

A Project Report
On
ASSERTIVE VISION USING DEEP LEARNING

Submitted in partial fulfillment of the
Requirement for the award of the degree of

Bachelor of Technology
In
Computer Science and Engineering



Under The Supervision of

Dr. T. Poongodi

Professor

Submitted By

Siddhant Singh Bhadauria (18SCSE1010024)
Dharmendra Bisht(18SCSE1010651)

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
MAY 2021-2022



**SCHOOL OF COMPUTING SCIENCE AND
ENGINEERING
GALGOTIAS UNIVERSITY, GREATER
NOIDA**

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **“ASSERTIVE VISION USING DEEP LEARNING.”** in partial fulfillment of the requirements for the award of the B.tech Computer Science submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of month, Year to Month and Year, under the supervision of Dr.T.Poongodi Associate Professor, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Siddhant singh Bhadauria

Dharmendra Bisht

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr.T.Poongodi
Professor

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of Siddhant Singh Bhadauria (18SCSE1010024), Dharmendra Bisht(18SCSE1010651) has been held on _____ and his/her work is recommended for the award of B.Tech Computer Science.

Signature of Examiner(s)

Signature of Supervisor(s)

Signature of Project Coordinator

Signature of Dean

Date:

Place: Greater Noida

Abstract

In this new era of technology, companies and developers around the world are talking about embracing artificial intelligence (AI), machine learning (ML), and deep learning (DL). Deep learning systems help a computer model to filter the input data through layers to predict and classify information. Assertive vision also focuses on artificial Intelligence and Machine learning resources and its other concepts for identifying the image and object on the basis of their attributes and features and then will provide caption to them and then the caption text which is generated will be converted to voice using API's. Computer vision based assertive devices for the blind is promising and efficient technology and help the blind people in understanding the surrounding. The purpose of this model is to generate captions for an image. Image captioning aims at generating captions of an image automatically using deep learning techniques. Initially, the objects in the image are detected using a Convolutional Neural Network (InceptionV3). Using the objects detected, a syntactically and semantically correct caption for the image is generated using Recurrent Neural Networks (LSTM) with attention mechanism. Computer vision has become ubiquitous in our society, with applications in several fields. In this project, we focus on one of the visual recognition facets of computer vision, i.e. image captioning. The problem of generating language descriptions for visual data has been studied from a long time but in the field of videos. In the recent few years emphasis has been lead on still image description with natural text. Due to the recent advancements in the field of object detection, the task of scene description in an image has become easier. Computer vision has become ubiquitous in our society, with applications in several fields. In this project, we focus on one of the visual recognition facets of computer vision.

Keywords: Computer Vision, CNN, LSTM, Deep learning.

TABLE OF CONTENTS

Title	Page No
CERTIFICATE	ii
ACKNOWLEDGEMENT	iii
ABSTRACT	iv
CONTENTS	v
LIST OF TABLES	vii
LIST OF FIGURES	viii
ABBREVIATIONS	ix
CHAPTER 1: INTRODUCTION	1
1.1 Motivation	2
1.2 Deep Learning	3
1.3 Model Overview	4
1.4 Image Captioning	5-6
CHAPTER 2: LITERATURE REVIEW	7
2.1 Image Captioning Methods	8
2.1.1 Template-Based Approaches	8
2.1.2 Retrieval-Based Approaches	9
2.1.3 Novel Caption Generation	10
2.2 Deep Learning Based Image Captioning Methods	10
2.2.1 Visual Space vs. Multimodal Space.	10
2.3 Supervised Learning Vs. Other Deep Learning	11
2.3.1 Supervised Learning-Based Image Captioning	11
2.3.2 Other Deep Learning-Based Image Captioning	11
2.4 Dense Captioning Vs. Captions For The Whole Scene	12
2.4.1 Dense Captioning	12
2.4.2 Captions For The Whole Scene	12
2.5 Encoder-Decoder Architecture Vs. Compositional Architecture	13
2.5.1 Encoder-Decoder Architecture-Based Image Captioning	13
2.5.2 Compositional Architecture-Based Image Captioning	13
2.6 Lstm Vs. Others	14

CHAPTER 3: PROBLEM FORMULATION	15
3.1 Problem Identification	15
3.1.1 The Vanishing Gradient Problem	15
CHAPTER 4: PROPOSED WORK	16
4.1 Convolutional Neural Network	17
4.2 Long Short Term Memory	20
CHAPTER 5: SYSTEM DESIGN	21
5.1 FLICKR8K DATASET	21
5.2 Image Data Preparation	22
5.3 Caption Data Preparation	24
5.3.1 Data Cleaning	26
CHAPTER 6: IMPLEMENTATION	27
6.1 Pre-Requisites	
6.2 Project File Structure	28
6.3 Building The Python Based Project	28
6.3.1 Getting And Performing Data Cleaning	30
6.3.2 Extracting The Feature Vector From All Images	31
6.3.3 Loading dataset for Training the model	31
6.3.4 Tokenizing The Vocabulary	31
6.3.5 Create Data generator	31
6.3.6 Defining the CNN-RNN model	32
6.3.7 Training the model	33
6.3.8 Testing the model	34
CHAPTER 7: CONCLUSION, LIMITATION AND FUTURE SCOPE	
8.1 Conclusion	35
8.2 Limitations	36
8.3 Future Scope	37
REFERENCE	38-39

