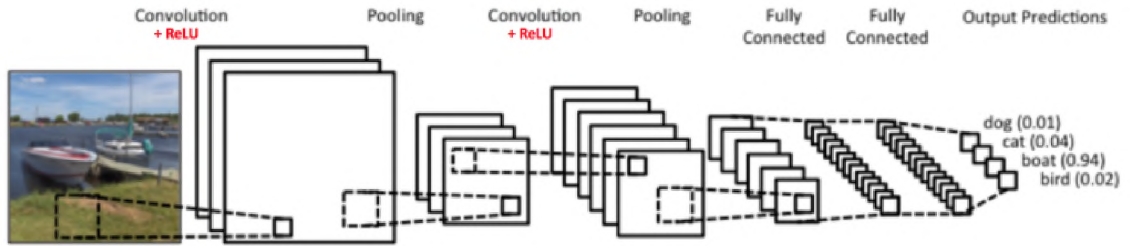


Figure 5: A simple ConvNet. [47]

CNNs consists of two main parts in terms of functionality: Feature Extraction and Classification which are achieved by four main operations: Convolution, Pooling or Sub Sampling, Non-Linearity- also known as ReLU and finally Fully Connected Layer. In feature extraction, each layer of the network passes its output as the input to the next layer and gets its input from the previous layer. The nodes of convolution and max pooling combine and result in a grouped 2D plane called feature mapping. These operations of convolution, pooling, normalization and fully connected layers are the basic building blocks of any CNN, hence the thesis discusses them and how they work in the following pages.



### 3.1 Convolution Layer

The name ConvNets comes from the ‘convolution’ operator. The primary purpose of the Convolution layer in ConvNets is to extract features from the input image. As shown in figure 6 below [16], consider a 6x6 image represented as a matrix of pixel values ranging from 0 to 255 and a 3x3 matrix called a ‘filter’ or ‘kernel’. The filter slides over the image matrix and the computed dot product of the two results in an ‘Activation Map’ or ‘Feature Map’. Convolution operation maintains the spatial relationship between pixels by when made to learn the image features using smaller segments from the input data [47].

Furthermore, these filters are made to learn about the dot products of the entries of the filter and the input width, height and depth of the images. So, the filters slide spatially over the image and extends through the full depth of the image. These filters, when convolved over the input dimensions of the width and height of the input image, produce an entire set of filters in each layer to give a 2-dimensional activation map as a result of the response of that filter at every spatial position. Together, these stacked activations maps along the depth dimension produce the output volume.

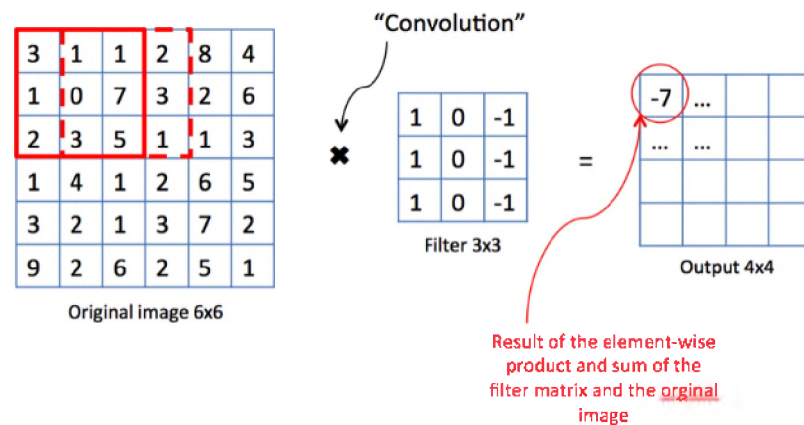


Figure 6: Convolution layer [16]

Mathematically, the Convolution is represented by the following notation. Given functions  $u(t)$  and  $w(t)$ , the *Convolution* is an integral of the product of  $u(t)$  and  $w(t)$  functions after one is reversed and shifted. The resultant function  $s(t)$  can be written as [45]:

$$s(t) = \int u(a) w(t-a) da$$

$$\text{Or } s = (u * w)$$

$$\text{or } s(t) = (u * w)(t)$$

The Convolution is represented as the weighted average of the function  $u(t)$  at time  $t$  where the weight is given by  $w(t)$  moved by amount  $t$  [45]. The weighting function focusses on different parts of the input function with change in the value of  $t$ . The discrete Convolution of  $u$  and  $w$  is represented as [45]:

$$s_t = \sum_{a=-\infty}^{+\infty} u_a w_{t-a}$$

$u_t$  and  $w_t$  are assumed to be 0 if not defined. The convolution of two finite sequences  $u$  and  $w$  is “...[d]efined by extending the sequences to finitely supported functions on the set of integers. When the sequences are the coefficients of two polynomials, then the coefficients of the ordinary product of the two polynomials are the convolution of the original two sequences” [48].

### 3.2 Non-Linearity: (ReLU)

ReLU stands for Rectified Linear Unit. The rectified linear activation function has become the most widely used activation function when it comes to adding non-linearity in the network of real-world information as it's easier to train and achieves better

performance. It has proven to perform better than sigmoid or tanh activation functions. Figure 7 shows the graph representing ReLU function [17].

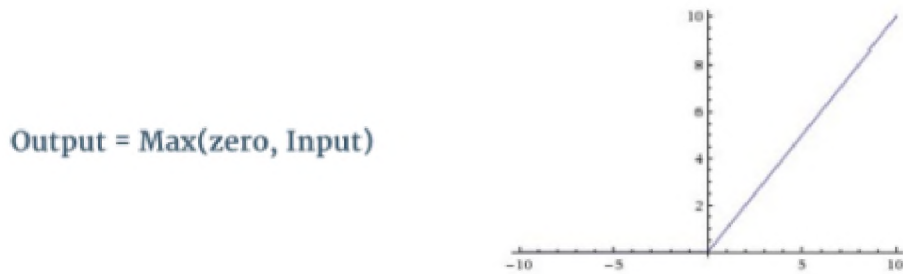


Figure 7: ReLU function [17]

ReLU is an element (per pixel) wise operation that replaces all negative pixel values in the feature map by zero and keep the positive pixel values as it is. It introduces non-linearity in ConvNet. Since the real-world data in the ConvNet is mostly non-linear, so a function like ReLU is introduced to account for non-linearity [48]. The ReLU operation is shown in Figure 8 below. It shows that when ReLU operation applied to a feature map obtained from the convolution layer, results in an output feature map, referred to as the ‘Rectified’ feature map.

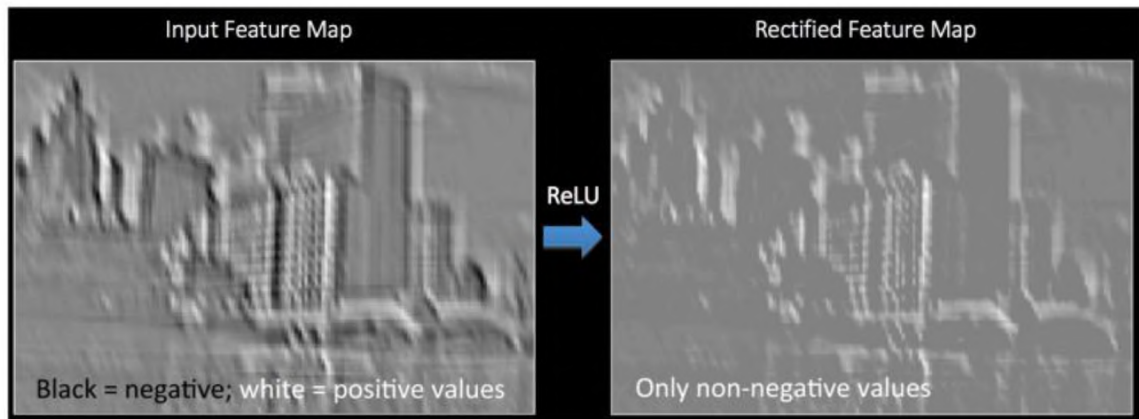


Figure 8: ReLU operation [45]

### 3.3 Pooling or Sub-Sampling

Spatial pooling or Sub-sampling layer follows the convolution layer. It is mainly used to down-sample the output of a convolution layer by reducing the number of parameters to be learned by the network along both the spatial dimensions of height and width but retaining the important information. This helps in overcoming the overfitting problem resulting in an increased performance and accuracy [18]. There are various types of pooling layers: max pooling, average pooling, sum pooling, etc. Figure 9 shows the max pooling layer, with a stride of 2. In this case, a spatial neighborhood of  $2 \times 2$  window in this case is defined and the largest element from the rectified feature map or the average (in case of average pooling) or sum of all elements in that window is considered within that window. Practically, max pooling gives better results than its counterparts [45].

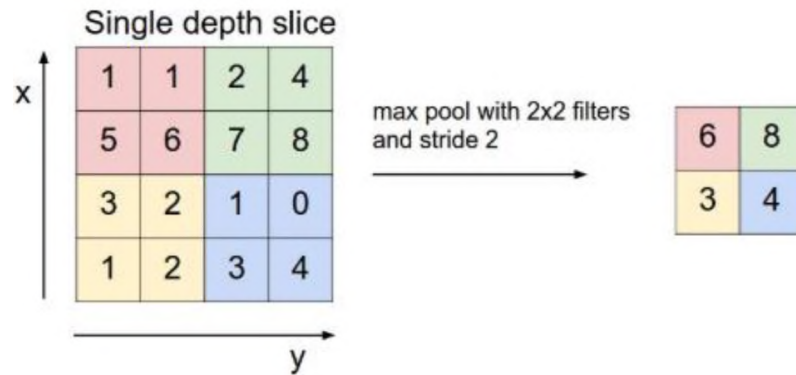


Figure 9: Max pooling [5]

When pooling is performed on the rectified feature map received after the ReLU operation done on Figure 9 above, the resultant output representation is shown in figure 10.

