DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

Institute of Science and Technology

IMAGE CAPTION GENERATOR USING CNN AND RNN

A PROJECT REPORT

Submitted to

Department of Computer Science and Information Technology

DWIT College

In partial fulfillment of the requirements for the Bachelor's Degree in Computer Science and Information Technology

Submitted by

Aawaj Shrestha

Sagar Shrestha

December, 2020

DWIT College

DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

SUPERVISOR'SRECOMENDATION

I hereby recommend that this project prepared under my supervision by AAWAJ SHRESTHA and SAGAR SHRESTHA entitled **"IMAGE CAPTION GENERATOR USING CNN AND RNN"** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for the evaluation.

……

Mr. Bhas Raj Pathak

Senior Lecturer

Deerwalk Institute of Technology

DWIT College

DWIT College

DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

LETTER OF APPROVAL

This is to certify that this project prepared by AAWAJ SHRESTHA and SAGAR SHRESTHA entitled **"IMAGE CAPTION GENERATOR USING CNN AND RNN"** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

…… Mr. Bhas Raj Pathak Senior Lecturer DWIT College	…… Dr. Subarna Shakya Professor IOE, Tribhuvan University

ACKNOWLEDGEMENT

We would like to express our gratitude to our supervisor, Mr. Bhas Raj Pathak, Senior Lecturer, Department of Computer Science for his valuable guidance in completion of our final year project titled IMAGE CAPTION GENERATOR USING CNN AND RNN. His knowledge and insights into the research topic proved crucial in our completion of the project.

We would also like to thank our family and friends who gave us encouragement to complete our project within the limited time frame.

Aawaj Shrestha

Roll No.: 5-2-1175-5-2016

Sagar Shrestha

Roll No.: 5-2-1175-35-2016

Date: - December, 2020

ABSTRACT

In the past few years, the problem of generating descriptive sentences automatically for images has garnered a rising interest in natural language processing and computer vision research. Image captioning is a fundamental task which requires semantic understanding of images and the ability of generating description sentences with proper and correct structure. In this study, we propose a hybrid system employing the use of multilayer Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) models to generate vocabulary describing the images and a Long Short-Term Memory (LSTM) to accurately structure meaningful sentences using the generated keywords. We showcase the efficiency of our proposed model using the Flickr8K and Flickr30K datasets and show that their model gives superior results. We discuss the foundation of the techniques to analyze their performances, strengths and limitations. We also discuss the datasets and the evaluation metrics popularly used in deep learning based automatic image captioning.

Keywords: *Caption Generator, Image Recognition, Long Short-Term Memory.*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATIONS

RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
CNN	Convolutional Neural Network
ConvNets	Convolutional Neural Network
RGB	Red Green Blue
VGG16	Visual Geometry Group
BLEU	Bilingual Evaluation Understudy
COCO	Common Objects in Context
API	Application Programming Interface
GPU	Graphics Processing Unit
R-CNN	Region-Convolutional Neural Network
NL	Natural Language
NLP	Natural Language Processing
CPM	Critical Path Method
WBS	Work Breakdown Structure

CHAPTER 1: INTRODUCTION

1.1 Overview

Every day, we encounter a large number of images from various sources such as the internet, news articles, document, diagrams and advertisements. These sources contain images that viewers would have to interpret themselves. Most images do not have a description, but the human can largely understand them without their detailed captions. However, machine needs to interpret some form of image captions if humans need automatic image captions from it. Image captioning is important for many reasons. For example, they can be used for automatic image indexing. Image indexing is important for Content-Based Image Retrieval (CBIR) and therefore, it can be applied to many areas, including biomedicine, commerce, military, education, digital libraries, and web searching. Social media plat forms such as Facebook and Twitter can directly generate descriptions from images. The descriptions can include where we are (e.g., beach, cafe), what we wear and importantly what we are doing there.

There are basically two main approaches to Image Captioning: bottom-up and top-down. Bottom-up approaches generate items observed in an image, and then attempt to combine the items identified into a **c**aption [1]**.** Top-down approaches attempt to generate a semantic representation of an image that is then decoded into a caption using various architectures, such as recurrent neural networks [2] . The latter approach follows in the footsteps of recent advances in statistical machine translation, and the state-of-the-art models mostly adopt the top-down approach.

Our approach draws on the success of the top-down image generation models listed above. We use a deep convolutional neural network to generate a vectorized representation of an image that we then feed into a Long-Short-Term Memory (LSTM) network, which then generates captions. Figure 1 provides the broad framework for our approach.

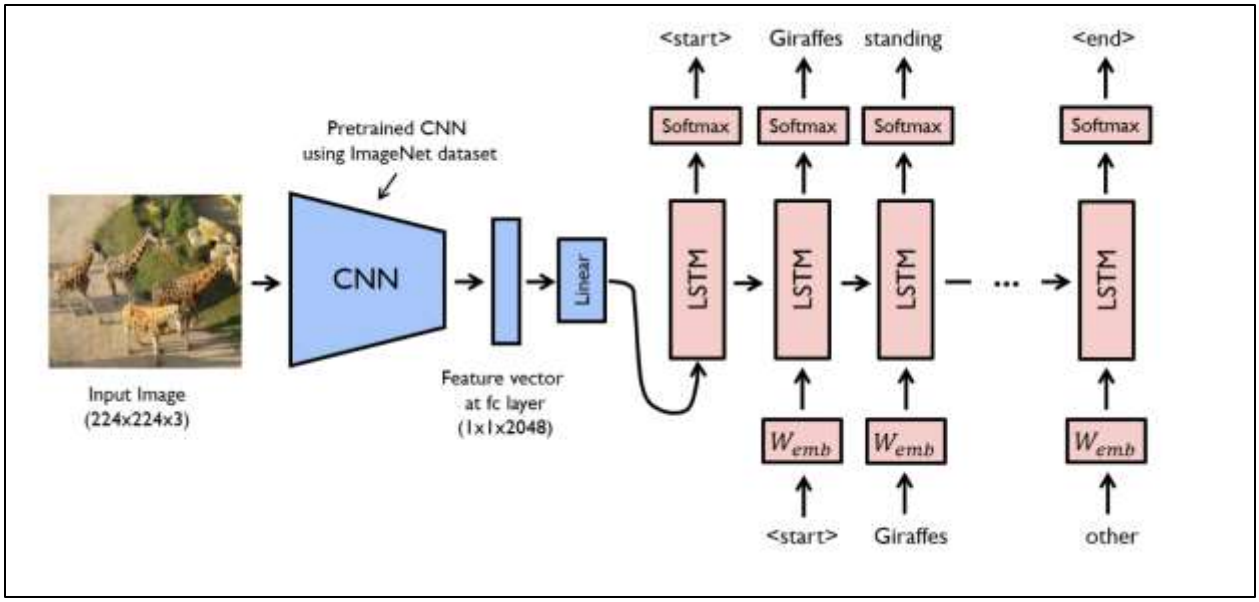


Figure 1- CNN LSTM Architecture [3]

CNN-LSTM architecture, use a deep convolution neural network to create a semantic representation of an image, which we then decode using a LSTM network. All LSTMs share the same parameter. The vectorized image representation is fed into the network, followed by a special start of sentence token. The hidden state produced is then used by the LSTM predict/generate the caption for the given image.

1.2 Background and Motivation

In this section, we describe relevant background on recurrent neural networks and image caption generation. Recently, several methods have been experimented with for automatic image caption generation [1]. we propose learning a mapping between images, meanings and captions using a graphical model based on human-engineered features. The pioneering use of neural networks for image caption generation was suggested by the multi-model pipeline [3] **,** which demonstrated that neural networks could decode image representations from a CNN encoder and that also showed that the resulting hidden dimensions and word embedding contained semantic meaning (i.e. ”image of a blue car” - ”blue” + ”red” produces vectors close to that produced by ”image of a red car”).