List of figures

Figure No	Title Name	Page No
1	An overview of image captioning model	4
2	An overall taxonomy of deep learning-based image captioning	6
3	Novel Caption Generation	9
4	A Block Diagram Multimodal Space - Based Image Captioning	11
5	A Block Diagram of Simple Encoder- Decoder Architecture Based Image Captioning	12
6	A Block Diagram of a Compositional Network Based Captioning	14
7	A Simple Convent Architecture	17
8	A Gray Scale Image as Matrix of Numbers	18
9	Image and Filter	19
10	Convolutional Operator	19
11	Output after a Rely Operation	20
12	Max Pooling Operation	21
13	An Example of Fully Connected Layer of data with four classes	22
14	Accuracy and Loss Plot on Training and Validation Set	22
15	Model Image Caption Generator	23
16	Forget Gate, Input Gate, Output Gate	24
17	Feature Extraction in Image using VGG	27
18	Flicker Data set Text Format	30
19	Flicker Data Set Python File	32
20	Final Model Structure	36

1. INTRODUCTION

Artificial Intelligence (AI) is now at the heart of innovation economy and thus the base for this project is also the same. In the recent past a field of AI namely Deep Learning has turned a lot of heads due to its impressive results in terms of accuracy when compared to the already existing Machine learning algorithms. The task of being able to generate a meaningful sentence from an image is a difficult task but can have great impact, for instance helping the visually impaired to have a better understanding of images.

Automatically generating captions to an image shows the understanding of the image by computers, which is a fundamental task of intelligence. For a caption model, it not only needs to find which objects are contained in the image and also needs to be able to express their relationships in a natural language such as English. Recently work also achieve the presence of attention, which can store and report the information and relationship between some most salient features and clusters in the image. This project describes approaches to caption generation that attempt to incorporate a form of attention with two variants: a “hard” attention mechanism and a “soft” attention mechanism.

In his work, the comparison of the mechanism shows “soft” works better and we will implement the “soft” mechanism in our project. If we have enough time we will also implement a “hard” mechanism and compare the results. In our project, we do image-to-sentence generation. This application bridges vision and natural language. If we can do well in this task, we can then utilize natural language processing technologies to understand the world in images. In addition, we introduced an attention mechanism, which can recognize what a word refers to in the image, and thus summarize the relationship between objects in the image. This will be a powerful tool to utilize the massive unformatted image data, which dominates the whole data in the world. We will then try to convert that captioned to text to voice so that physically challenged peoples have a better understanding of the surrounding.

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. Captions for every image on the internet can lead to faster and descriptively accurate images searches and indexing.

Ever since researchers started working on object recognition in images, it became clear that only providing the names of the objects recognized does not make such a good impression as a full human-like description. As long as machines do not think, talk, and behave like humans, natural language descriptions will remain a challenge to be solved. Image captioning has various applications in various fields such as biomedicine, commerce, web searching and military etc. Social media like Instagram , Facebook etc. can generate captions automatically from images.

1.1 MOTIVATION

Generating captions for images is a vital task relevant to the area of both Computer Vision and Natural Language Processing. Mimicking the human ability of providing descriptions for images by a machine is itself a remarkable step along the line of Artificial Intelligence. The main challenge of this task is to capture how objects relate to each other in the image and to express them in a natural language (like English).Traditionally, computer systems have been using pre-defined templates for generating text descriptions for images. However, 1 this approach does not provide sufficient variety required for generating lexically rich text descriptions. This shortcoming has been suppressed with the increased efficiency of neural networks. Many state of art models use neural networks for generating captions by taking image as input and predicting next lexical unit in the output sentences.

1.2 WHAT IS DEEP LEARNING

Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain—albeit far from matching its ability—allowing it to “learn” from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.

Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention. Deep learning technology lies behind everyday products and services (such as digital assistants, voice-enabled TV remotes, and credit card fraud detection) as well as emerging technologies (such as self-driving cars). Machine learning and deep learning models are capable of different types of learning as well, which are usually categorized as supervised learning, unsupervised learning, and reinforcement learning. Supervised learning utilizes labeled datasets to categorize or make predictions; this requires some kind of human intervention to label input data correctly. In contrast, unsupervised learning doesn’t require labeled datasets, and instead, it detects patterns in the data, clustering them by any distinguishing characteristics. Reinforcement learning is a process in which a model learns to become more accurate for performing an action in an environment based on feedback in order to maximize the reward.