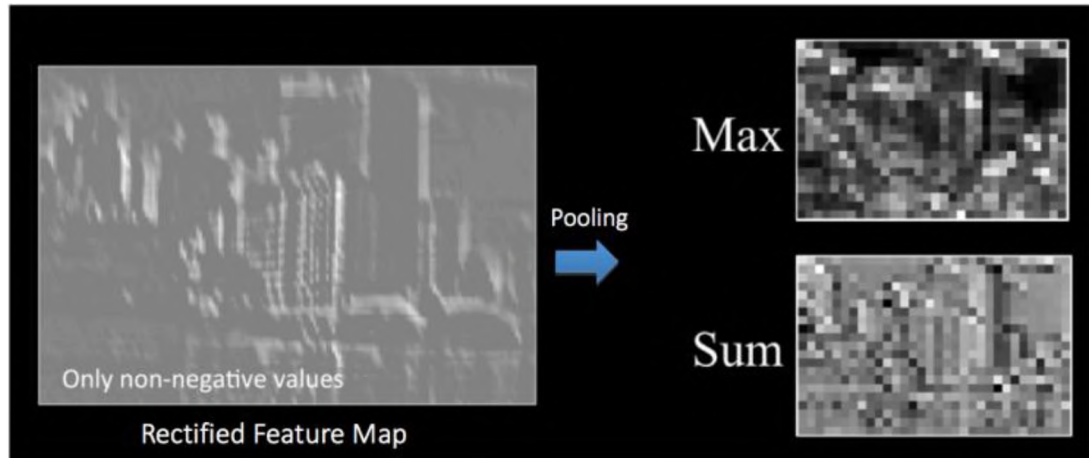


Figure 10: Pooling [45]

### 3.4 Fully-Connected Layer

The features extracted as a result of the convolutional and max pooling layers, act as an input to the fully connected layer. As seen in figure 5, there are two sets of each Convolution, ReLU & Pooling layers. The second convolution layer takes the output of the first pooling layer as its input and performs convolution using six filters. This results in a total of six feature maps. Then, ReLU is applied on each of these six feature maps individually. The next step is to apply the max pooling operation on each of these six rectified feature maps separately. When combined, these layers extract the useful features from the input images, introduces non-linearity in the network and reduces the feature dimension. The features are aimed to be equivariant to scale and translation [45].

The output from the second pooling layer now becomes an input for the current fully connected layer. A fully-connected layer is a traditional multi-layer perceptron that classifies an image into a label by taking input from the convolution/pooling layer, depending on the training data. It uses a Softmax activation function in the output layer to ensure the sum of output probabilities from the fully connected layer is 1 [45]. The term

‘fully connected’ indicates that every neuron of the previous and the next layer are connected to each other making it a ‘fully connected’ layer [45].

### 3.5 Softmax Function

When Softmax is applied in the last layer of fully connected layer, it takes a vector of random real-valued scores and squeezes the values to be in-between zero and one that sum to one to form an output vector [45].

The standard (unit) Softmax function is [49]:

$$\sigma : \mathbb{R}^K \rightarrow \mathbb{R}^K$$

and defined by the formula:

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \text{ for } i = 1, \dots, K \text{ and } \mathbf{z} = (z_1, \dots, z_K) \in \mathbb{R}^K$$

where  $K$  = input vector of real numbers and standard exponential function to each element  $Z_i$  of input vector  $Z$  is applied.

The values are normalized by dividing by the sum of all these exponentials to make sure that the sum of the components in the output vector  $\sigma(z)$  is one [49].

Softmax takes  $K$  as input and normalizes it into  $K$  probability distribution proportional to the exponentials of the input numbers. If there are any vector components which are negative or greater than one or might not sum to 1, applying Softmax to such a vector results in each component to be in the interval (0,1) and the components add up to 1 [49]. Softmax is often used in neural networks (where larger input components correspond to larger probabilities), to map the output of a network (which are not normalized) to a probability distribution over predicted output classes [49].

### 3.6 Cost Function

A cost function is used to quantify the effectiveness of a neural network with respect to the input training data and the expected output, sometimes also depending on weights and biases variables. A cost function is a single value that assesses how good the neural network did.

The neural network model is trained using batch gradient descent. If there is a single training example  $(x, y)$  [44], the *Cost Function* with respect to that example is defined as:

$$J(W, b; x, y) = \frac{1}{2} \|h_{W,b}(x) - y\|^2.$$

The equation shown above is a (one-half) squared-error cost function [44]. Given a training set of  $m$  examples  $(\{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\})$ , the *Overall Cost Function* is defined as:

$$\begin{aligned} J(W, b) &= \left[ \frac{1}{m} \sum_{i=1}^m J(W, b; x^{(i)}, y^{(i)}) \right] + \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (W_{ji}^{(l)})^2 \\ &= \left[ \frac{1}{m} \sum_{i=1}^m \left( \frac{1}{2} \|h_{W,b}(x^{(i)}) - y^{(i)}\|^2 \right) \right] + \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (W_{ji}^{(l)})^2 \end{aligned}$$

Where,  $J(W, b)$  is the average sum-of-squares error term and weight decay term  $\lambda$  is a regularization term which helps prevent overfitting by decreasing the magnitude of the weights. This cost function is often used for classification as well as regression problems [44].

Finally, the core behind CNN is the *Backpropagation Algorithm* which is explained as: “...[G]iven a training example  $(x, y)$ , run a ‘forward pass’ to compute all the activations throughout the network, including the output value of the hypothesis  $h_{W,b}(x)$ . For each node  $i$  in layer  $l$ , compute an ‘error term’  $\delta_i^{(l)}$  which measures for any errors in the output for every node. Measure the difference between the network’s activation and the true target

value for an output node, which is used to define  $\delta_i^{(n_l)}$  (where layer  $n_l$  is the output layer). For hidden units, compute  $\delta_i^{(l)}$  based on a weighted average of the error terms of the nodes that uses  $a_i^{(l)}$  as an input” [44].

### 3.7 Batch Normalization

Batch normalization addresses the problems within the feature maps related to internal covariance shift. The internal covariance shift can be described as a revision in the values of hidden unit distribution, which slows the convergence, as the learning rate is pushed to a smaller value [11]. Batch normalization for a transformed feature map  $T_l^k$  can be represented as:

$$N_l^k = \frac{F_l^k - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

Where,  $N_l$  = normalized feature map

$F_l^k$  = input feature map

$\mu_B$  and  $\sigma_B^2$  represent mean and variance of a feature map respectively.

Batch normalization brings the feature map values to unit variance and zero mean, thus fusing the distribution of feature map values [11]. Along with unifying the feature map values, batch normalization also regularizes the flow of gradient and acts as a smoothening factor, which helps generalize of the network in a better way [11].

### 3.8 Dropout

“...[D]ropout introduces regularization within the network, which ultimately improves generalization by randomly skipping some units or connections with a certain probability. In NNs, multiple connections that learn a non-linear relation are sometimes co-adapted, which causes overfitting. This random dropping of some connections or units produces several thinned network architectures, and finally one representative network is selected with small weights. This selected architecture is then considered as an approximation of all of the proposed networks” [11].



To summarize this chapter, to enable a convolution neural network for feature extraction and classification, the input data passes through four main operations of convolution, non-linearity (ReLU), pooling/sub sampling, fully connected layer to get the final classification. The overall training process of the ConvNet is summarized as below [44]:

1. Initialize all filters and parameters/weights with random values.
2. Forward propagation step (convolution, ReLU and pooling operations along with forward propagation in the fully connected layer) on an input image- find output probability for each class.
3. Assuming the output probabilities for the boat image in figure 5 are [0.1, 0.6, 0.2, 0.1], they are still random, since the weights are randomly assigned for the first training input image.
4. Calculate sum of error at output layer (for all 4 classes) [44]

$$\text{Total Error} = \sum \frac{1}{2} (\text{target probability} - \text{output probability})^2$$

5. Backpropagate to find the gradients of error with respect to all network weights [44].
6. Use gradient descent to minimize output error. Update filter values/weights and parameter values.
  - 6.1 Weights adjustment is proportional to their contribution to total error.
  - 6.2 The same image will have different output probabilities when inputted next time taking it closer to the target vector. This implies that means that by adjusting the weights and filters correctly, the network has learnt to classify eventually reducing the output error.
7. Repeat steps 2-6 for all images in training set.

The above steps will correctly classify images from the input training set by optimizing all the weights and parameters of the convolution network. The next chapter describes Recurrent Neural Networks, laying emphasis on long short-term memory units and their internal working as they are used in question embedding tasks for The Framework.

## **CHAPTER IV**

### **INTRODUCTION TO RECURRENT NEURAL NETWORKS**

#### **Overview**

Recurrent Neural Network (RNN) is unique to neural networks in the sense that it is designed to identify patterns in sequences of data (time-series, handwriting, numerical data series etc.) generated from different sources like various sensors, stock markets, handwritten books etc. RNNs take two sources as the input, the data from the present step, and the most recent past (previous step), based on which a decision on the new data is made [15]. The decision a recurrent neural network will reach at time  $t$  is affected by the decision it has reached at time  $t-1$  [15]. The outputs are fed back to the inputs continuously, making it the vital differing point with the feed-forward neural network. Figure 11 shows a complete sequence of an RNN.

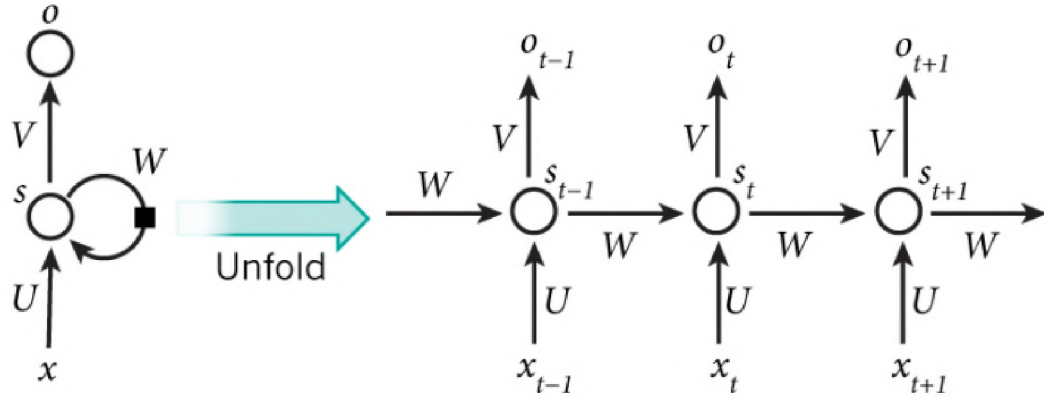


Figure 11: Recurrent Neural Network structure [15]

Recurrent Neural Networks (RNN) tie the weights at each time step where the hidden state is updated in a deterministic nonlinear way. The main advantage of RNNs over traditional neural network is that the RAM requirement only scales with number of words. There are two sources of input to a recurrent neural network at every timestep, the current and the most recent previous, which together combine to give the output/decision.