One primary motivation of computational visual recognition models is to emulate remarkable human capability to comprehend visual scenes and extract detailed information from them with astonishing accuracy [4]. Many sophisticated models have been developed to extract visual information from images based on visual categorization of objects in the images [5] . The visual recognition procedures thus pursued in most cases are demanding both in terms of computational complexity and obtaining desired accuracy. One popular endeavor of the visual recognition modeling is to extract concise natural language description of an image, i.e., to generate a single sentence representing an image most faithfully by reducing the complexities [6] . Figure 2 depicts an example where the image has been utilized to extract a single sentence dialect from visual data

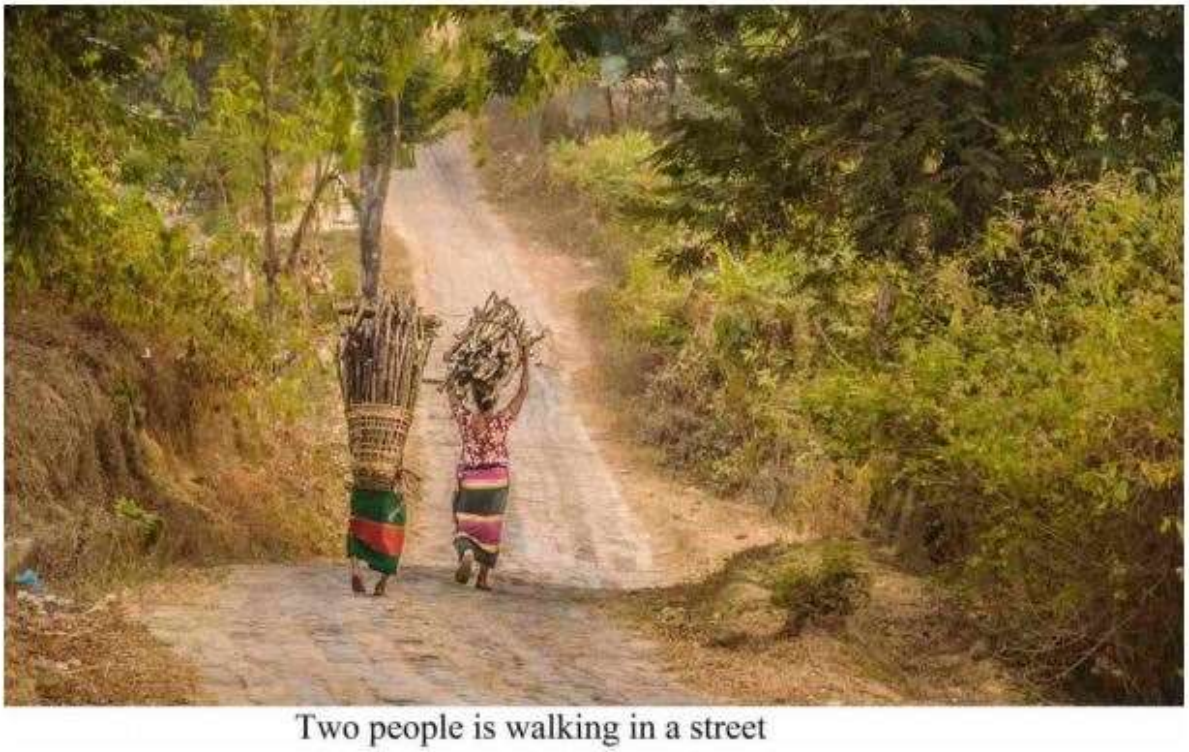


Figure 2- Motivation figure illustrating extraction of simple natural language description from visual data

1.3 Problem Statement

The model framework adopted is to analogous the recent successful approaches in statistical machine translation. Using an encoder RNN, these models learn an expressive representation of the original sentence, and use another recurrent neural network to decode that representation in the

target language. The advantages of using RNN and the model architecture are the ability to handle sequences of arbitrary length, and more importantly, the end-to-end maximization of the joint probability of the original and target sentence, which have produced state-of-the-art results in machine translation [7].

1.4 Objective of the Project

The main objective of this project is

- To develop a hybrid model for two benchmark datasets: Flickr8k and Flickr30k, Also, change in layer size, learning rate, optimization for achieving the better accuracy.
- The main objective of CNN is to scan the predefined model and find patterns in images to recognize objects, faces, and scenes which is send to LSTM in the form of thought vector.
- The main objective of RNN is to implement temporal or sequential information such that it uses data points in a sequence to make a better production.

1.5 Scope of the Project

The project consists of generating a caption for the given image. The generated caption of the image will be displayed in a web-based system with GUI Interface. The system will generate and display the caption based on the image provided by the user.

1.6 Outline of the Report

The report is organized as follows:

Preliminary Section: This section consists of the title page, abstract, table of contents and list of figures and tables.

Introduction Section: In this section, the background of the project, problem statement, its objectives and scope are discussed.

Requirement and Feasibility Analysis Section: Literature review, Requirement analysis, and feasibility analysis makes the bulk of this section.

System Design Section: The section consists of the methodology that was implemented in the project and the system design as well.

Implementation and Testing Section: The section consists of the methodology that was implemented in the project and the system design as well.

Conclusion and Recommendation: The section consists of the final findings and the recommendations that can be worked on in order to improve the project.

CHAPTER 2: LITERATURE REVIEW

The image captioning problem and its proposed solutions have existed since the advent of the Internet and its widespread adoption as a medium to share images. Numerous algorithms and techniques have been put forward by researchers from different perspectives.

Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, Xiaodong He, "AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks" [8] and Shikhar Sharma, Dendi Suhubdy, Vincent Michalski, Samira Ebrahimi Kahou, Yoshua Bengio, "ChatPainter: Improving Text to Image Generation using Dialogue" [9] ,we see that they were mainly focused on CNN features of their paper. They used a deep attention model with a combination of LSTM and AttGAN model and showed that how a better image can be created using MS COCO dataset. They also utilized Vis-Dial exchanges alongside MS COCO dataset inscriptions to create pictures. But, their experiment only CUB and MS COCO dataset and didn't use another dataset which couldn't show the better performance. Also, if we see the paper of In the meantime, if we discuss about the paper of Richard Socher, Andrej Karpathy, Quoc V. Le*, Christopher D. Manning, Andrew Y. Ng, " Grounded Compositional Semantics for Finding and Describing Images with Sentences" [10] ,we see that they based on DT-RNN model for generating text from the image region. They also focused on using semantic embedding system, and showed that how neural network can works and detect images region. They used the dataset which contains 1000 pictures, each with 5 sentences. For example, protest recognition and picture division, to foresee answers to basic inquiries regarding pictures. They took after RNN model and LSTM. They utilized the dataset of COCO-QA, DAQUAR. They utilized the VGG-16 design for its cutting-edge Performance but, their result of model was so poor and that was only 0.27.

Finally, if we follow the paper of Neural Talk 1, Andrej Karpathy et. Introduced "Deep Visual-Semantic Alignments for Generating Image Descriptions", CVPR, 2015 [11] ,we see they mainly focused on how to recognition an image and give an output in text (txt) format using BRNN and attention mechanism based model. They presented a model that generates natural language

descriptions of images and their regions. In this paper both CNN and RNN, Bidirectional Recurrent Neural Network (BRNN) with attention mechanism-based model used. They used a fixed vocabulary size which is 10,000 and use 5 sentences for each image. These papers are state of art of our project. They used three benchmark datasets: Flickr8K, Flickr30K, and MSCOCO.

Our proposed hybrid RNN model is a field of study that seeks to provide knowledge and learning to machines through data. We always use CNN for classification of whole dataset. A convolutional neural network (CNN) mainly works for image classification. But, R-CNN works for region and object detection. R-CNN is a hybrid model which is always used for region and object detections [12].

CHAPTER 3: REQUIREMENT AND FEASIBILITY ANALYSIS

3.1 Requirement Analysis

3.1.1 Functional Requirement

The functional requirements of this project are as follows:

- The user shall upload one image of which the caption is to be generated.
- The system shall generate the caption of the image and display it to the user.