Deep neural networks consist of multiple layers of interconnected nodes, each building upon the previous layer to refine and optimize the prediction or categorization. This progression of computations through the network is called forward propagation. The input and output layers of a deep neural network are called visible layers. The input layer is where the deep learning model ingests the data for processing, and the output layer is where the final prediction or classification is made.

1.3 Model Overview

The model proposed takes an image I as input and is trained to maximize the probability of $p(S|I)$ where S is the sequence of words generated from the model and each word S_t is generated from a dictionary built from the training dataset. The input image I is fed into a deep vision Convolutional Neural Network (CNN) which helps in detecting the objects present in the image. The image encoding is passed on to the Language Generating Recurrent Neural Network (RNN) which helps in generating a meaningful sentence for the image as shown in the fig. 1. An analogy to the model can be given with a language translation RNN model where we try to maximize the $p(T|S)$ where T is the translation to the sentence S . However, in our model the encoder RNN which helps in transforming an input sentence to a fixed length vector is replaced by a CNN encoder. Recent research has shown that the CNN can easily transform an input image to a vector.

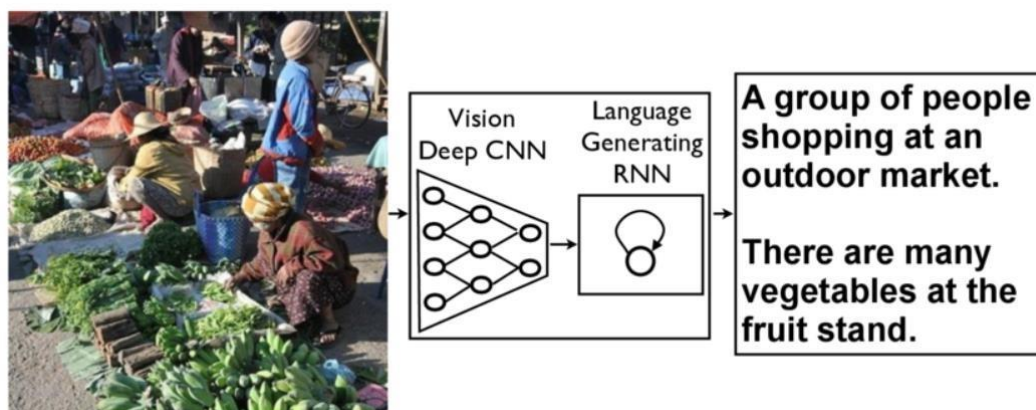


Figure 1.- An overview of image captioning model

For the task of image classification, we use a retrained model VGG16. The details of the models are discussed in the following section. A Long Short-Term Memory (LSTM) network follows the retrained VGG16 [2]. The LSTM network is used for language generation. LSTM differs from traditional Neural Networks as a current token is dependent on the previous tokens for a sentence to be meaningful and LSTM networks take this factor into account.

1.4 IMAGE CAPTIONING

Process:-Image Captioning is the process of generating textual description of an image. It uses both Natural Language Processing and Computer Vision to generate the captions. Image captioning is a popular research area of Artificial Intelligence (AI) that deals with image understanding and a language description for that image. Image understanding needs to detect and recognize objects. It also needs to understand scene type or location, object properties and their interactions. Generating well-formed sentences requires both syntactic and semantic understanding of the Language. Understanding an image largely depends on obtaining image features. 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, the military, education, digital libraries, and web searching. Social media platforms 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.

Techniques: - The techniques used for this purpose can be broadly divided into two categories:

- (1) Traditional machine learning based techniques
- (2) Deep machine learning based techniques.

In traditional machine learning, hand crafted features such as Local Binary Patterns (LBP) Scale-Invariant Feature Transform (SIFT), the Histogram of Oriented Gradients (HOG) , and a combination of such features are widely used. In these techniques, features are extracted from input data. They are then passed to a classifier such as Support Vector Machines (SVM) in order to classify an object. Since hand crafted features are task specific, extracting features from a large and diverse set of data is not feasible. Moreover, real world data such as images and video are complex and have different semantic interpretations.