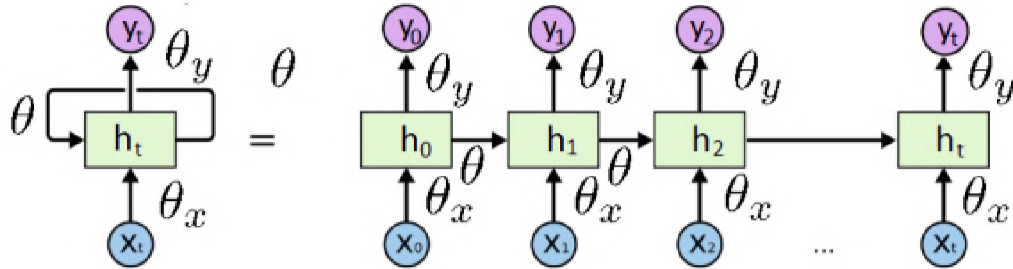


Figure 12: Recurrent Neural Network representation [19]

Mathematically, RNNs can be represented as [19] shown in figure 13 below.

$$\mathbf{h}_t = \theta \phi(\mathbf{h}_{t-1}) + \theta_x \mathbf{x}_t$$

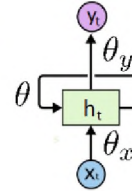
$$\mathbf{y}_t = \theta_y \phi(\mathbf{h}_t)$$


Figure 13: Equation for Recurrent Neural Networks [19]

Where,  $x_t$  = input at  $t$ .  $x_t$  considered as one-hot vector regarded as the second word of the sentence.

$h_t$  = hidden state (memory) at time  $t$ .  $h_t$  is calculated by taking the current input value and the previous hidden state.

$y_t$  = output at time  $t$ . The predicted next word of a sentence. This can be done considering a vector of probabilities in the vocabulary.

$\theta, \theta_x, \theta_y$  = same distinct weights at all time steps

#### **4.1 Back Propagation Through Time (BPTT)**

Backpropagation Through Time (BPTT), is the application of the Backpropagation training algorithm to RNN data which is a sequence like data as explained above (like time series). BPTT is recursive in nature with respect to the weights applied at each step and its effect on the loss which distributes over time [50]. BPTT follows the process of initializing the weight matrices randomly to begin with. Then, forward propagation is applied to compute the predictions. Next, the loss is computed at each time-step using the chain rule, and backpropagation is applied to compute the gradients. Finally, the weights are updated based on gradients and these steps are repeated. Once the gradients are available for the input weights, they are updated, and the same steps are repeated with the backpropagation workflow [50]. BPTT works for an ordered sequence of  $k$  input-output pairs. The RNN is unfolded in time to begin the BPTT. The unfolded network contains  $k$  inputs and outputs where the shared parameters are present in every step of the network [51]. The backpropagation algorithm helps identify the gradient of the cost with respect to the network parameters. Eventually, the weights are updated in each instance of the recurrent layer and are summed together [51].

## 4.2 The Problem of Long-Term Dependencies & Vanishing Gradient

RNNs connect the information from the past step to the present step by adding a correction term at the output of each neuron which may be enough for predicting the next word in each sentence or word sequence but may not be enough when more context is needed. For example, in the following sentence, the previous step's information may be enough: "*the car is in the garage*" but might need more context for the following sentence: "*I have a car... it gives excellent mileage*". In RNNs, during the back propagation through time gradient phase, the gradient signal or the correction term gets multiplied multiple times ending in a correction term greater than one. If the gradients are greater than one, the problem of exploding gradients can occur due to which the learning diverges. A known solution for exploding gradient would be to limit the gradients to a certain max value. On the other hand, if the gradients are very small, also called as vanishing gradients, the learning rate of the network becomes very slow or even stops. The solution to such a problem of vanishing gradient is the use of Long Short-Term Memory (LSTM) networks as described in the next section.

## 4.3 Long Short-Term Memory (LSTM) Networks

Long Short-Term Memory (LSTM) networks solve the problem of vanishing gradient as explained above in the RNNs. They are capable of learning long-term dependencies [20]. Unlike the regular RNNs, where a node has an activation function, every node in LSTM network can store not only its own state or its previous step's information, but also of states which occurred many time steps back this is achieved by using a memory cell which uses logistic and linear units with multiplicative interactions.

There are three main components of LSTM network: *input gate*, *forget gate* and *output gate*. The cell can store the information whenever the input gate is on. When the forget gate is off, the information from the cell is thrown away. Finally, the information can be retrieved from the cell by turning on its output gate. Figure 14 below shows the LSTM network cell (Hochreiter et al., 1997).

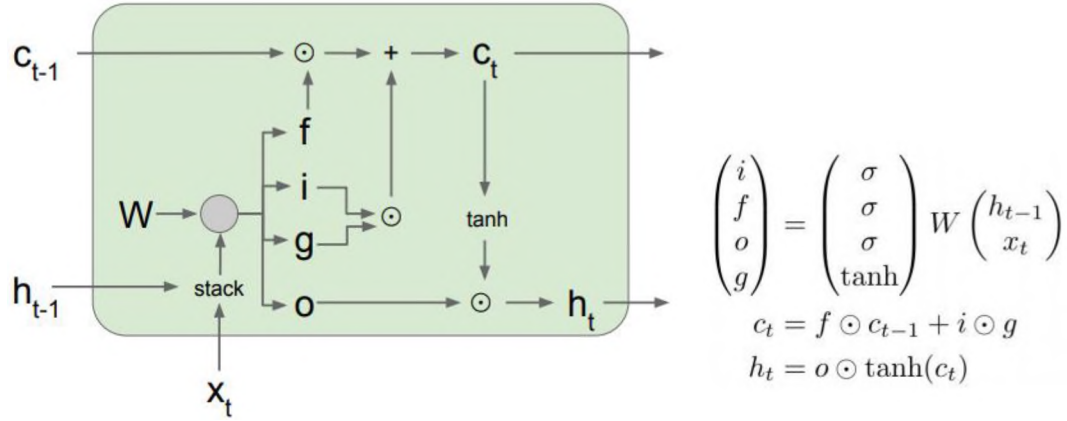


Figure 14: Long Short-Term Memory (LSTM) (Hochreiter et al., 1997)

Here,  $C_t$  is the memory cell

$$\tilde{c}_t = \text{Tanh}(W_c \cdot [h_{t-1}, x_t] + b_c)$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$$

Forget gate,  $f_t$  is in  $[0, 1]$ , which resets the old memory cell value

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Input gate,  $i_t$  is in  $[0, 1]$ , which writes input to the memory cell

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

Output gate,  $o_t$  is in  $[0, 1]$ , which reads output from memory cell

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

Output  $h_t$

$$h_t = o_t * \text{Tanh}(c_t)$$

There are three major steps in LSTM networks. First, the cell state remembers only the appropriate information from of the previous cell state. The rest of the irrelevant information can be forgotten. Second, the cell state is updated selectively depending on the new input. Finally, a decision is made on the new hidden state from the portion of the cell state that should be considered as output [15].

The information can flow along the memory cell unchanged or it can be removed or written to the memory cell which is regulated by the input and output gates. The gates are a way to decide whether to let information pass through or not. A *sigmoid* layer outputs number between 0 and 1, deciding how much of each component should be let through which is then multiplied to the input to get the output. Similar concept is applied to input and forget gates as well. A *Tanh* will create a vector of new values  $\tilde{C}_t$  to write to the memory cell. Once the decision has been made on which values are to be reset and overwritten, the LSTM applies the decisions on the memory cell, where the sigmoid layer decided which value of the memory cell must go through output. Finally, the memory cell is multiplied by the output gate and passed through *Tanh*.

LSTM have become very popular in the field of natural language processing ever since they have been in place. Their additive interactions in the network improve the gradient vanishing problem.

The next chapter presents the detail technical solution, approach and implementation details of The Framework.



## **CHAPTER V**

### **TECHNICAL APPROACH AND METHODOLOGY**

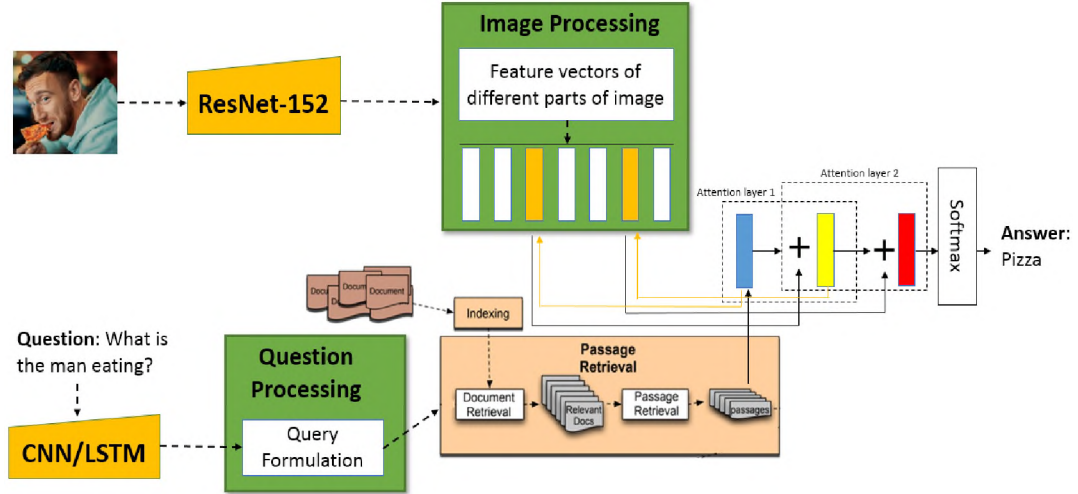
#### **Overview**

In 2015, when Deep Residual Learning for Image Recognition [22] also called as ResNet, won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in image classification, detection, and localization, as well as won the MS COCO 2015 detection, and segmentation by a significant margin, it was intriguing to understand how this is going to change the existing industry-wide algorithms and ways in which image recognition is done. This chapter formalizes the task of Visual Question Answering system and discusses the technical approach for the ResNet and LSTM with attention-based solution in detail.

#### **5.1 Architecture**

The Framework uses a 152-layer deep residual network (ResNet) to compute the image features and LSTM to compute question embedding. Depending on the question embedding, two attention layers are applied to output two most significant glimpses of the image features. For answer prediction, a probabilistic classifier is used over an answer set. The two focused glimpses of image features along with the final LSTM output are passed as an input to a classifier to generate an answer.

Figure 15 below shows the complete architecture of The Framework.



(a) Architecture of The Framework for VQA



(b) Visualization of multiple attention layers. The stacked attention network focuses on referred features first including all different kinds of objects in the image. In the second layer, it narrows down the focus and finds out the answer pizza.