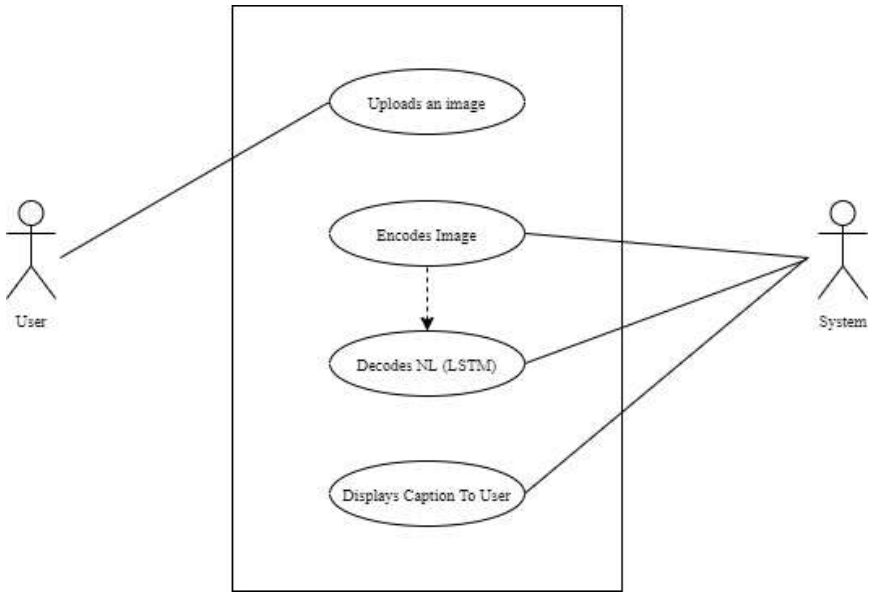


Figure 3- Use-case diagram of Image Caption Generator

Figure 3 shows the use case diagram of the caption generator system developed in this project. The user performs the solitary action by uploading an image. Then, the system **Encodes Image** using RNN and CNN whereas **Decodes NL** using LSTM. Once the NL is decoded the resulted caption is provided to the user.

3.1.2 Non-Functional Requirement

The non-functional requirements of the project are as follows:

- The system must display accuracy caption based on probability score.
- The system must have easy to use user interface.
- The system must generate the caption in the absence of internet connection.
- The system response time must be less than 0.6 seconds.

3.2 Feasibility Analysis

After gathering of the required resources, whether the completion of the project with the gathered resource is feasible is checked using the following feasibility analysis.

3.2.1 Technical Feasibility

In this project, the system was implemented as a web-based application and it runs in systems with MacOS, Windows or Linux operating systems. The computer system must include a web browser to access the system.

The primary programming language selected to build the system was Python (version 3.6.5), which is an open-source programming language. The programming language was selected due its ease of use and my experience in working with the programming language.

Keras framework was used in building the language model architecture. Keras is a high-level machine learning library built on top of TensorFlow library which allows its users to build a machine learning architecture quickly. Keras was chosen considering the limited time frame for completion of the project.

To train the language model, Google's free Google Collaboratory service was used. Google Collaboratory is a free Jupiter notebook environment that requires no setup and runs entirely in the cloud. Google Collaboratory was used to speed up the training of the language model with the free GPU service provided by it.

The tools, systems, modules and libraries needed to build the system are all open source, freely available and are easy to use. Hence, the project was determined technically feasible.

3.2.2 Economic Feasibility

The expenses incurred in the project are mostly indirect expenses. We used our personal devices to build the system and used personal internet subscription to do the research online. Moreover, training the language model, which is computationally expensive and time consuming, was done using Google's free Google Collaboratory service.

3.2.3 Operational Feasibility

The system built in the project follows a client-server architecture. The web-based application can operate in a system having Windows, Linux or MacOS. The user-interface can be accessed remotely or from within the same system where the application is hosted. The resource-intensive part of the project was training the language model. However, it is only a one-time process and hence need not be carried out when using the system. Considering the above cases, the project is deemed operationally feasible.

3.2.4 Schedule Feasibility

The schedule feasibility analysis is carried out using the CPM method. With CPM, critical tasks were identified and interrelationship between tasks were identified which helped in planning that defines critical and non-critical tasks with the goal of preventing time-frame problems and process bottlenecks. The CPM analysis was carried out as follows:

First, the activity specification table with WBS is constructed as shown in the table:

Table 1- Activity specification with WBS

Activity	Time (Days)	Predecessor
Data Collection and Preprocessing (A)	20	-
Research on Previous work and algorithms to implement (B)	40	-
Building the caption generator model (C)	3	A, B

User Interface Design (D)	5	C
System Testing €	1	C
Documentation (F)	12	A, B, C, E

Then, identification of critical path and estimates for each activity are analyzed with the help of a network diagram, which is based on Table 1. The network diagram is shown in figure 4.

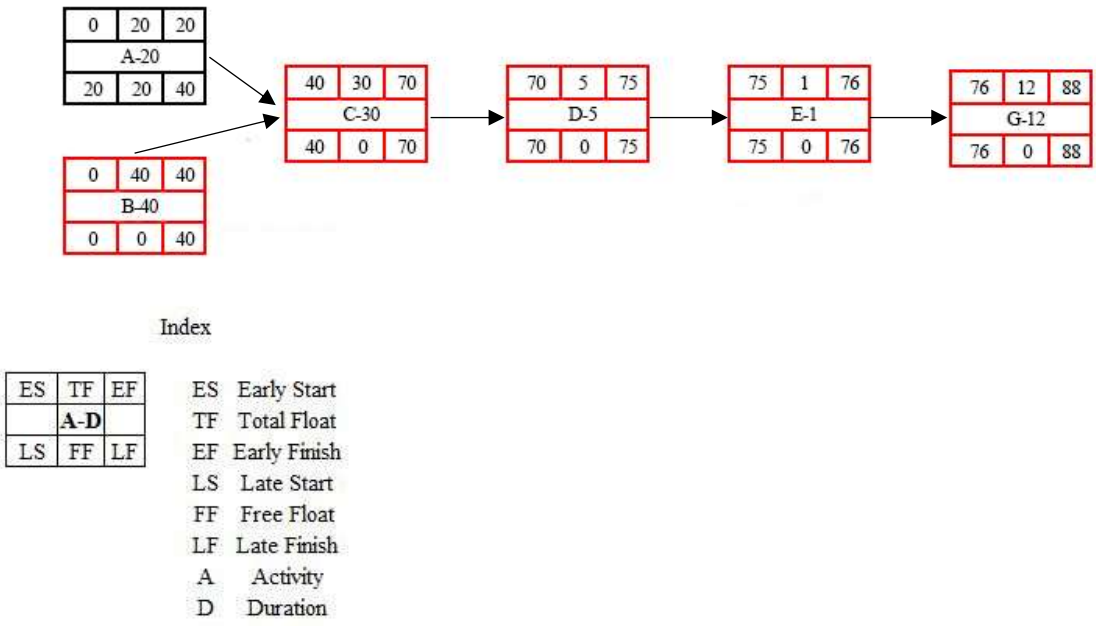


Figure 4- Network diagram to identify critical path

Figure 4 shows the Activity Network Diagram of the project. As shown in the above figure, it is observed that the critical tasks are (A) data collection and preprocessing, (C) building the caption generator model, (D) user-interface design, (E) system testing and (F) documentation. The total duration of the critical path (B-C-D-E-F) is 88 days, which is in the project deadline range. Hence, this project is feasible in terms of the schedule since the project is completed in time if the critical tasks are carried out within the specified task's duration in Table 1.

CHAPTER 4: METHODOLOGY

4.1 Data Preparation

4.1.1 Data Collection

Dataset is the top most important part of our project. Dataset is mainly create by using a huge number of data and then implement in the machine learning or deep learning sector. There are different types of datasets are already existing and they are Pascal VOC dataset, Flickr and MSCOCO Dataset.

We chose Flickr datasets because it contains thousands of images and data that can be used for training neural networks. Raw images are kept in a separate folder, identified only by a stream of numbers, from the JSON files that include image identification numbers, crowd-sourced captions that allow for neural network training. Since neural network training is generally broken into three parts — training, validation, and testing — the Flickr datasets are split similarly. The training portion trains the network by having it perform computations and adjust its weights accordingly; the validation portion is run through the network after training is done as a simulation of testing new input images, but may also tweak the network weights; the testing portion contains new images that can be tested on the refined network. It should be noted that despite having similar formats and coming from the same source, the Flickr datasets are completely separate entities, so there is no overlap between the two. A Python script was written to confirm this exclusivity.