On the other hand, in deep machine learning based techniques, features are learned automatically from training data and they can handle a large and

diverse set of images and videos. For example, Convolutional Neural Networks (CNN) are widely used for feature learning, and a classifier such as Softmax is used for classification. CNN is generally followed by Recurrent Neural Networks (RNN) or Long Short-Term Memory Networks (LSTM) in order to generate captions. Deep learning algorithms can handle complexities and challenges of image captioning quite well

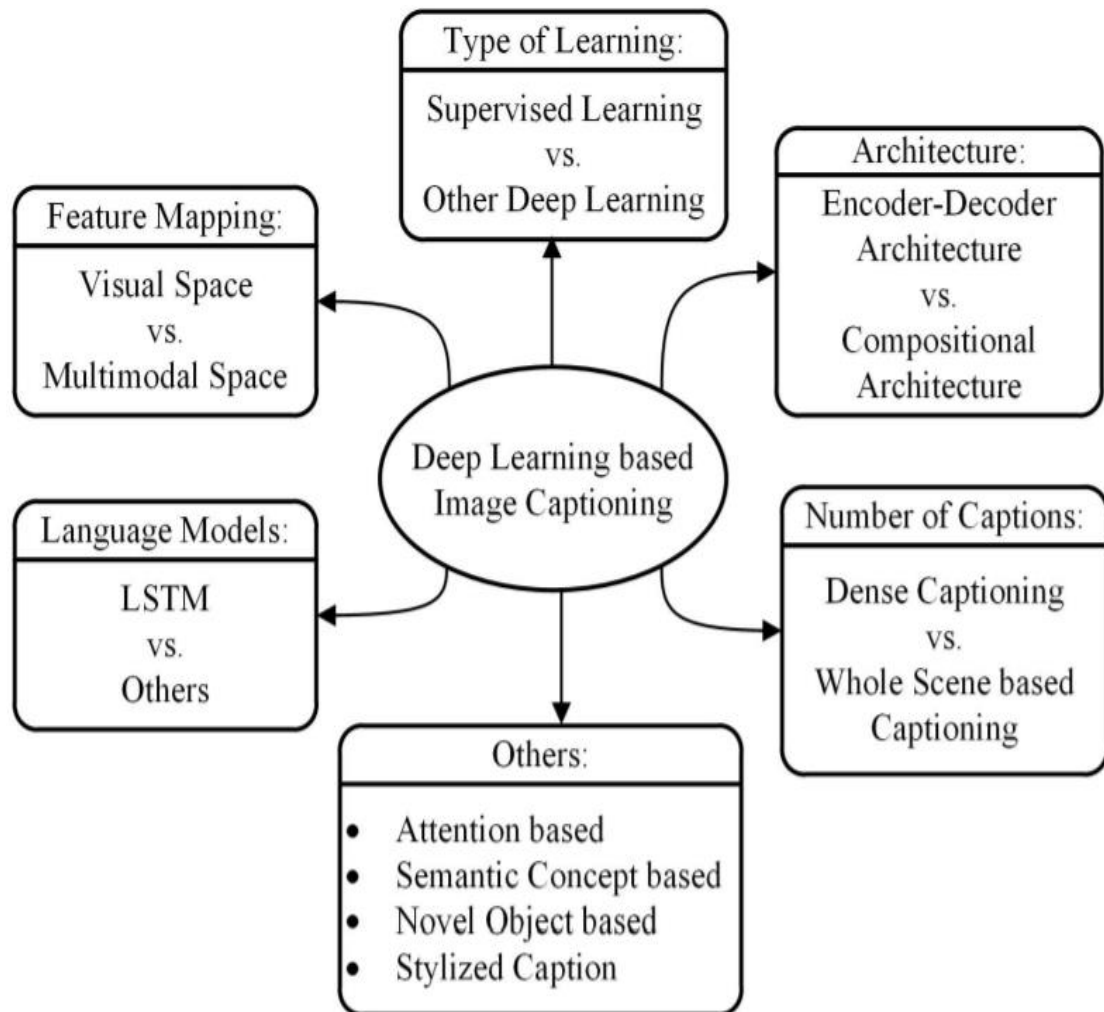


Figure.2 An overall taxonomy of deep learning-based image captioning

2. LITERATURE REVIEW

Image captioning has recently gathered a lot of attention specifically in the natural language domain. There is a pressing need for context based natural language description of images, however, this may seem a bit farfetched but recent developments in fields like neural networks, computer vision and natural language processing has paved a way for accurately describing images i.e. representing their visually grounded meaning. We are leveraging state-of-the-art techniques like Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and appropriate datasets of images and their human perceived description to achieve the same. We demonstrate that our alignment model produces results in retrieval experiments on datasets such as Flickr.

2.1 IMAGE CAPTIONING METHODS

There are various Image Captioning Techniques some are rarely used in present but it is necessary to take a overview of those technologies before proceeding ahead. The main categories of existing image captioning methods and they include template-based image captioning, retrieval-based image captioning, and novel caption generation. Novel caption generation-based image caption methods mostly use visual space and deep machine learning based techniques. Captions can also be generated from multimodal space. Deep learning-based image captioning methods can also be categorized on learning techniques: Supervised learning, Reinforcement learning, and unsupervised learning. We group the reinforcement learning and unsupervised learning into Other Deep Learning. Usually captions are generated for a whole scene in the image. However, captions can also be generated for different regions of an image (Dense captioning). Image captioning methods can use either simple Encoder-Decoder architecture or Compositional architecture. There are methods that use attention mechanism, semantic concept, and different styles in image descriptions. Some methods can also generate description for unseen objects. We group them into one category as "Others". Most of the image captioning methods use LSTM as language model. However, there are a number of methods that use other language models such as CNN and RNN. Therefore, we include a language model-based category as "LSTM vs. Others".