Figure 15: Model architecture and visualization

In the design of The Framework, the task of visual question answering is assumed as a conventional classification problem. For a given input image and text question, The Framework predicts and answer from a set of answers  $\hat{A}$ , depending on the image content.

$$A = \arg_{\hat{A}} \max P(\hat{A} | I, Q)$$

The Framework has three main components which are discussed in detail below:

1. Image model: This is implemented to extract features of the image using 152-layer ResNet, a CNN model

2. Question model: This is implemented using a long short-term memory unit to encode the semantics of the given question.
3. Stacked Attention: This focusses on the most relevant part of the image instead of predicting an answer on the whole image.

### 5.1.1 Image Model

The image model computes a high-level representation  $f_i$  of the given image  $I$ , a pretrained convolutional neural network (CNN) model based on residual network architecture [6] which is a pre-trained model.  $f_i$  is image feature map extracted from the raw image  $I$ .

$$f_i = CNN_{ResNet}(I)$$

The feature map  $f_i$  has 14 X 14 X 2048 dimensions. This the three-dimensional tensor from the output from the last layer of ResNet, which is then applied to the final pooling layer. By performing this, the spatial information of the images can be preserved. The image model “...[f]urthermore perform  $l_2$  normalization on the depth (last) dimension of image features which enhances learning dynamics” [21].

### 5.1.2 Question Model

The input question  $e$  is tokenized and word embeddings  $W_e$  are extracted from it.  $W_e = \{W_1, W_2, \dots, W_n\}$  where  $N$  is the number of words in  $e$ . The embeddings  $W_e$  are then passed to LSTM as input. The final state of LSTM is represented as  $s$ .

$$s = LSTM(W_e)$$

### 5.1.3 Stacked Attention

Stacked attention mechanism is one of the recent successful method to consistently improve the overall accuracy of neural network models. It gives neural networks the

flexibility to compute localized image features. When we weight the spatial dimensions of the Convolution Neural Network's image features according to the importance of the assessed image features, we call it as 'attention mechanism'. In other words, there can be multiple objects in an image, like dog, cat, telephone, lamp etc. and predicting an answer using a global image feature vector can lead to noise around the regions irrelevant to the required object's image resulting in sub-optimal result. Attention mechanism gradually filters out the noises focusing only on the regions relevant to the image producing higher level of accuracy.

In The Framework, the question embeddings are not fixed size when passed as unlike the image embeddings which are of the size  $14 \times 14 \times 2048$ . A single vector for each region is formed by concatenating the word embedding vector with different image features. This single vector representation consists of the localized region of the image along with the entire question semantics. With the application of multiple Convolution and ReLu layers on the concatenated vectors, the resultant section of the initial Convolution, ReLu layers identify the significant areas of the image based on the question embeddings [15]. The next convolution layer after the ReLU, results in a vector by focusing on the significant regions identified in the previous section. This convolution layer then goes through a Softmax and passes to the weighted average layer. The input to the weighted average layer is a condensed vector representation of the concatenated vectors [15].

The image feature vector is sent as an input to the attention distribution to include varied spatial feature regions to produce a concentrated representation of the input image that concentrates on the most significant spatial areas as compared to others [15]. Finally, the

question representation vector is fused with the focused image feature vector and passed through a Softmax layer to get an output textual answer.

As described in [20], The Framework adds an ‘attention gate’ to the LSTM architecture. Eventually, the new equations with attention and LSTM become

$$\begin{pmatrix} i \\ f \\ o \\ g \\ a_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \\ \text{softmax} \end{pmatrix} \circ W \begin{pmatrix} I \circ a_{t-1} \\ h_{t-1} \\ x \end{pmatrix}$$

where  $a_t$  = attention parameters at time  $t$ . Rister et al. 2016 explains the equations in detail and says that “The attention parameters define a linear transformation of the input image. Furthermore, the Softmax function ensures that for all elements we have  $a_t > 0$  and for the whole vector we have  $|a_t|_1 = 1$ . In other words, the attention weights are positive and sum to one.” [20]

## 5.2 Feature Extraction

The feature extraction block is be divided in two separate steps for input question and image respectively. The Framework applies the advance neural network architecture called ResNet for image feature extraction which is explained below. For the question feature extraction, The Framework uses question tokens by an embedding layer and then passing it to LSTM module. LSTM gives us the representation of the final question feature.

### 5.2.1 Image Feature – ResNet

The Deep Residual Network (ResNet) architecture is getting extreme deep networks because of the use of residual connections explained later. The main question ResNet tries to solve is that what happens when deeper and deeper layers are stacked on a plain convolutional neural network. The hypothesis of the ResNet is that it is mainly an

optimization problem where deeper models are harder to optimize (He et al., 2015) and the deeper do not guarantee a better performance.

ResNet architecture presents answers to three main problems which its predecessors had. The first problem seen in previous implementations of CNN is when the input and output are equal for a dataset and a multi-layered network needs to be trained on it. One solution to this can be to keep weights equal to one and biases as zero for all hidden layers [23]. But this kind of a network produces a complex mapping when trained through backpropagation as the weights and biases have a varied range in their values [23].

The second problem ResNet answers is the *vanishing gradient problem*. Adding more layers to existing neural network decreases the accuracy for the network, whereas it should increase if the over-fitting is taken care of. As the depths of the layers increase, the prediction becomes significantly small at the initial layers and it becomes hard to change the weights at the end of the network as the required signal is negligibly small. Hence, the initial layers are almost negligibly learned [9]. This is called *vanishing gradient*.

Third problem with training deeper network is getting higher training error when there are huge number of parameters and adding more layers to handle it leads to the problem, also called as degradation problem.

ResNet architecture uses shortcut connections to solve the vanishing gradient problem. The basic building block of ResNet is a Residual block. It is repeated throughout the network. Figure 16 below shows the “plain layers” and the residual block [21].

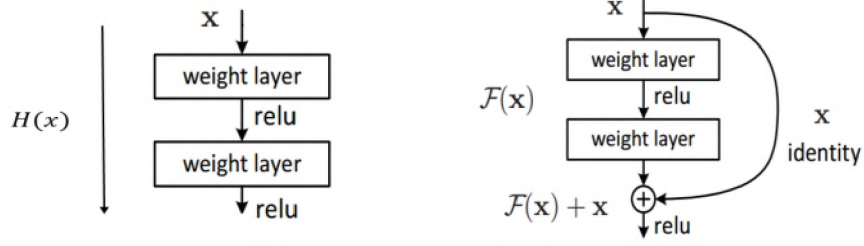


Figure 16: Plain layers and residual block in ResNet [21]

### 5.2.1.1 Residual Network

The residual block solves the degradation problem using what is known as shortcut connections, which connects the feed forward networks in stages. Instead of directly experimenting to fit a desired mapping, it makes use of the network layers to fit the residual mapping [24]. These shortcut connections give a reasonable solution to the degradation problem without new complexity. ResNet stacks the blocks on top of each other (figure 17) and each has two 3x3 convolution layers. At a fixed specific interval, the number of filters get doubled and down sampled spatially using a stride of 2. The network learns the mapping from  $x \rightarrow F(x) + G(x)$  unlike the straight-forward network where the mapping is of the form  $x \rightarrow F(x)$  [22]. "...[W]hen the dimension of the input  $x$  and output  $F(x)$  is same, the function  $G(x) = x$  is an identity function and the shortcut connection are called identity connection. The identical mapping is learned by zeroing out the weights in the intermediate layer during training, since it's easier to zero out the weights than push them to one" [22].

When the stride length is 1 in the convolution layers in-between, there may be cases when the dimensions of  $F(x)$  are different from  $x$ . In such cases, instead of identity connection, projection connections are implemented. The identity function  $G(x)$  as explained above, modifies the dimensions of input value  $x$  to be same as that of output  $F(x)$  [22].

Figure 17 and 18 [46] below show the overall architecture of ResNet [21] and it is further explained why ResNet works despite an increased number of layers.

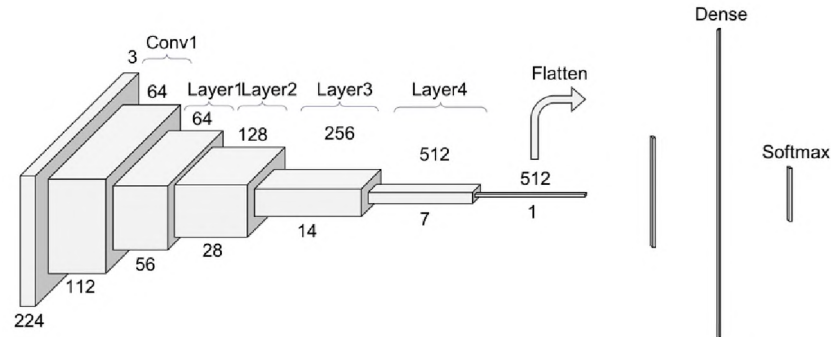


Figure 17: ResNet architecture [46]

The first layer in ResNet is conv1 consisting of convolution + batch normalization + max pooling operation [46]. For the convolution operation, feature map size is 64 and the kernel size is 7. There is a padding with zeros 3 times on each dimension. From figure 19, we can infer that the output size will be a 112x112 volume for that operation. The resultant output volume will be 112x112x64, as each of the 64-convolution filter contributes to one channel in the output volume [46].