4.1.2 Data Selection/Filtering

From the Flickr dataset, we utilize the Flickr8K and Flickr30K datasets for our project. Flickr8K dataset contain 8,000 and Flickr30K dataset contain 31,000 images. For Flickr8K and Flickr30K dataset, we utilize 1,000 pictures for validation, 1,000 for testing and the rest pictures for training.

We preprocess of our dataset before train up. We convert all sentences of our dataset which is Flickr8K, Flickr30K. We channel words which is occur 5 times in the training set, which result in 2538 words for Flickr8k, 7414 words for Flickr30k dataset. For the beginning period, we select Flickr8K dataset and we use NVIDIA G1 GAMING GPU for train the dataset. Then later we choose the Flickr30K. After completing training and testing of Flickr8K dataset, then procedure for Flickr30K dataset. Since, the Flickr dataset worked completely well and meet our expectations, so we later decided using Flickr for the dataset.

4.2 Algorithms Studied and Implemented

4.2.1 Algorithms

Deep Learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It’s achieving results that were not possible before.

In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the- art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contain many layers.

4.2.1.1 Convolutional Neural Network

A Convolutional Neural Network (CNN) is contains at least one convolutional layer and at least one completely associated layer as in a standard multilayer neural system. The architecture of a CNN is intended to exploit the 2D structure of an information picture. This is accomplished with nearby associations and tied weights took after by some type of pooling which brings about interpretation invariant highlights. Another advantage of CNNs is that they are less demanding to

prepare and have numerous less parameters than completely associated systems with a similar number of concealed units.

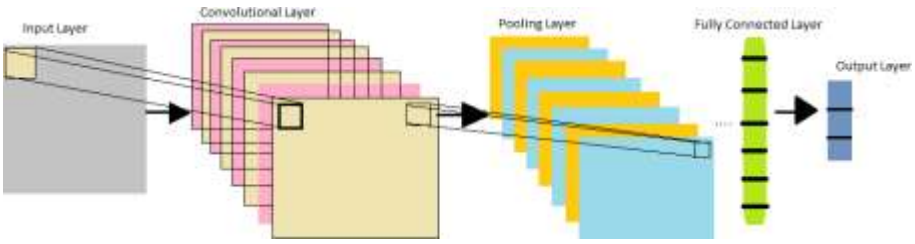


Figure 5- Convolutional Neural Network (CNN)

4.2.1.1.1 Architecture of CNN

CNN are basically used for image recognition, video analysis system, natural language processing and many more. In CNN, input layer, convolutional layer, polling layer, fully connected layer and output layer exist. In input layer there are three measurements and they are width, height and depth. It is a framework of pixel esteem. At that point the convolutional layer existing. A piece of the picture is associated with the following Convolutional layer in light of the fact that if every one of the pixels of the info is associated with the Convolutional layer. Filter, Kernel, or Feature Detector is a little matrix used for highlights location. After convolutional layer, at that point the pooling layer part exists [13]**.** Pool Layer plays out a capacity to decrease the spatial measurements of the information, and the computational unpredictability of our model. After pooling layer, fully connected layer part existing and fully connected layers interface each neuron in one layer to each neuron in another layer. The last fully connected layer utilizes a softmax initiation work for characterizing the produced highlights of the information picture into different classes in light of the training dataset and after completing this layer then we get an output [14].

4.2.1.1.2 Convolutional Layer

The convolutional layer applies a specified number of convolutional filters to an image. For each sub region of the image, the convolutional layer performs mathematical functions on the sub

region, ultimately producing a single value for the output. The output of a CNN in the image processing and classification field of interest is usually a feature map, so the single value output from the convolutional layer is added to the output feature map.

4.2.1.1.3 Pooling Layer

The pooling layer takes image data extracted by the convolutional layer as input. It aims to reduce dimensionality of feature. That is, the pooling layer aims to decrease the processing time when given an input image's resultant feature map. The pooling layer possesses a **max pooling** function, which partitions the input image from the convolutional layer into a set of non-overlapping rectangles, for this purpose. For each sub region (non-overlapping rectangle), a max function is applied to each value in the sub region. The result is a pool of the maximum values of each sub region. Figure 5 visualizes how max pooling works.

4.2.1.2 Recurrent Neural Network (RNN)

Recurrent neural network is a class of simulated neural system where associations between units shape a coordinated diagram along an arrangement. This enables it to show dynamic transient conduct for a period grouping. The thought behind RNNs is to make utilization of consecutive data. Recurrent Neural Network takes the previous output or hidden state as inputs. RNN are helpful for their middle of the road esteems can store data about past contributions for a period that isn't settled from the earlier. RNN basically utilized for language demonstrating and creating content, machine translation, speech recognition, generating image description. In RNN, process sequences are different type like as one to one, one to many, many to one, many to many existing [15].

A RNN is a sort of neural systems, which can send input signals. RNN models a dynamic framework, where the hidden state h_t is not just reliant on the present perception xt yet depend the previous hidden state h_{t-1}. We can represent h_t like this

$$h_t = f(h_{t-1}, x_t)$$

Where f is non-linear mapping. From the above equation, we got h_t which contains information about the whole sequence, which can be derived from the recursive definition in above equation. As such, RNN can utilize the concealed factors as a memory to catch long haul data from a sequence. The following RNN model is

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t + b_h)$$

$$z_t = \text{softmax}(W_{hz}h_t + b_z)$$

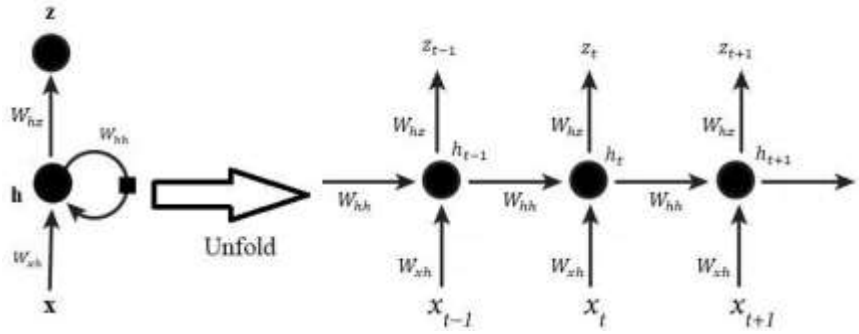


Figure 6- Recurrent Neural Network [15]

Thinking about the differing length for each sequential data, we likewise accept the parameters in each time step are the same over the entire sequential analysis. Else it will be difficult to compute the gradients. Moreover, sharing the weights for any sequential length can sum up the model well. With respect to consecutive naming, we can utilize the most extreme probability to gauge model parameters [16]. As it were, we can limit or minimize the negative log probability the objective function such as

$$L(x,y) = -\sum_t y_t \log z_t$$

In the accompanying, we will utilize documentation L as the objective function for complexity.
In any case, there is gradient vanishing or exploding issues to RNNs. Notice that in final equation of backpropagation demonstrates matrix multiplication over the grouping. Since RNNs need to back propagate gradients over a long arrangement, gradient esteem will recoil layer over layer, and in the long run vanish after a couple of time steps. In this manner, the states that are far from the current time step does not add to the parameters gradient computing. Another bearing is the gradient detonating, which ascribed to huge qualities in matrix multiplication.

4.2.1.3 Long Short-Term Memory (LSTM)

Long short-term memory (LSTM) units are a building unit for layers of a repetitive neural system (RNN). The LSTM utilizes this thought of "Constant Error Flow" for RNNs to make a "Constant Error Carousel" (CEC) which ensures that gradients don't decay. A LSTM unit is fixed of a cell, an input, an output and a forget gate. LSTMs are an uncommon sort of RNN, fit for adapting long haul conditions. LSTM mainly used for robot controlling, rhythm learning, speech recognition, grammar learning, human action recognition, sign language recognition, semantic parsing [17]. LSTM cell stores an esteem which is long or brief eras. This is accomplished by utilizing initiation work for the memory cell. LSTM were intended to battle vanishing slopes through a gating instrument.