2.1.1 TEMPLATE-BASED APPROACHES

Template-based approaches have fixed templates with a number of blank slots to generate captions. In these approaches, different objects, attributes, actions are detected first and then the blank spaces in the templates are filled. For example, Farhadi et al. use a triplet of scene elements to fill the template slots for generating image captions. Li et al. extract the phrases related to detected objects, attributes and their relationships for this purpose. A Conditional Random Field (CRF) is adopted by Kulkarni et al. to infer the objects, attributes, and prepositions before filling in the gaps. Template-based methods can generate grammatically correct captions. However, templates are predefined and cannot generate variable-length captions. Moreover, later on, parsing based language models have been introduced in image captioning which are more powerful than fixed template-based methods. Therefore, in this paper, we do not focus on these template based methods.

2.1.2 RETRIEVAL-BASED APPROACHES

Captions can be retrieved from visual space and multimodal space. In retrieval-based approaches, captions are retrieved from a set of existing captions. Retrieval based methods first find the visually similar images with their captions from the training data set. These captions are called candidate captions. The captions for the query image are selected from these captions pool. These methods produce general and syntactically correct captions. However, they cannot generate image specific and semantically correct captions.

2.1.3 NOVEL CAPTION GENERATION

Novel image captions are captions that are generated by the model from a combination of the image features and a language model instead of matching to an existing captions. Generating novel image captions solves both of the problems of using existing captions and as such is a much more interesting and useful problem.

Novel captions can be generated from both visual space and multimodal space. A general approach of this category is to analyze the visual content of the image first

and then generate image captions from the visual content using a language model. These methods can generate new captions for each image that are semantically more accurate than previous approaches. Most novel caption generation methods use deep machine learning based techniques. Therefore, deep learning based novel image caption generating methods are our main focus in this literature.

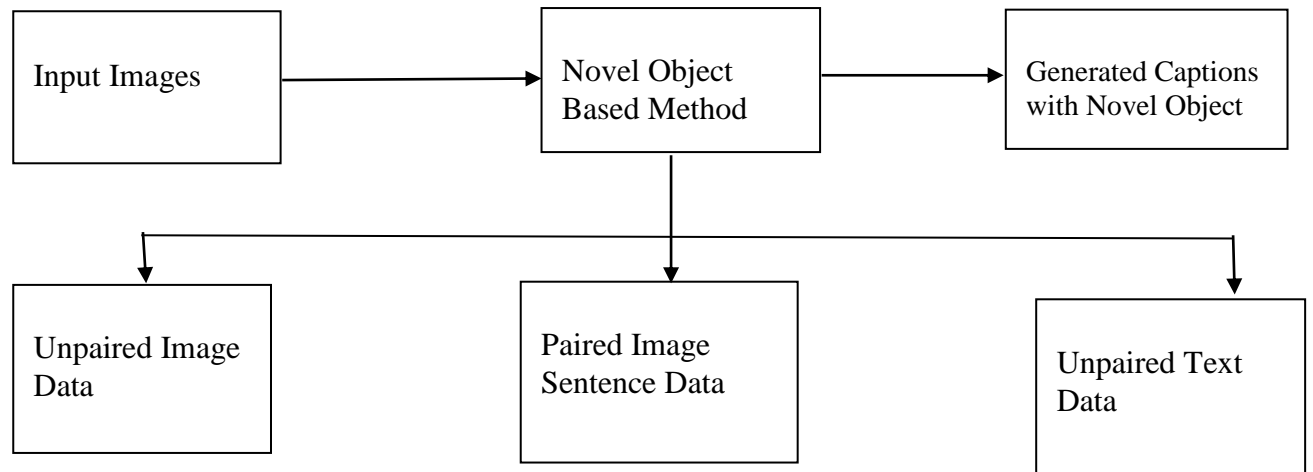


Figure. 3 NOVEL CAPTION GENERATION.

2.2 DEEP LEARNING BASED IMAGE CAPTIONING METHODS

We draw an overall taxonomy in Figure 2 for deep learning-based image captioning methods. We discuss their similarities and dissimilarities by grouping them into visual space vs. multimodal space, dense captioning vs. captions for the whole scene, Supervised learning vs. Other deep learning, Encoder-Decoder architecture vs. Compositional architecture, and one „Others“ group that contains Attention-Based, Semantic Concept-Based, Stylized captions, and Novel Object-Based captioning. We also create a category named LSTM vs. Others.