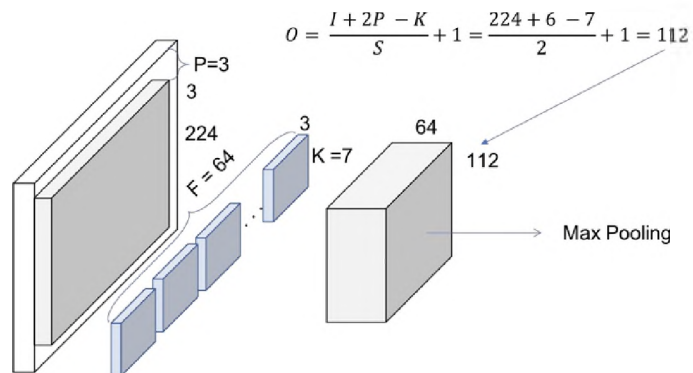


Figure 19: Conv1- Convolution [46]



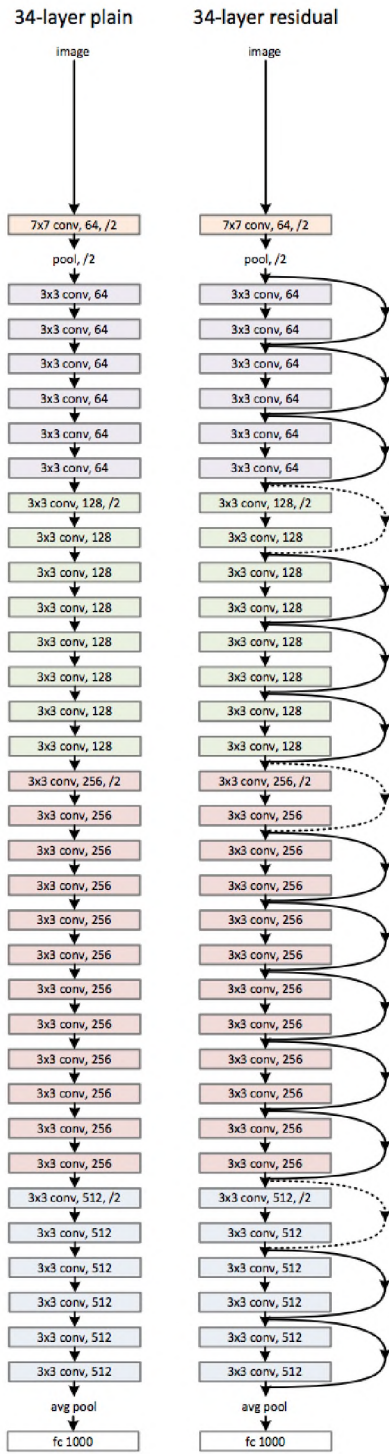


Figure 18: ResNet architecture [21]

Next is the batch normalization, which is an operation done for each element without modifying the size of the volume. Finally, there is a 3x3 Max Pooling operation with a stride of 2. As the input volume is first padded, the final volume has the desired dimensions [46].

There are several blocks in every layer of ResNet. As depth in ResNets is achieved “...[b]y increasing the number of operations within a block, the number of total layers remains the same. An ‘operation’ here refers to a convolution, batch normalization and ReLU activation to an input, except the last operation of a block that does not have the ReLU” [46]. This process is stretched to the entire layer as shown in Figure 20, therefore we have the  $[3 \times 3, 64] \times 3$  times the process within each layer [46].

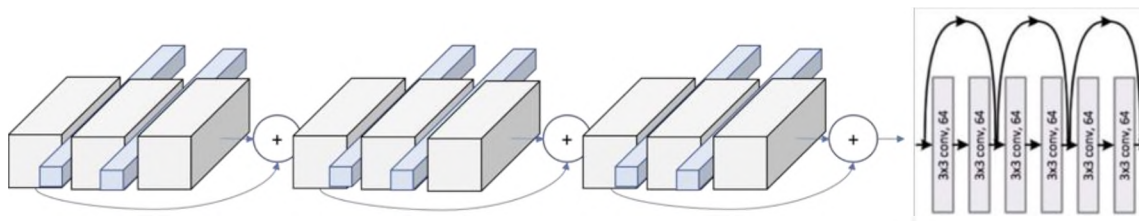


Figure 20: Layer 1 [46]

### 5.3 Comparing ResNet with AlexNet and VGG Architectures

There have been several improvements in CNN with respect to structural reformation, parameter optimization and regularization to make it scalable to large and complex problems. CNN based applications of image identification became widespread after the extraordinary results given by AlexNet on ImageNet dataset [11]. The researchers shifted their focus from layer-wise visualization of features to extraction of features at low spatial resolution [37]. One such successful architecture was VGG [37]. In 2015, the concept of skip connections introduced by ResNet [21] for the training of deep CNNs became a

success when it became possible to have CNN model with layers as deep as 152. The researches were then more oriented towards improved layer architectural design rather than parameter optimization and connections readjustment.

In AlexNet, there were seven layers for feature extraction stages with 650k units and 60 million parameters to make CNN pertinent for images of varied categories. AlexNet made some adjustments to use large size filters (11x11 and 5x5) at the initial layers as compared to prior work. Due to high efficiency in training the model of AlexNet, it had a major role to play in the new era of CNNs and started a revolution in research and architectural advancements of CNNs. AlexNet has input images of dimension 227x227x3 images. The first layer (CONV1) has 96 (11x11) filters applied at stride 4 resulting in an output volume size of 55. The total output volume of 55x55x96 gives 35k parameters, calculated as  $(11 \times 11 \times 3) \times 96 = 35K$ . The second layer (POOL1) has 3x3 filters applied at stride 2. The output volume size is 27 and the total output volume is 27x27x96.

VGG model, which was proposed by Oxford Robotics Institute was made with a depth of 19 layers. VGG has a stack of 3x3 filters replacing the 11x11 and 5x5 of AlexNet, with stride 1 and 2x2 max pool with stride 2 and demonstrated through rigorous experiments that the simultaneous positioning of 3x3 filters can have the same effect of the larger size of filters. Use of the small size filters also help in reducing the number of parameters required for computation and have a low complexity [11]. VGG places 1x1 filters in between the convolutional layers to regularize the complexity of network and additionally learn a linear combination of the resultant feature maps [11]. For tuning purposes of the network, max pooling layer is placed after the convolutional layer in VGG and padding was performed to maintain the spatial resolution. Despite small size filters, VGG still has

a high computational cost as it uses about 140 million parameters. But as we go deeper in the VGG network, more non-linearities are introduced in the data and parameters are decreased per layer.

ResNet, which is by far the most efficient CNN architecture with a depth of 152 layers and researches are still going on to extend it up to thousand layers is more than 20 times deeper than AlexNet and 8 times deeper than VGG. ResNet clearly displayed less computational complexity than previously proposed architectures [11]. ResNet got 3.5 of 7% top 5 error with 152-layer model for ImageNet. It gained 28% improvement on the COCO image recognition benchmark dataset. ResNet is made up of stacked residual blocks as explained in section 5.2.1 where every residual block in ResNet has two 3x3 conv layers. The number of filters is doubled periodically and with a stride of 2, it is down-sampled spatially. There is an additional convolution layer at the beginning and no fully connected layers at the end. For networks that are more than 50 layers deep, ResNet uses bottleneck layers similar in order to improve efficiency. Within each block, it has 1x1 convolution filter, that first projects it down to a smaller depth. For example, if there is a 28x28x256 input, its projecting depth is taken down to get 28x28x64 using 64 filters. Next, the 3x3 convolution is applied only on 64 feature maps. Again, 1x1 convolution is applied which projects the depth back up to 256. This makes the total number of operations as roughly 600k. ResNet also uses batch normalization after every conv layer.

Figure 21 below shows the winners of the ImageNet Large Scale Visual Recognition challenge (ILSVRC) challenge over the years where ResNet had 3.6% top 5 error rate [43].

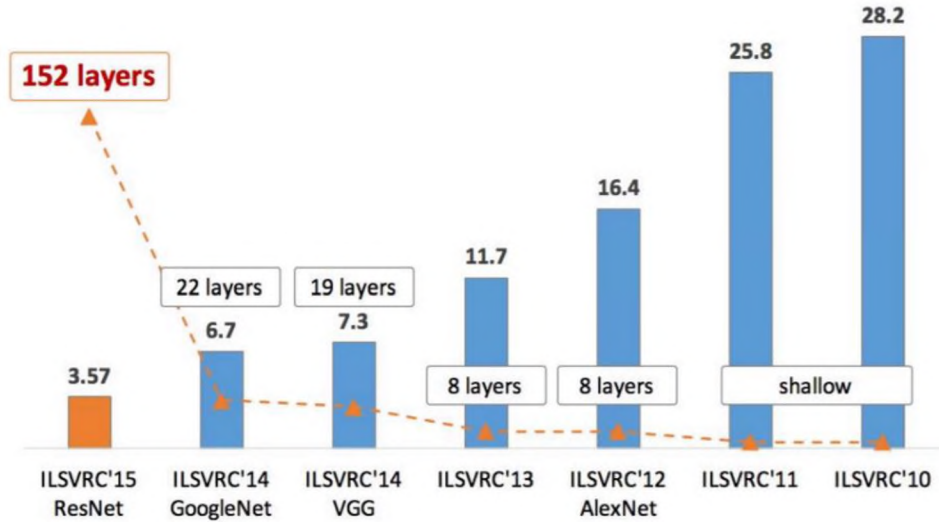


Figure 21: ILSVRC challenge error rates 2010-15 [43]

Figure 22 below compares the computational complexity of various CNN models [43]. As seen from this figure, the *x-axis* is the number of operations and the *y-axis* is the top 1% accuracy, so the higher is better. The circles become bigger with more memory usage. VGG has the highest memory consumption and the greatest number of operations are performed in VGG. AlexNet has the lowest accuracy. Even though it's relatively smaller to compute as it's a smaller network, but it's not very memory efficient. ResNet, on the other hand, is in the middle of VGG and AlexNet in terms of memory consumption and number of operations performed but has the highest accuracy [43].

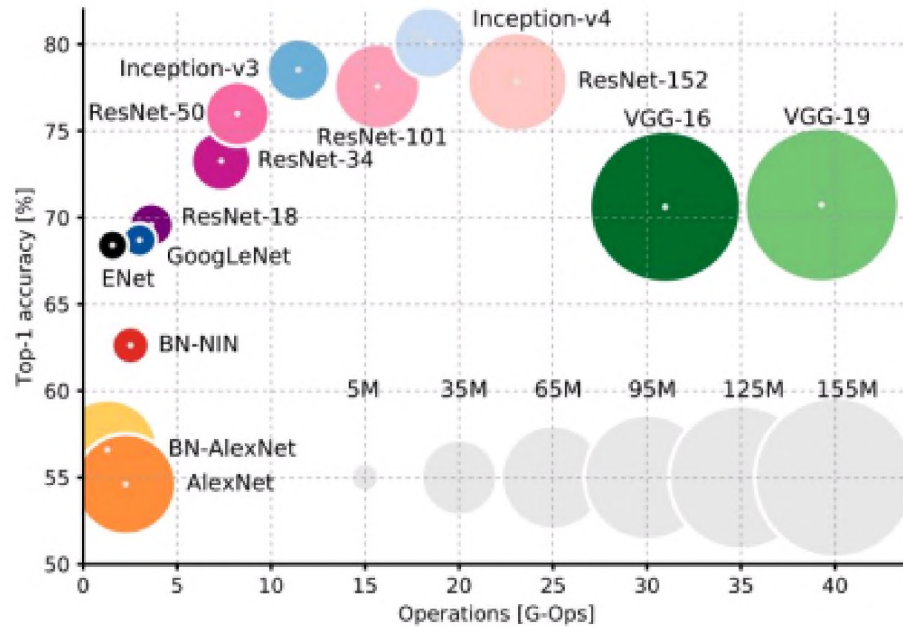


Figure 22: Computational complexity of CNN models [43]

The next chapter discusses the experimental results and evaluation for The Framework and its comparison in terms of accuracy to other models developed with similar and different CNN models.