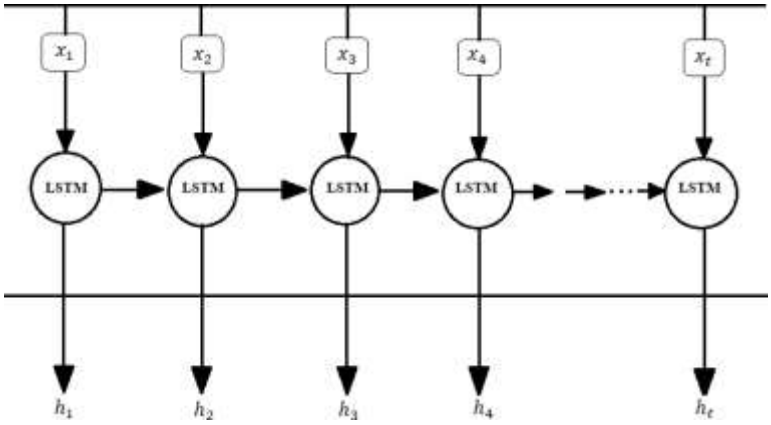


Figure 7 – Long Short-Term Memory (LSTM)

Long short-term memory (LSTM) are a special part of RNN (Recurrent Neural Network) equipped for learning long-term dependencies [18]. LSTM are unequivocally intended to maintain a strategic distance from the long-haul reliance issue. Recalling data for significant lots of time is for all intents and purposes their default conduct, not something they battle to learn. For discuss about the LSTM network, initial phase in our LSTM is to choose what data we will discard from the cell state [19] . This choice is made by a sigmoid layer called the "forget gate layer." It takes a gander at h_{t-1} and, and yields a number somewhere in the range of 0 and 1 for each number in the cell state C_{t-1}. First of all, 1 speaks to "totally keep this" while a 0 speaks to "completely get rid of this."

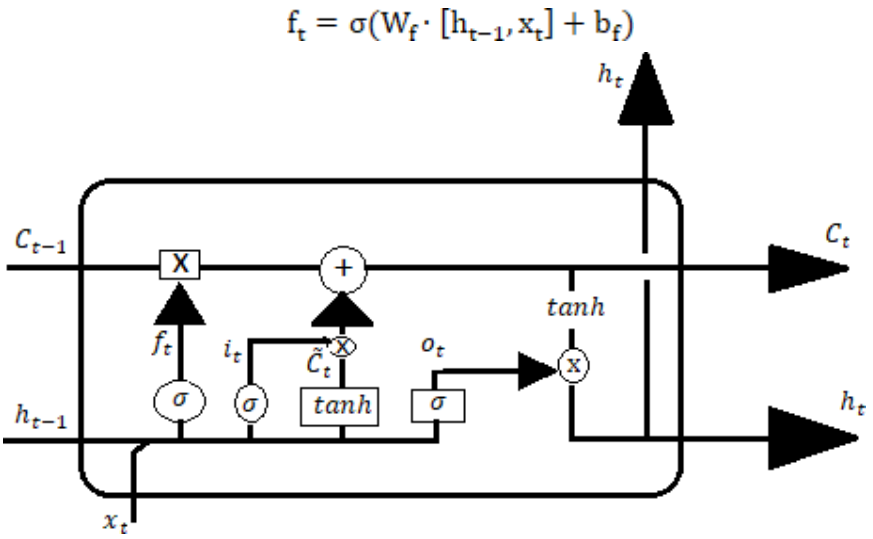


Figure 8 – Long Short Term Memory (LSTM) [17]

4.2.1.4 Hybrid Model

CNN (Convolutional Neural network) is a feed forward neural network, which mean the output of the neuron does not loop back to become the input of the previous time step. Hence, R-CNN is Recurrent Convolutional Neural Network, which means the output of the neuron is loop back to become the input of the previous time steps. A convolutional neural network (CNN) mainly works for image classification. But, R-CNN works for region and object detection. Generally, a normal CNN can locate the class of objects but not where they are located. If three are any multiple objects are staying in the visual data, then the CNN bounding box cannot work properly due to interference. But, in R-CNN the CNN forced to focusing on a single region at a time. Because of it is expected that, only a single object of interest will dominate in a given region. The region of the R-CNN is detected by selective search algorithms followed by resizing so that the region is equal size before all data train up.

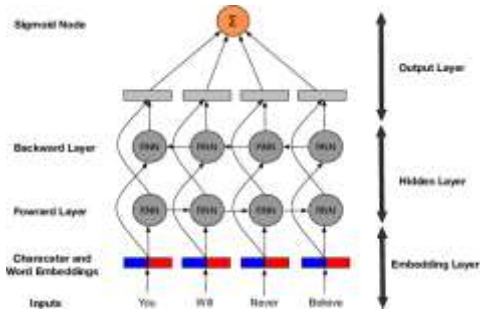


Figure 9 – Architecture of a Bidirectional Recurrent Neural Network (BRNN) [16]

Bidirectional Recurrent Neural Network (BRNN) is a part of RNN section and which is use a finite sequence to prediction. In BRNN model, there are label each element of the sequence based on the past and future context element. In this BRNN, this task done by the close the output of two RNN and one processing of the sequence is from left to right, another sequence from right to left. The joined outputs are the prediction of the given target signals. This technique proved to especially useful when combined with LSTM and RNNs method. In BRNN model, first of all stay input layer. After that, word embedding and forward layer exist. Then finally backward layer existing with sigmoid node which is called output layer in BRNN. The BRNN model used for the speech recognition (combined with LSTM), translation, handwritten recognition, dependency parsing and many more.

4.3 System Design

4.3.1 Flow Diagram

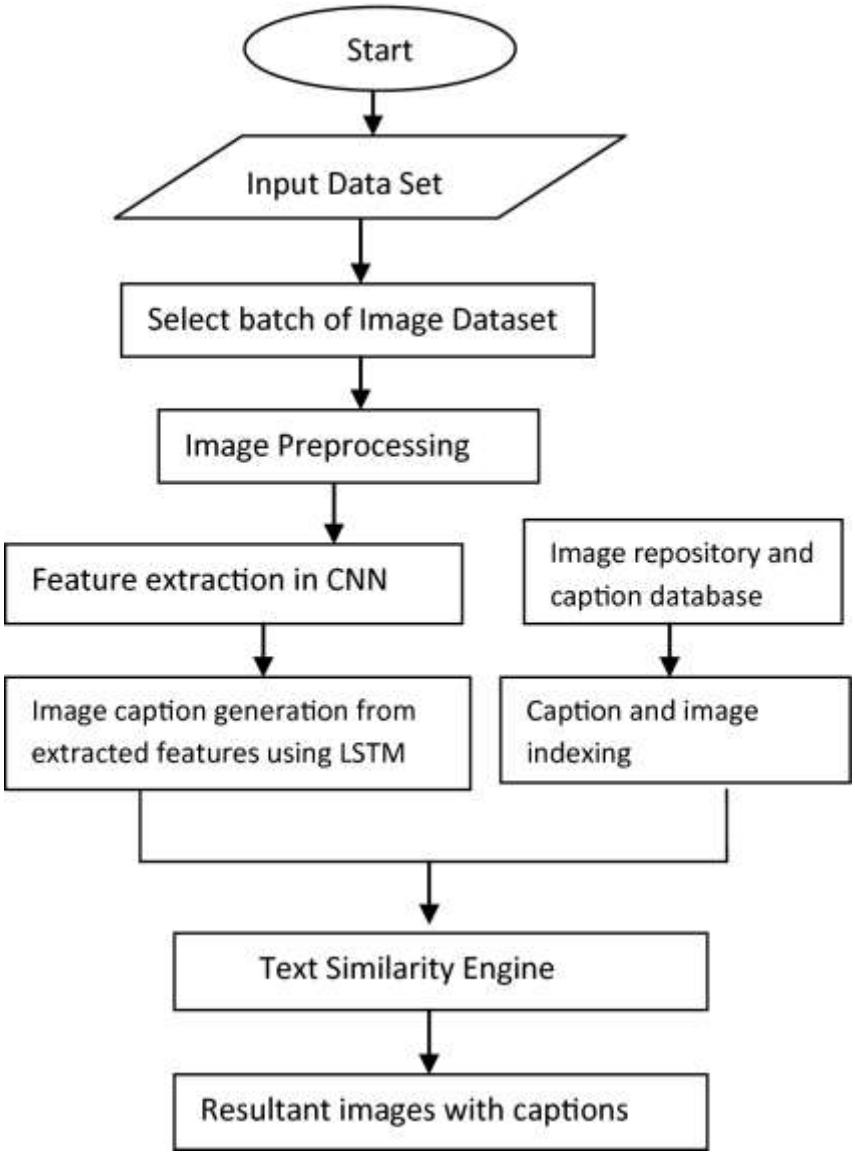


Figure 10- Flow Diagram of the System

Figure 10 shows the block diagram of the system. First, the input dataset is acquired where the batch of image dataset is selected. Then the image is preprocessed. After preprocessing, the image is extracted using CNN and image caption is generated using extracted features using LSTM. Then, image repository and caption database are used such that the caption for the images are

indexed with later functions with caption generated by LSTM for the similarity. The resultant image along with the caption is obtained with the help of text similarity engine.