A brief overview of the deep learning-based image captioning methods is shown in table. It contains the name of the image captioning methods, the type of deep neural networks used to encode image information, and the language models used in

describing the information. In the final column, we give a category label to each captioning technique based on the taxonomy in Figure 2.

2.2.1 VISUAL SPACE VS. MULTIMODAL SPACE

Deep learning-based image captioning methods can generate captions from both visual space and multimodal space. Understandably image captioning datasets have the corresponding captions as text. In the visual space-based methods, the image features and the corresponding captions are independently passed to the language decoder. In contrast, in a multimodal space case, a shared multimodal space is learned from the images and the corresponding caption-text. This multimodal representation is then passed to the language decoder.

VISUAL SPACE

Bulk of the image captioning methods use visual space for generating captions. In the visual space-based methods, the image features and the corresponding captions are independently passed to the language decoder.

MULTIMODAL SPACE

The architecture of a typical multimodal space-based method contains a language Encoder part, a vision part, a multimodal space part, and a language decoder part. A general diagram of multimodal space-based image captioning methods is shown in Figure 4. The vision part uses a deep convolutional neural network as a feature extractor to extract the image features. The language encoder part extracts the word features and learns a dense feature embedding for each word. It then forwards the semantic temporal context to the recurrent layers. The multimodal space part maps the image features into a common space with the word features.

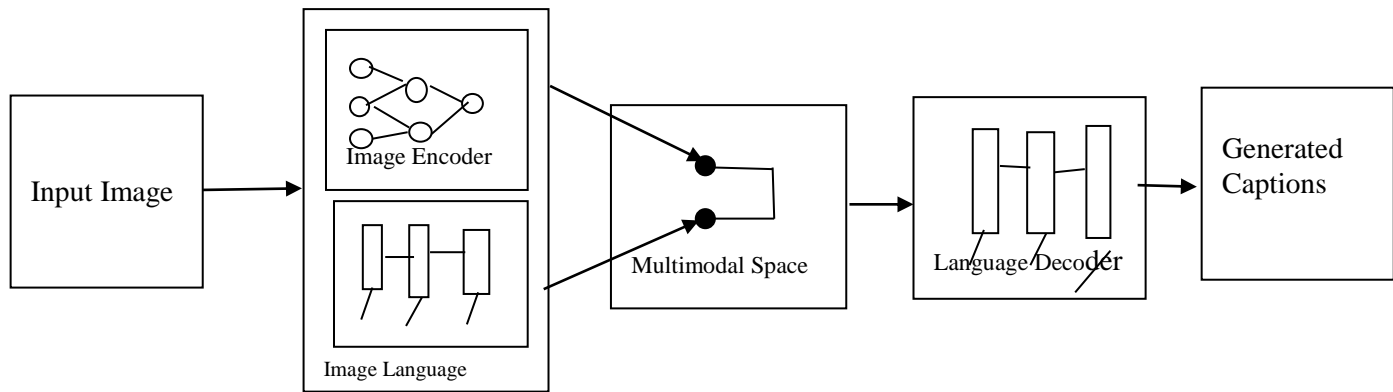


Figure 4: A block diagram of multimodal space-based image captioning.

2.3 SUPERVISED LEARNING VS. OTHER DEEP LEARNING

In supervised learning, training data come with desired output called label. Unsupervised learning, on the other hand, deals with unlabeled data. Reinforcement learning is another type of machine learning approach where the aims of an agent are to discover data and/or labels through exploration and a reward signal. A number of image captioning methods use reinforcement learning and GAN based approaches. These methods sit in the category of "Other Deep Learning". Generative Adversarial Networks (GANs) are a type of unsupervised learning

2.3.1 SUPERVISED LEARNING-BASED IMAGE CAPTIONING

Supervised learning-based networks have successfully been used for many years in image classification, object detection and attribute learning. This progress makes researchers interested in using them in automatic image captioning. In this paper, we have identified a large number of supervised learning-based image captioning methods. We classify them into different categories: (i) Encoder-Decoder Architecture, (ii) Compositional Architecture, (iii) Attention based, (iv) Semantic concept-based, (v) Stylized captions, (vi) Novel object-based, and (vii) Dense image captioning.

2.3.2 OTHER DEEP LEARNING-BASED IMAGE CAPTIONING

In our day to day life, data are increasing with unlabeled data because it is often

impractical to accurately annotate data. Therefore, recently, researchers are focusing more on reinforcement learning and unsupervised learning-based techniques for image captioning.

2.4 DENSE CAPTIONING VS. CAPTIONS FOR THE WHOLE SCENE

In dense captioning, captions are generated for each region of the scene. Other methods generate captions for the whole scene.

2.4.1 DENSE CAPTIONING

The previous image captioning methods can generate only one caption for the whole image. They use different regions of the image to obtain information of various objects. However, these methods do not generate region wise captions. Johnson proposed an image captioning method called Dense Cap. This method localizes all the salient regions of an image and then it generates descriptions for those regions.