## CHAPTER VI

### EXPERIMENTAL RESULTS AND EVALUATION

#### 6.1 Datasets used for The Framework

COCO dataset is

“...[t]he state-of-the-art in object recognition by placing the question of object recognition in the context of the broader question of scene understanding. This is achieved by gathering images of complex everyday scenes containing common objects in their natural context. Objects are labeled using per-instance segmentations to aid in precise object localization. Our dataset contains photos of 91 objects types that would be easily recognizable by a 4-year-old. With a total of 2.5 million labeled instances in 328k images, the creation of our dataset drew upon extensive crowd worker involvement via novel user interfaces for category detection, instance spotting and instance segmentation.” [13]

As mentioned in chapter II, the image dataset used to evaluate the model in The Framework is Common Objects in Context (COCO) 2014 [14]. It is provided by Microsoft and is most widely used. COCO dataset has half a million images which are divided into training, validation, and test sets. It has 82,783 images for training dataset and 40,504 images for validation dataset. For question and answers, The Framework uses VQA 1.0 dataset [32]. It also consists of human annotations for the training and validation sets where each annotation is a 10-15 words with 5-7 annotations per image. For most of experiments, training data was set to be the train set and the results are validated and reported from the validation set. The results were also taken from test set to compare them with the prior

work done. VQA 1.0 data set uses images from the COCO data set [14]. For question and answers, there are three questions per image. For every question, there are ten labelled answers by human annotators [4]. Table 1 below shows the number of questions and images available in the VQA dataset. There are 3129 answer label classes, 443757 training questions and 214354 validation questions in the data set.

<b>Mode</b>	<b>Training</b>	<b>Validation</b>	<b>Testing</b>
Images	82783	40504	81434
Questions	443757	214354	447793
Answers	443757	214354	-

Table 1: Number of questions and images in VQA dataset

## 6.2 Evaluation Metrics

The Framework is modeled as an open-ended task or multiple-choice questions, where the questions are open ended or as a multiple-choice task. For multiple-choice types of questions, simple accuracy can be used as an evaluation metric as there can be a single right choice. For open-ended questions, it is imperative that the ground truth answer and the predicted answer matches exactly. As simple accuracy metrics do not produce good results, there are various other different evaluation metrics proposed to evaluate open-ended visual question answering problems to find accuracy of different types of questions [15]. Wu-Palmer Similarity (WUPS) [24] measures the difference between the predicted answer and the ground truth based on their semantic meaning. In VQA 1.0 Dataset, there are ten ground truth answers which are annotated by human for every question [15]. The metric used for evaluation is given by [15]:



$$ACCURACY_{VQA} = \min(1, n/3)$$

where  $n$  = number of annotators that marked same answers as The Framework. Figure 23 below presents a table to compare various evaluation metrics proposed for VQA [15].

	Pros	Cons
<b>Simple Accuracy</b>	<ul style="list-style-type: none"> <li>• Very simple to evaluate and interpret</li> <li>• Works well for small number of unique answers</li> </ul>	<ul style="list-style-type: none"> <li>• Both minor and major errors are penalized equally</li> <li>• Can lead to explosion in number of unique answers, <ul style="list-style-type: none"> <li>• especially with presence of phrasal or sentence answers</li> </ul> </li> </ul>
<b>Modified WUPS</b>	<ul style="list-style-type: none"> <li>• More forgiving to simple variations and errors</li> <li>• Does not require exact match</li> <li>• Easy to evaluate with simple script</li> </ul>	<ul style="list-style-type: none"> <li>• Generates high scores for answers that are lexically related but have diametrically opposite meaning</li> <li>• Cannot be used for phrasal or sentence answers</li> </ul>
<b>Consensus Metric</b>	<ul style="list-style-type: none"> <li>• Common variances of same answer could be captured</li> <li>• Easy to evaluate after collecting consensus data</li> </ul>	<ul style="list-style-type: none"> <li>• Can allow for some questions having two correct answers</li> <li>• Expensive to collect ground truth</li> <li>• Difficulty due to lack of consensus</li> </ul>
<b>Manual Evaluation</b>	<ul style="list-style-type: none"> <li>• Variances to same answer is easily captured</li> <li>• Can work equally well for single word as well as phrase or sentence answers</li> </ul>	<ul style="list-style-type: none"> <li>• Can introduce subjective opinion of individual annotators</li> <li>• Very expensive to setup and slow to evaluate, especially for larger datasets</li> </ul>

Figure 23: Comparison of different evaluation metrics for VQA [15]

### 6.3 Data Preprocessing

The question text was first preprocessed by tokenizing the questions by checking for missing values, removing stop words and punctuations, lemmatization, splitting them into individual words and converting all the text in lower-case. The distribution of the question lengths is plotted as shown in figure 24, to pad the question tokens so that the input sequences to LSTM are of the same length [30]. The maximum question length is fixed at

15 as there are hardly any questions exceeding that length. Questions with length less than 15 are padded before being passed as an input to the embedding layer.

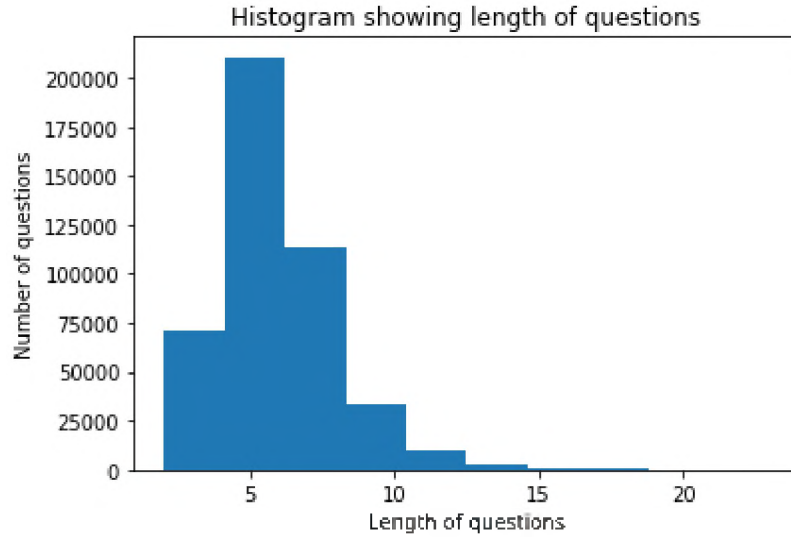


Figure 24: Distribution of the length of the questions from the training set [30]

## 6.4 Model Training

The Framework uses ResNet model with 152 layers, pretrained on the ImageNet dataset [25] for image feature extraction.  $l_2$  normalization is performed on the third dimension of depth on the last block of ResNet before the final sub-sampling layer of  $14 \times 14 \times 2048$  dimensions is considered.  $D = 300$  vector dimension is the representation of the question embeddings. The non-linearity function  $\tanh$  is applied to the question embeddings before passing them as input to LSTM. Total 512 residual channels are present. The batch size during the training is kept as 64 on total of 50 epochs. The learning rate is 0.001 and Adam optimizer is used to optimize the model. The dropout value of the convolutional layers and fully connected layers is set to 0.6.

## 6.5 Training Result

The below table 2 shows the accuracy numbers recorded for The Framework after training the model for 50 epochs with the above-mentioned configuration.

Model	Accuracy
ResNet + LSTM without attention	46.63%
ResNet + LSTM with attention model	59%

Table 2: Percentage Accuracy for The Framework

Figure 25 below is a graphical representation of the output accuracy for ResNet model with LSTM versus ResNet model with LSTM and attention model. The highest accuracy without the attention model for 50 epochs run is 46.63%. The highest accuracy recorded after combining the existing model with attention network increases to 59% for the same 50 epochs run.

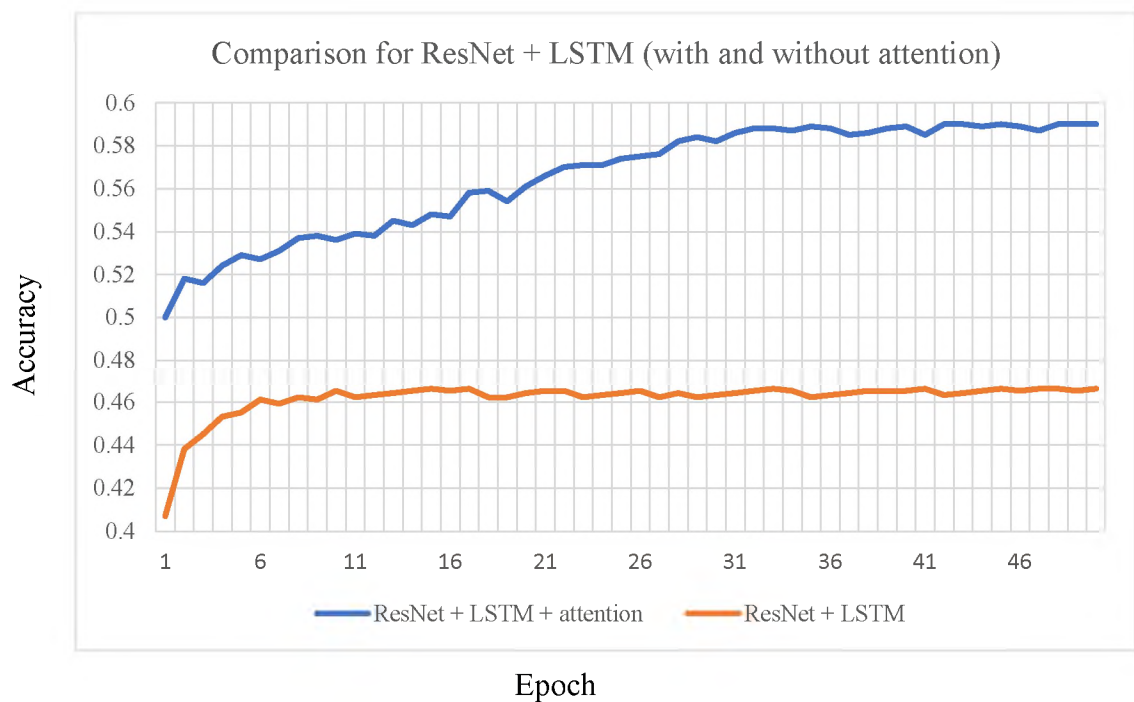


Figure 25: Comparison for ResNet + LSTM (with and without attention)

Table 3 shows a comparison of The Framework with prior researches on COCO and VQA 1.0 Dataset using a similar model of ResNet with LSTM and attention model. The authors of [20] mention that all the baseline input images were scaled while preserving aspect ratio and center cropped to 299x299 dimensions. The rest of the parameters and image dimensions are the same. The authors of [19] on the other hand developed a multiple-layer Stacked Attention Network which progressively queries an image to generate answers multiple times, with the rest of the methodology same as The Framework. The authors of [39] reasons about the question in a hierarchical fashion via a 1-dimensional CNN.