4.3.2 Block Diagram

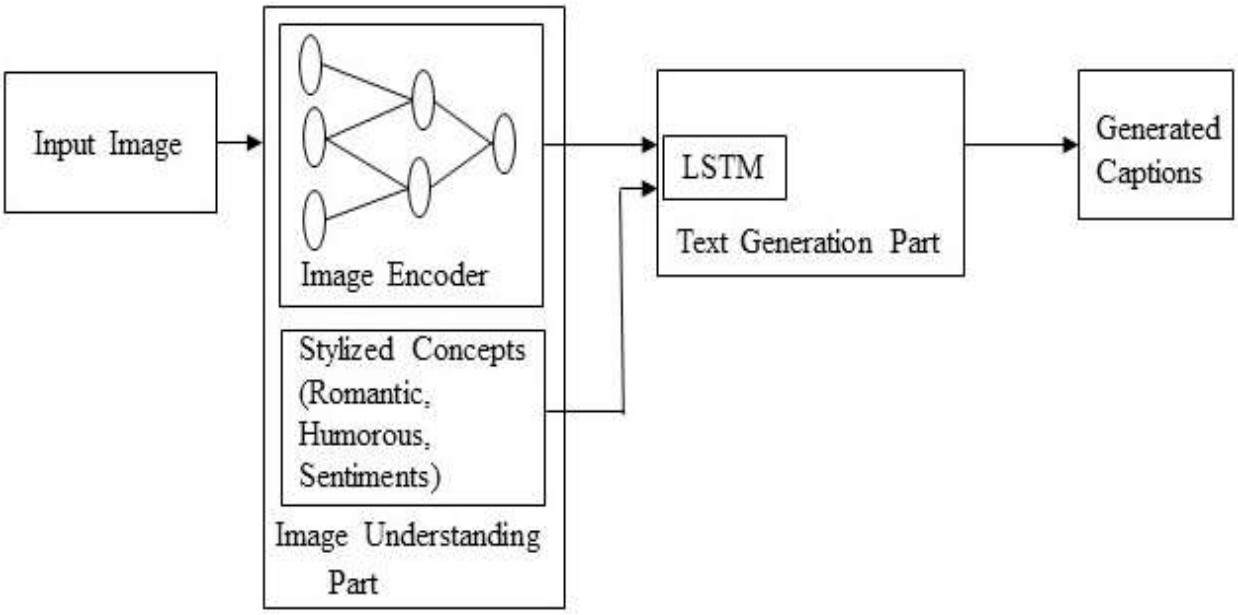


Figure 11- Block diagram of Caption Generator

Figure 11 shows the block diagram of Caption generation. Here, the input image is processed to image encoder, where the image is encoded using CNN with addition of different datasets. The encoded image is then sent towards LSTM along with RNN for the text generation and arrangement part such that the Caption are generated.

CHAPTER 5: IMPLEMENTATION AND EVALUATION

5.1 Tools and Technologies Used

This section describes the tools and technologies used in the project.

CASE tools:

- Draw.io

Client Side:

- HTML is used structure the user interface and display the output results.
- Bootstrap Framework is used to beautifying the user interface.

Server Side:

- Python programming language is used as the server-side programming language.
- OS and NumPy are used for file handling and array processing functionalities respectively.
- Keras machine learning library is used to build the language model architecture and train the language model architecture.

Hardware:

- The language model was trained on a desktop having following specs:
  a) Configuration Processor: Intel®CoreTM i7 RAM: 16 GB
  b) Hard Disk: 500 GB
  c) Graphics: NVIDIA G1 GAMING GPU-1650 Ti - 4 GB
  d) Operating System: Ubuntu 16.04 LTS Programming Language: Python

5.2 Implementation

5.2.1 Image Processing

We need to resize images all of our dataset. We also need to the dataset JSON file, and VGG CNN features for our benchmark dataset Flickr8k and Flickr30k. We use raw image files of each dataset alongside JSON file and VGG CNN features. The input is dataset of images and 5 sentences descriptions which were collected with Amazon Mechanical Turk.

Figure 12 – Example of sentence anticipated by our model.

In the training section, all of images are fed as input to RNN and RNN asked to predict the word of the sentences. For the prediction part, images are passed to RNN and RNN generates the sentence word at a time and we get result of our evaluation with BLEU and METEOR scale.

We use json, datetime, pickle, math, caffe, numpy, scipy, tensorflow, code, socket, argparse, os, and time library for our work. We also use vgg_feats.mat which is a. mat file and that stores the CNN features. We use 512 hidden layers and from imagernn.data_provider use getDataProvider for this project. We also use imread, imresize for image resizing or reshaping. After completing resize of images, then we attempt to train up the whole dataset. Flickr8k take 1 day and Flickr30k take 2 days (approx.) to completing whole dataset train up.

5.2.2 Representation

Representing image is most important part for image processing and we get a lot of ideas to review many recent works [20]. We watch that sentence description make visit references to objects and their attributes. The CNN is pre-prepared on ImageNet [21] what's more, fine-tuned on the 200 classes of the ImageNet Detection Challenge [22]. We maintain the technique for Girshick et al. [23] to detect each object in each image with a Region Convolutional Neural Network (RCNN). The RCNN model has two parts, a region proposal network and another one is binary mask classifier. Following Karpathy et al. [4] , we use the primary 19 identified area despite the whole picture. Then compute the representation in light of the pixels I_b inside each bounding box as takes after:

$$v = W_m[CNN\theta_c(I_b)] + b_m$$

The CNN (I_b) changes the pixels inside the bounding box (I_b) to 4096-dimensional enactment of the fully connected layer in a split second before the classifier. The CNN parameters θ_c contain around 60 million parameters. The matrix W_m has measurements h×4096, where h is the extent of the multimodal inserting space. Each image represents as h-dimensional vectors.

Representing sentence is most important section for our model. We established the inter-model relationship and like to represent the words in a sentence in the same established h-dimensional space as the image regions. The direction does not consider any order and word name in a sentence in formation. We propose to utilize a Bidirectional Recurrent Neural Network (BRNN) [24] to compute the word representation. Bidirectional Recurrent Neural Network (BRNN) is a part of RNN section and which is use a finite sequence to prediction. In BRNN model, there are label each element of the sequence based on the past and future context element. In this BRNN, this task done by the close the output of two RNN and one processing of the sequence is from left to right, another sequence from right to left. The joined outputs are the prediction of the given target signals. This technique proved to especially useful when combined with LSTM and RNNs method. The BRNN model used for the speech recognition (combined with LSTM), translation, handwritten recognition, dependency parsing and many more. For our model, the BRNN takes a sequence of N words and then it transforms each to h-dimensional vector. Utilizing the list t = 1…..N to indicate the situation of a word in a sentence, the exact shape of the BRNN is as per the following:

$$x_t = W_w \mathbb{I}_t$$
$$e_t = f(W_e x_t + b_e)$$
$$h_t^f = f(e_t + W_f h_{t-1}^f + b_f)$$
$$h_t^b = f(e_t + W_b h_{t+1}^b + b_b)$$
$$s_t = f(W_d(h_t^f + h_t^b) + b_d)$$

The weights W_w determine a word inserting network that we instate with 300- dimensional word2vec [25] weights and keep settled as a result of overfitting concerns. II_t is a pointer column vector that has a single one at the record of the t-th word in a word vocabulary. The BRNN comprises of two independent streams of handling, one moving left to right (h^f_t) and the other right to left (h^b_t). We take in the parameters W_e, W_f, W_b, W_a and the individual inclinations b_e, b_f, b_b, b_d. We set the activation function f to the rectifier linear unit (ReLU).