A typical method of this category has the following steps:

- (1) Region proposals are generated for the different regions of the given image.
- (2) CNN is used to obtain the region-based image features.
- (3) The outputs of Step 2 are used by a language model to generate captions for every region. A block diagram of a typical dense captioning method is given.

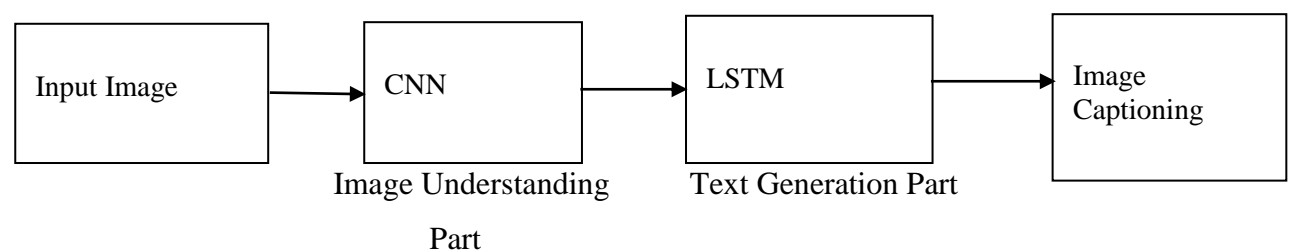


Figure5 .Block diagram of simple Encoder-Decoder architecture-based image captioning

2.4.2 CAPTIONS FOR THE WHOLE SCENE

Encoder-Decoder architecture, Compositional architecture, attention-based, semantic concept-based, stylized captions, Novel object-based image captioning, and other

deep learning networks-based image captioning methods generate single or multiple captions for the whole scene.

2.5 ENCODER-DECODER ARCHITECTURE VS. COMPOSITIONAL ARCHITECTURE

Some methods use just simple vanilla encoder and decoder to generate captions. However, other methods use multiple networks for it.

2.5.1 ENCODER-DECODER ARCHITECTURE-BASED IMAGE CAPTIONING

The neural network-based image captioning methods work as just simple end to end manner. These methods are very similar to the encoder-decoder framework-based neural machine translation .In this network, global image features are extracted from the hidden activations of CNN and then fed them into an LSTM to generate a sequence of words. A typical method of this category has the following general steps:

- (1) A vanilla CNN is used to obtain the scene type, to detect the objects and their relationships.
- (2) The output of Step 1 is used by a language model to convert them into words, combined phrases that produce an image captions. A simple block diagram of this category is given.

2.5.2 COMPOSITIONAL ARCHITECTURE-BASED IMAGE CAPTIONING

Compositional architecture-based methods composed of several independent functional building blocks: First, a CNN is used to extract the semantic concepts from the image. Then a language model is used to generate a set of candidate captions. In generating the final caption, these candidate captions are re-ranked using a deep multimodal similarity model.

A typical method of this category maintains the following steps:

- (1) Image features are obtained using a CNN.

- (2) Visual concepts (e.g. attributes) are obtained from visual features.
- (3) Multiple captions are generated by a language model using the information of Step 1 and Step 2.
- (4) The generated captions are re-ranked using a deep multimodal similarity model to select high quality image captions.

A common block diagram of compositional network-based image captioning methods is given in Figure.