Model	Accuracy (%)
Resnet + LSTM + attention [20]	59.76
Resnet + LSTM + attention [19]	57.2
ResNet + LSTM + Question-attention [39]	54.8
The Framework (ResNet + LSTM + attention)	59

Table 3: Comparing accuracy of The Framework with other ResNet+LSTM+attention models

Figure 26 below shows the graphical representation of the accuracy comparison of The Framework compared to prior work done with attention model.

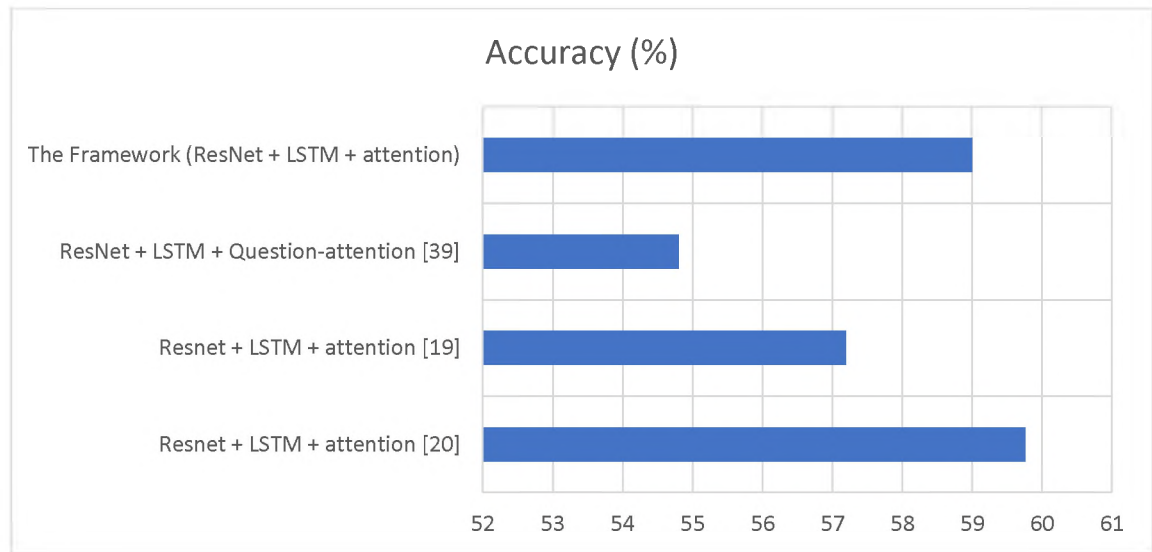


Figure 26: Comparison of The Framework to prior work done with attention model

Table 4 shows a comparison of The Framework with prior researches on COCO and VQA 1.0 Dataset in Visual Question Answering systems.

<b>Model</b>	<b>Accuracy (%)</b>
CNN + LSTM [26]	52
ResNet + LSTM [27]	51
LSTM + VGG [32]	54.1
Resnet + bytenet [15]	60
Resnet + LSTM + attention [20]	59
Resnet + LSTM [33]	58
VGG + LSTM + attention [4]	58.9
The Framework (ResNet + LSTM + attention)	59

Table 4: Comparing percentage accuracy of The Framework with prior work

Figure 27 below shows the accuracy comparison of The Framework compared to other prior work done and indicates that the accuracy with attention model proves to be amongst the better model out of all.

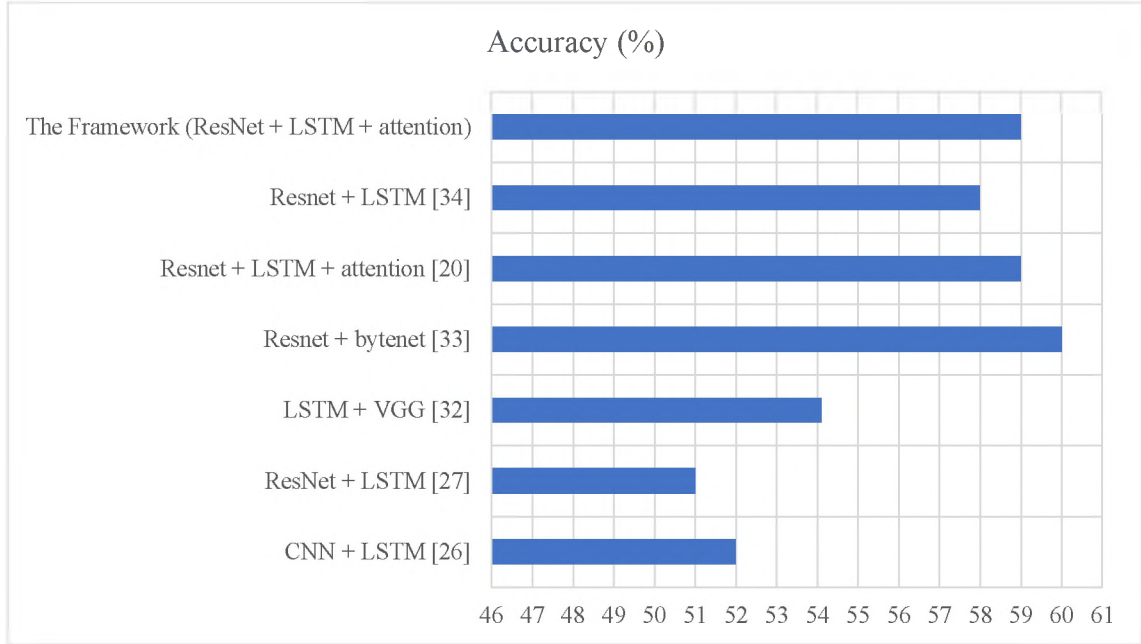


Figure 27: Comparing accuracy of The Framework with other models

The results show the comparison of The Framework with other researches and prior work done in similar area to develop a VQA system. A separate comparison of attention-based models has been presented and a separate comparison with other image identification models is done using VGG and byteNet models. The results indicate that the attention model developed in The framework proves to be amongst the better models in terms of accuracy when compared with prior work.

## **CHAPTER VII**

### **CONCLUSION AND FUTURE DIRECTION**

#### **7.1 Conclusion**

The methodologies to build a Visual Question Answering system using question attention-based deep neural network were studied. The study explored the research in building a Visual Question Answering system using deep learning algorithms based on CNN for image recognition and LSTM networks for NLP for question analysis for generating attention. The research considers the visual question answering task as a classification problem [15]. For a given input image and a textual question in natural language, the model developed in this research estimates the most likely answer from a fixed set of answers basing it and matching it from the contents of the image. The model uses a deep residual network (ResNet), an advanced CNN model to compute the image features, and LSTM, a special type of RNN to compute question embedding combined with an attention mechanism to focus on important parts of the input image features and a classifier is used to generate textual answer probabilities over a given set.

The Framework offers an effective method that can be applied to various image recognition tasks. Its challenges and future directions in the field of Visual Question Answering system were investigated and discussed. Two challenging tasks to build a



Visual Question Answering system for image identification and sentence processing were discussed with ResNet architecture for image feature extraction, and Long Short-Term Memory (LSTM) network for question processing. The attention-based method to combine the outputs from these two components were applied to get the final answer prediction in VQA.

Prior to using deep residual networks with LSTM and attention model, experiments were carried out using deep residual networks with LSTM for generating answers in Natural Language. From the results in the previous section, it was observed that using ResNet with LSTM and attention model answer generation resulted in a significant increase in the accuracy when compared to the model not using attention model. Some experiments for question processing were also performed by replacing long short-term memory unit (LSTM) with ByteNet and results were compared for accuracies.

## **7.2 Future Work**

By building The Framework, the various concepts, advantages as well as limitations of multiple CNN models were explored, which is essential to leverage its potential with the goal of developing strong knowledge-base and skill set in machine learning and deep learning networks. The Framework can also be extended to answer open-ended questions on multiple images at the same time. Having said that, the possibility of extending The Framework to video question answering system can also be explored, whose application can be seen in the field of video surveillance systems and crowd surveillance systems.

## BIBLIOGRAPHY

- [1] HUBEL DH, WIESEL TN. Receptive fields of single neurones in the cat's striate cortex. J Physiol. 1959 Oct;148(3):574-91. PubMed PMID: 14403679; PubMed Central PMCID: PMC1363130. DOI=  
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1363130/>
- [2] Noh Hyeonwoo Seo, Paul Hongsuck, and Han, Bohyung. (2016). Image Question Answering Using Convolutional Neural Network with Dynamic Parameter Prediction. 30-38. 10.1109/CVPR.2016.11.
- [3] Chen, Kan & Wang, Jiang & Chen, Liang-Chieh & Gao, Haoyuan & Xu, Wei & Nevatia, Ram. (2015). ABC-CNN: An Attention Based Convolutional Neural Network for Visual Question Answering.
- [4] Yang, Zichao & He, Xiaodong & Gao, Jianfeng & Deng, li & Smola, Alex. (2016). Stacked Attention Networks for Image Question Answering. 21-29. 10.1109/CVPR.2016.10.
- [5] "CS231n Convolutional Neural Networks for Visual Recognition." Github.io, 2012, [cs231n.github.io/convolutional-networks/#pool](https://github.com/jbrownlee/DLbooks/blob/master/convnets/cs231n_convnets.ipynb).
- [6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM 60, 6 (May 2017), 84-90. DOI: <https://doi.org/10.1145/3065386>
- [7] Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. ACL.