5.2.3 Decoding

Decoding considers a picture from the training set and its comparing sentence. We are ultimately interested in producing snippets of content of single words, we might want to align extended, adjacent sequences of words to a single bounding box. We can translate the amount $v_i^T s_t$ as the unnormalized log likelihood of the t-th word depicting any of the bounding boxes in the image. Note that the naive arrangement that assigns each word freely to the highest scoring locale is lacking in light of the fact that it prompts words getting scattered conflictingly to various regions. We regard the genuine arrangements as inactive factors in a Markov Random Field (MRF) where the binary collaborations between neighboring words urge an arrangement to a similar district. Solidly, given a sentence with N words and a picture with M jumping boxes, we present the inactive arrangement variable $a_j \in \{1 \ldots . M\}$ for. Here, define a MRF in a chain structure along the sentence as takes after:

$$E(a) = \sum_{j=1\ldots N} \psi_j^U(a_j) + \sum_{j=1\ldots N-1} \psi_j^B(a_j, a_{j+1})$$

$$\psi_j^U(a_j = t) = u_i^T s_t$$

$$\psi_j^B(a_j, a_{j+1}) = \beta \mathbb{I}[a_j = a_{j+1}]$$

Here, β is a hyperparameter that controls the partiality towards longer word phrases. This parameter enables us to introduce between single-word arrangements (β = 0) and adjusting the whole sentence to a solitary, maximally scoring area when β is extensive. The yield of this procedure is a set of image areas explained with fragments of content.

5.2.4 Optimization

For Flickr8K and Flickr30K dataset, we utilize SGD with mini batch of 100 picture sentence sets furthermore, speed of 0.9 to optimization to the alignment model. We likewise utilize dropout regularization in all layers with the exception of in the recurrent layers [26] and clip gradient element wise at 5 (essential). We cross-approve the learning rate and the weight rot. The generative RNN is harder to optimization because of the word frequency difference between uncommon words and common words. We accomplished the best outcomes utilizing RMSprop [27] That method is a versatile advance size strategy that scales the refresh of each weight by a running normal of its gradient standard.

5.3 Testing and their Results

Here we discuss about result and discussion part of ImageToText. In section 4.3.1, we mainly represent tabular and graphical result of Flickr8K and Flickr30K. We also show our result and compared with other methods for Flickr8K and Flickr30K dataset in this section.

5.3.1 Representation of Results for Three Standard Image Datasets

In this section, we represent tabular representation of test result of ImageToText using three benchmark datasets. We also discussed about graphical representation of ImageToText in this section.

We used BLEU and METEOR metric for our evaluation system. BLEU (bilingual evaluation understudy) is an algorithm for evaluating the quality of text which has been machine-translated

from one natural language to another. BLEU, or the Bilingual Evaluation Understudy, is a score for comparing a candidate translation of text to one or more reference translations. METEOR (Metric for Evaluation of Translation with Explicit Ordering) is a metric for the evaluation of machine translation output.

5.3.2 Tabular Representation of Test Result

We maintain 512 hidden layers and evaluation of full image predictions on 1,000 test images. We follow the BLEU-1, 2, 3, 4 evaluation and METEOR metrics for our evaluation result. We use 1,000 images from the Flick8k dataset for test result. We show our two dataset result in Table 2 and compare to other methods, (-) indicates an unknown metric. For the model evaluation, BLEU-1, 2, 3, 4 evaluations and METEOR metric are used and compare our results with benchmark results of Mao et al. [28] Google NIC [29], LRCN [30], MS Research [31], Chen and Zitnick [6] model. From the Table of 2, we showed that using our method we get better result in BLEU-2 evaluation compared with Mao et al. [28] model.

Table 2- Our evaluation result for 8K dataset

Dataset	**Model**	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**	**METEOR**
Flickr8K	Mao et al. [33]	58	28	23	-	-
	Google NIC [34]	63	41	27	-	-
	LRCN [35]	-	-	-	-	-
	MSResearch [36]	-	-	-	-	-
	Chen and Zitnick [3]	-	-	-	14.1	-
	Proposed Model	52.6	34.4	21.8	14.1	16.495543

For working with Flickr30k dataset, we evaluate of full image predictions on 1,000 test images. We follow the BLEU-1, 2, 3, 4 evaluation and METEOR metrics for our evaluation result. We use 1,000 images from the Flickr30k dataset for test result. We show full dataset result in Table 3 and compare to other methods, (-) indicates an unknown metric. Here, we got better accuracy in BLEU-1, BLEU-2 evaluation compared with Mao et al. model which is 56.8 and 37.3 respectively.

Table 3 – Our evaluation result for Flickr30K datasest

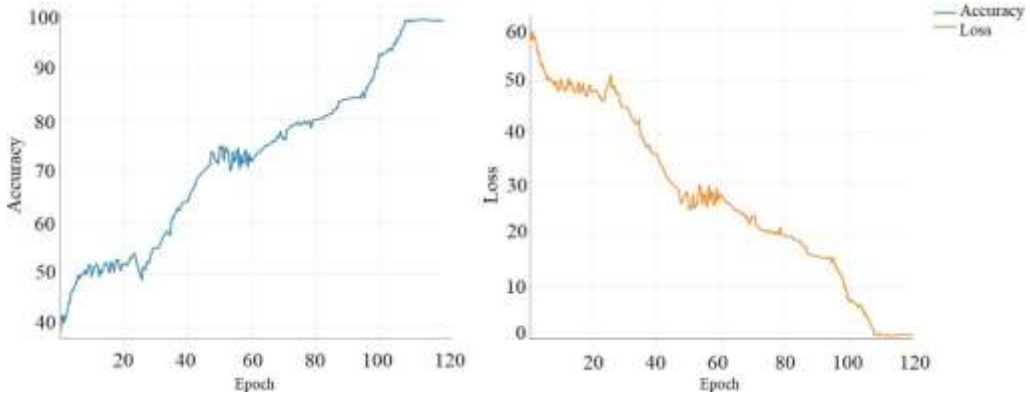
Dataset	Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
Flickr30K	Mao et al. [33]	55	24	20	-	-
	Google NIC [34]	66.3	42.3	27.7	18.3	-
	LRCN [35]	58.8	39.1	25.1	16.5	-
	MSResearch [36]	-	-	-	-	-
	Chen and Zitnick [3]	-	-	-	12.6	-
	Proposed Model	56.8	37.3	24.1	15.6	19.441452

5.3.3 Graphical Representation of Test Result

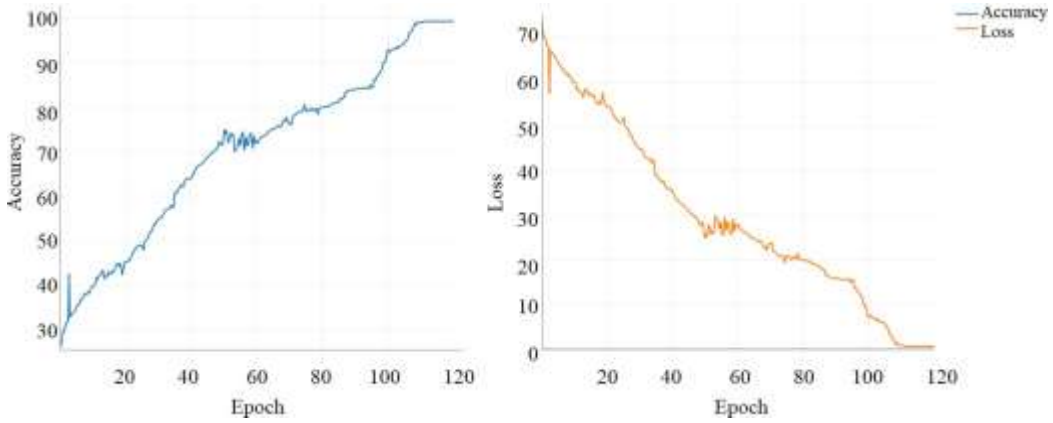
We use two different benchmark dataset Flickr8k and Flickr30k. We represent training and testing result of each dataset in different graph. In Figure 4.3.3, we represent the training time accuracy, loss vs. epoch using different graph. In training graph, we show graphically epoch vs. loss and epoch vs. accuracy for two different dataset. For two different dataset of training graph, in horizontally, we show epoch number and in vertically, we show accuracy and loss part.