- [8] Chidambar, A., Karambelkar, S., Prabhu, K., Darade, V. and Sonawani, S.  
(2018). Image Question Answering: A Review.  
<https://www.irjet.net/archives/V5/i6/IRJET-V5I6540.pdf>.
- [9] “ResNet, AlexNet, VGGNet, Inception: Understanding Various Architectures of Convolutional Networks - CV-Tricks.Com.” CV-Tricks.Com, 9 Aug. 2017, [cv-tricks.com/cnn/understand-resnet-alexnet-vgg-inception/](http://cv-tricks.com/cnn/understand-resnet-alexnet-vgg-inception/).
- [10] Ordonez, Juanita. “What is in that picture ? Visual Question Answering System.” (2017).
- [11] Khan, Asifullah, et al. A Survey of the Recent Architectures of Deep Convolutional Neural Networks. [arxiv.org/ftp/arxiv/papers/1901/1901.06032.pdf](http://arxiv.org/ftp/arxiv/papers/1901/1901.06032.pdf).
- [12] Kelly, Adam. “Create COCO Annotations From Scratch.” Immersive Limit, Immersive Limit, 11 Jan. 2019, <http://www.immersivelimit.com/tutorials/create-coco-annotations-from-scratch>.
- [13] Lin, Tsung-Yi, et al. Microsoft COCO: Common Objects in Context. [arxiv.org/pdf/1405.0312.pdf](http://arxiv.org/pdf/1405.0312.pdf).
- [14] Cavaioni, Michele. “DeepLearning Series: Convolutional Neural Networks.” Medium, Machine Learning bites, 23 Feb. 2018, [medium.com/machine-learning-bites/deeplearning-series-convolutional-neural-networks-a9c2f2ee1524](https://medium.com/machine-learning-bites/deeplearning-series-convolutional-neural-networks-a9c2f2ee1524).
- [15] Koduri, Lavanya Abhinaya. A Convolutional Neural Network Based Approach For Visual Question Answering. doi:10.31979/etd.g2w5-abud.

- [16] Pooling or subsampling layer. “Deep Learning Essentials.” O’Reilly | Safari, 2019, [www.oreilly.com/library/view/deep-learning-essentials/9781785880360/17c0dae4-8c97-4501-a179-b17d62fd38cd.xhtml](http://www.oreilly.com/library/view/deep-learning-essentials/9781785880360/17c0dae4-8c97-4501-a179-b17d62fd38cd.xhtml).
- [17] Zolaktaf, Nasim. Recurrent Neural Networks. 2016, [www.cs.ubc.ca/labs/lci/mlrg/slides/rnn.pdf](http://www.cs.ubc.ca/labs/lci/mlrg/slides/rnn.pdf).
- [18] Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). Long Short-term Memory. Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735.
- [19] Rister, Blaine. Image Captioning with Attention. [cs231n.stanford.edu/reports/2016/pdfs/362\\_Report.pdf](http://cs231n.stanford.edu/reports/2016/pdfs/362_Report.pdf).
- [20] Kazemi, Vahid, and Ali Elqursh. Show, Ask, Attend, and Answer: A Strong Baseline For Visual Question Answering. [arxiv.org/pdf/1704.03162.pdf](http://arxiv.org/pdf/1704.03162.pdf).
- [21] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun, “Deep Residual Learning for Image Recognition,” arXiv:1512.03385.
- [22] Aqeel Anwar. “Difference between AlexNet, VGGNet, ResNet and Inception.” Medium, Towards Data Science, 7 June 2019, [towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaecccc96](https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaecccc96).
- [23] Li, Tianxiao, et al. “Inadequate Expansion Lead to Delayed Enterprise Stent Migration.” Neurology India, vol. 65, no. 2, 2017, p. 377, doi:10.4103/neuroindia.ni\_946\_15.
- [24] M. Malinowski and M. Fritz, “A multi-world approach to question answering about real world scenes based on uncertain input,” in Advances in Neural Information Processing Systems (NIPS), 2014.

- [25] Fei-Fei, L., et al. “ImageNet: Constructing a Large-Scale Image Database.” *Journal of Vision*, vol. 9, no. 8, Mar. 2010, pp. 1037–1037, doi:10.1167/9.8.1037.
- [26] avisingh599. “Avisingh599/Visual-Qa.” GitHub, 12 June 2017, [github.com/avisingh599/visual-qa](https://github.com/avisingh599/visual-qa).
- [27] DenisDsh. “DenisDsh/VizWiz-VQA-PyTorch.” GitHub, 17 Oct. 2018, [github.com/DenisDsh/VizWiz-VQA-PyTorch](https://github.com/DenisDsh/VizWiz-VQA-PyTorch).
- [28] Xu, Kelvin, et al. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. [arxiv.org/pdf/1502.03044.pdf](https://arxiv.org/pdf/1502.03044.pdf).
- [29] Gu, Jiuxiang, et al. Stack-Captioning: Coarse-to-Fine Learning for Image Captioning. [arxiv.org/pdf/1709.03376.pdf](https://arxiv.org/pdf/1709.03376.pdf).
- [30] Nithin Rao. “Visual Question Answering—Attention and Fusion Based Approaches.” Medium, Medium, 30 Apr. 2019, [medium.com/@nithinraok\\_/visual-question-answering-attention-and-fusion-based-approaches-ebef62fa55aa](https://medium.com/@nithinraok_/visual-question-answering-attention-and-fusion-based-approaches-ebef62fa55aa).
- [31] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus, “Simple baseline for visual question answering,” *arXiv preprint arXiv:1512.02167*, 2015
- [32] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, “VQA: Visual Question Answering”, in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2425-2433, 2015.
- [33] Malinowski, Mateusz, et al. “Ask Your Neurons: A Deep Learning Approach to Visual Question Answering.” *International Journal of Computer Vision*, vol. 125, no. 1–3, Aug. 2017, pp. 110–35, doi:10.1007/s11263-017-1038-2.

- [34] Hassantabar, Shayan. Visual Question Answering: Datasets, Methods, Challenges and Oppurtunities.  
[www.cs.princeton.edu/courses/archive/spring18/cos598B/public/projects/LiteratureReview/COS598B\\_spr2018\\_VQAreview.pdf](http://www.cs.princeton.edu/courses/archive/spring18/cos598B/public/projects/LiteratureReview/COS598B_spr2018_VQAreview.pdf).
- [35] Shayan and Prem, “Visual Question and Answering”, 26-Mar.-2018. [Online]. Available:  
<https://www.cs.princeton.edu/courses/archive/spring18/cos598B/public/outline/Visual%20Question%20and%20Answering.pdf>.
- [36] M. Malinowski and M. Fritz. Towards a visual turing challenge. arXiv preprint arXiv:1410.8027, 2014.
- [37] Simonyan, Karen, and Andrew Zisserman. Published as a Conference Paper at ICLR 2015 VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION. [arxiv.org/pdf/1409.1556.pdf](http://arxiv.org/pdf/1409.1556.pdf).
- [38] A. Mikulik, O. Chum, and J. Matas. Image retrieval for online browsing in large image collections. In 6th International Conference, SISAP, 2013.
- [39] Lu, Jiasen, et al. Hierarchical Question-Image Co-Attention for Visual Question Answering. [arxiv.org/pdf/1606.00061.pdf](http://arxiv.org/pdf/1606.00061.pdf).
- [40] Kevin J Shih, Saurabh Singh, and Derek Hoiem. Where to look: Focus regions for visual question answering. In CVPR, 2016.
- [41] Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic memory networks for visual and textual question answering. In ICML, 2016.

- [42] Huijuan Xu and Kate Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. arXiv preprint arXiv:1511.05234, 2015.
- [43] “Stanford University CS231n: Convolutional Neural Networks for Visual Recognition.” Stanford.Edu, 2019, cs231n.stanford.edu.
- [44] “Unsupervised Feature Learning and Deep Learning Tutorial.” Stanford.Edu, 2019, ufldl.stanford.edu/tutorial/supervised/MultiLayerNeuralNetworks/.
- [45] “Introduction to Deep Learning: Home Page.” Princeton.Edu, 2016, www.cs.princeton.edu/courses/archive/spring16/cos495/.
- [46] Ruiz, Pablo. “Understanding and Visualizing ResNets.” Medium, Towards Data Science, 8 Oct. 2018, towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8.
- [47] “An Intuitive Explanation of Convolutional Neural Networks.” The Data Science Blog, the data science blog, 29 May 2017, ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/.
- [48] Wikipedia Contributors. “Convolution.” Wikipedia, Wikimedia Foundation, 29 Apr. 2019, en.wikipedia.org/wiki/Convolution.
- [49] Wikipedia Contributors. “Softmax Function.” Wikipedia, Wikimedia Foundation, 29 Nov. 2019, en.wikipedia.org/wiki/Softmax\_function.
- [50] Khuong, Ben. “The Basics of Recurrent Neural Networks (RNNs).” Medium, Towards AI, 24 June 2019, medium.com/towards-artificial-intelligence/whirlwind-tour-of-rnns-a11effb7808f.

- [51] Wikipedia Contributors. “Backpropagation through Time.” Wikipedia, Wikimedia Foundation, 9 Nov. 2019, [en.wikipedia.org/wiki/Backpropagation\\_through\\_time](https://en.wikipedia.org/wiki/Backpropagation_through_time).
- [52] Wu J, Hu Z, Mooney RJ. Joint Image Captioning and Question Answering. arXiv.org. <https://arxiv.org/abs/1805.08389>. Published 2018.
- [53] VQA: Visual Question Answering. Visualqa.org. [https://visualqa.org/vqa\\_v1\\_download.html](https://visualqa.org/vqa_v1_download.html). Published 2017.
- [54] Meng, Weizhi. Scholars’ Mine Scholars’ Mine Masters Theses Student Theses and Dissertations A Sentence-Based Image Search Engine A Sentence-Based Image Search Engine.2015, [scholarsmine.mst.edu/cgi/viewcontent.cgi?article=8473&context=masters\\_theses](http://scholarsmine.mst.edu/cgi/viewcontent.cgi?article=8473&context=masters_theses).



**APPENDIX**  
**SYSTEM CONFIGURATION**

Specification	Value
Machine Name	Bigdata Windows Server 2016, Ubuntu bigdata1 server, Ubuntu bigdata2 server
Operating System	Ubuntu 16.04
Processor	Intel(R) Core (TM) i7-5820K CPU @ 3.30GHz
Graphic Processing Unit	GP102 [GeForce GTX 1080 Ti]
RAM (Memory)	64 GB
Programming Language	Python 2.7
Tensorflow	1.3.0
Other Libraries	Theano (Numerical computation), Keras (Neural Network Library), OpenCV (Computer Vision), DLib (Data mining and machine learning techniques)