For measure accuracy, we used classification accuracy metrics. It is the ratio of number of correct predictions to the total number of input samples. It works well only if there are equal number of samples belonging to each class. Accuracy measure equation is –

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions made}}$$



(a) Epoch vs. loss, and Epoch vs. accuracy for Flickr8K



(b) Epoch vs. loss, and Epoch vs. accuracy for Flickr30K

Figure 13 – Graphical representation of training time using two benchmark dataset (epoch vs. accuracy an epoch vs. loss)

In graphical representation, we show that epoch vs. loss, epoch vs. accuracy for two different datasets. We run 120 epochs and show their training time accuracy and loss vs. epoch graphically for two benchmark datasets: Flickr8K and Flickr30K in Figure 13. We represent those in two different graphs for two different dataset which is easily understandable. Our model increase training accuracy and reduce the loss during training time. We also get better accuracy and BLEU evaluation result compare with Karpathy [4].

5.3.4 Discussion

We use two benchmark datasets in our work: Flickr8k and Flickr30k. We compare our results with earlier benchmark results and observe that our model gives better accuracy than the previous results. We use specifically 512 hidden layers in our model which is discussed in the simulation part. For the model evaluation, BLEU-1, 2, 3, 4 evaluations and METEOR metric are used and compare our results with benchmark results of Mao et al. [28], Google NIC [29], LRCN [30], MS Research [31], Chen and Zitnick [6] model. Finally, we use METEOR metric evaluation and get 16.495543 and 19.441452 for the benchmark datasets respectively and observe improvements in our result.

CHAPTER 6: CONCLUSIONS AND LIMITATIONS

6.1. Conclusions

Image captioning has made significant advances in recent years. Recent work based on deep learning techniques has resulted in a breakthrough in the accuracy of image captioning. They can improve the content-based image retrieval efficiency, the expanding application scope of visual understanding in the fields of medicine, security, military and other fields, cross media retrieval, video captioning and video dialog which has a broad application prospect. We introduced with a recurrent neural network model which generates sentence from the given input image. The model identifies the image region and generates natural language description of images. Our approach includes a lowering of resolution images that adjusted parts of visual and language modalities through a typical multimodal embedding. We also introduced and implemented it in CNN features and RNN features. Then finally showed that how can generate caption from a given input image. Our main achievements for Flickr8K and Flickr30 dataset was increased accuracy and reduce loss. However, with the advent of deep learning architecture, automatic image captioning will remain active research for some time.

6.2. Limitations

- The training of the neural network needs to be tweaked, as word bias became evident and caused many captions to be inaccurate despite greater confidence in the predictions.
- The language model requires huge computing resource to train.

REFERENCES

[1] M. H. M. A. S. P. Y. C. R. Ali Farhadi, "Every picture tells a story: Generating sentences," in *11th European Conference on Computer Vision: Part IV, ECCV'10*, Berlin, Heidelberg, 2010.

[2] X. C. a. C. L. Zitnick, "Learning a recurrent visual representation for image," CoRR, 2014.

[3] R. S. a. R. S. Z. Ryan Kiros, "Unifying visual-semantic embeddings with multimodal neural language models," CoRR, abs/1411.2539, 2014.

[4] A. J. a. L. F.-F. A. Karpathy, Deep fragment embeddings for bidirectional image sentence mapping, arXiv preprint arXiv:1406.5679, 2014.

[5] P. O. V. B. T. L. a. C. Y. Kuznetsova, Treetalk: Composition and compression of trees for image descriptions, TACL, 2014.

[6] X. C. a. C. L. Zitnick., "Learning a recurrent visual representation for image," CoRR, abs/1411.5654, 2014.

[7] M. Soh, "Learning CNN-LSTM Architectures for Image," Stanford university.

[8] P. Z. Q. H. H. Z. Z. G. X. ao Xu, "AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks," arXiv:1711., 2017.

[9] D. S. V. M. S. E. K. Y. B. Shikhar Sharma, "ChatPainter: Improving Text to Image Generation using Dialogue," arXiv:1802.08216v1 [cs.CV], 2018.

[10] A. K. Q. V. L. C. D. M. A. Y. N. Richard Socher, "Grounded Compositional Semantics for Finding and Describing Images with Sentences," TACL, 2014.

[11] L. F.-F. Andrej Karpathy, "Deep Visual-Semantic Alignments for Generating," CVPR, 2015.

[12] M. A. JISHAN, "IMAGETOTEXT: IMAGE CAPTION GENERATION USING HYBRID," UNIVERSITY OF LIBERAL ARTS BANGLADESH, Dhaka, Bangladesh, 2019.

[13] W. X. &. Y. Y. &. J. &. Z. H. A. Y. Junhua Mao, "Deep Captioning With Multimodal Recurrent Neural Networks (M-RNN)," arXiv:1412.6632v5 [cs.CV], 2015.

[14] U. M. a. J. Ì. S. Dan Cires Ì˘gan, "Multi-column Deep Neural Networks for Image Classification," arXiv:1202.2745v1, 2012.

[15] C. G. y. C. a. Y. B. Junyoung Chun, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," arXiv:1412.3555v1, 2014.

[16] Y. Q. F. X. a. F. K. S. Yuchen Fan, "TTS Synthesis with Bidi rectional LSTM based Recurrent Neural Networks," in *Conference of the International Speech Communication Association*, 2014.

[17] S. T. J. X. F. W. a. Z. (. Z. Jun Song, LSTM-in-LSTM for generating long descriptions of images, Computational Visual Media, 2016.

[18] J. B. a. C. E. Zachary C. Lipton, "A Critical Review of," arXiv:1506.00019v4, 2015.

[19] X. G. H. L. R. L. a. S. S. Junhyuk Oh, "Action-Conditional Video Prediction using Deep Networks in Atari Games.," 2015.

[20] V. P. S. D. S. L. Y. C. A. C. B. a. T. L. B. G. Kulkarni, "Baby talk: Understanding and generating simple image descriptions," CVPR, 2011.

[21] W. D. R. S. L.-J. L. K. L. a. L. F.-. F. J. Deng, "Imagenet: A large-scale hierarchical image database," CVPR, 2009.

[22] J. D. H. S. J. K. S. S. S. M. Z. H. A. K. A. K. M. B. A. C. B. a. L. F.-F. O. Russakovsky, "Imagenet large scale visual recognition challenge," 2014.

[23] J. D. T. D. a. J. M. R. Girshick, "Rich feature hierarchies for accurate object detection and semantic segmentation," CVPR, 2014.

[24] M. S. a. K. K. Paliwal., "Bidirectional recurrent neural networks," Signal Processing, IEEE Transactions, 1997.

[25] I. S. K. C. G. S. C. a. J. D. T. Mikolov, "Distributed representations of words and phrases and their compositionality," NIPS, 2013.

[26] I. S. a. O. V. W. Zaremba, "Recurrent neural network regularization," arXiv preprint arXiv:1409.2329, 2014.

[27] T. T. a. G. E. Hinton., "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," 2012.

[28] W. X. Y. Y. J. a. A. L. Y. J. Mao, "Explain images with multimodal recurrent neural networks," arXiv preprint arXiv:1410.1090, 2014.

[29] A. T. S. B. a. D. E. O. Vinyals, "Show and tell: A neural image caption generator," arXiv preprint arXiv:1411.4555, 2014.

[30] L. A. H. S. G. M. R. a. S. V. J. Donahue, "Long-term recurrent convolutional networks for visual recognition and description," arXiv preprint arXiv:1411.4389, 2014.

[31] S. G. F. I. R. S. L. D. P. D. J. G. X. H. M. M. J. P. e. a. H. Fang, "From captions to visual concepts and back," arXiv preprint arXiv:1411.4952, 2014.

[32] A. K. Q. V. L. C. D. M. A. Y. N. Richard Socher, "Grounded Compositional Semantics for Finding and Describing Images with Sentences," TACL, 2014.