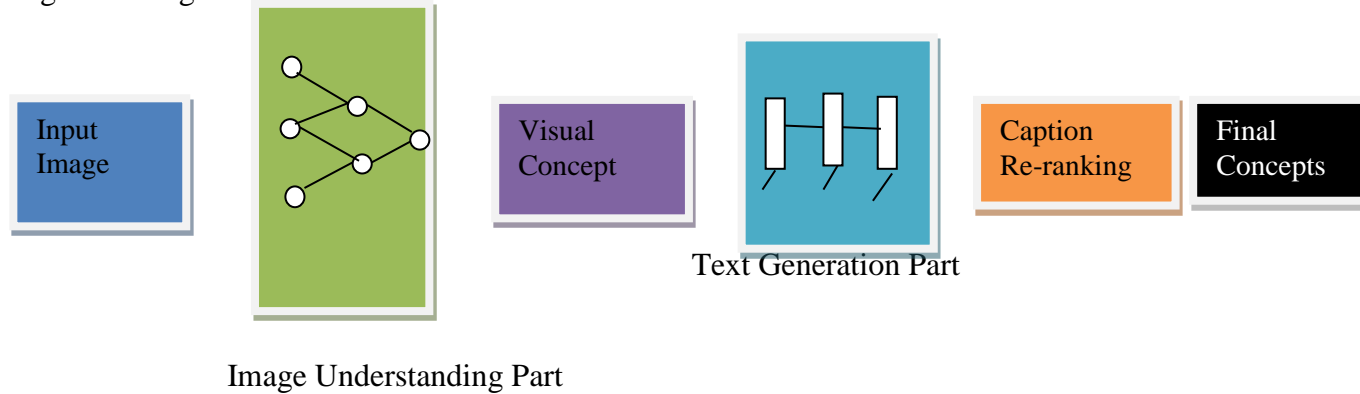


Figure. 6 . A block diagram of a compositional network-based captioning

2.6 LSTM VS. OTHERS

Image captioning intersects computer vision and natural language processing (NLP) research. NLP tasks, in general, can be formulated as a sequence to sequence learning. Several neural language models such as neural probabilistic language model, log-bilinear models, skip-gram models, and recurrent neural networks (RNNs) have been proposed for learning sequence to sequence tasks. RNNs have widely been used in various sequence learning tasks. However, traditional RNNs suffer from vanishing and exploding gradient problems and cannot adequately handle long-term temporal dependencies. LSTM networks are a type of RNN that has special units in addition to standard units. LSTM units use a memory cell that can maintain information in memory for long periods of time. In recent years, LSTM based models have dominantly been used in sequence to sequence learning tasks. Another network, Gated Recurrent Unit (GRU) has a similar structure to LSTM but it does not use separate memory cells and uses fewer gates to control the flow of information. However, LSTMs ignore the underlying hierarchical structure of a sentence. They

also require significant storage due to long-term dependencies through a memory cell. In contrast, CNNs can learn the internal hierarchical structure of the sentences and they are faster in processing than LSTMs. Therefore, recently, convolutional architectures are used in other sequence to sequence tasks, e.g., conditional image generation and machine translation. Inspired by the above success of CNNs in sequence learning tasks, GU proposed a CNN language model-based image captioning method. This method uses a language-CNN for statistical language modelling. However, the method cannot model the dynamic temporal behavior of the language model only using a language-CNN. It combines a recurrent network with the language CNN to model the temporal dependencies properly. Aneja proposed a convolutional architecture for the task of image captioning. They use a feedforward network without any recurrent function. The architecture of the method has four components:

- (i) input embedding layer
- (ii) image embedding layer
- (iii) convolutional module,
- (iv) Output embedding layer.

It also uses an attention mechanism to leverage spatial image features. They evaluate their architecture on the challenging MSCOCO dataset and shows comparable performance to an LSTM based method on standard metrics.

3. PROBLEM FORMULATION

3.1 PROBLEM IDENTIFICATION

Despite the successes of many systems based on the Recurrent Neural Networks (RNN) many issues remain to be addressed. Among those issues the following two are prominent for most systems.

1. The Vanishing Gradient Problem.
2. Training an RNN is a very difficult task.

A recurrent neural network is a deep learning algorithm designed to deal with a variety of complex computer tasks such as object classification and speech detection. RNNs are designed to handle a sequence of events that occur in succession, with the understanding of each event based on information from previous events. Ideally, we would prefer to have the deepest RNNs so they could have a longer memory period and better capabilities. These could be applied for many real-world use-cases such as stock prediction and enhanced speech detection. However, while they sound promising, RNNs are rarely used for real-world scenarios because of the vanishing gradient problem.

This is one of the most significant challenges for RNNs performance. In practice, the architecture of RNNs restricts its long-term memory capabilities, which are limited to only remembering a few sequences at a time. Consequently, the memory of RNNs is only useful for shorter sequences and short time-periods.

3.1.1 Vanishing Gradient problem arises while training an Artificial Neural Network. This mainly occurs when the network parameters and hyper parameters are not properly set. The vanishing gradient problem restricts the memory capabilities of traditional RNNs—adding too many time-steps increases the chance of facing a gradient problem and losing information when you use back propagation.

4. PROPOSED WORK

In order to tackle the image captioning task, recent work shows it is in one's interest to utilize neural networks. This frequently used term dates back to 1950s when notions such as the Perceptron Learning Algorithm were introduced. Modern neural networks draw on notions discovered in the era of a Perceptron. In this section, we first define a neuron as a fundamental part of modern neural networks. Then we elaborate on Convolutional Networks and Recurrent Networks.

The main aim of this project is to get a little bit of knowledge of deep learning techniques. We use two techniques mainly CNN and LSTM for image classification. So, to make our image caption generator model, we will be merging these architectures. It is also called a CNN-RNN model.

- CNN is used for extracting features from the image. We will use the pre-trained model exception.
- LSTM will use the information from CNN to help generate a description of the image.

4.1 Convolutional Neural Network

Convolutional Neural Networks (ConvNets or CNNs) are a category of Artificial Neural Networks which have proven to be very effective in the field of image recognition and classification. They have been used extensively for the task of object detection, self-driving cars, image captioning etc. First convnet was discovered in the year 1990 by Yann Lecun and the architecture of the model was called as the LeNet architecture. A basic convnet is shown in the fig. below

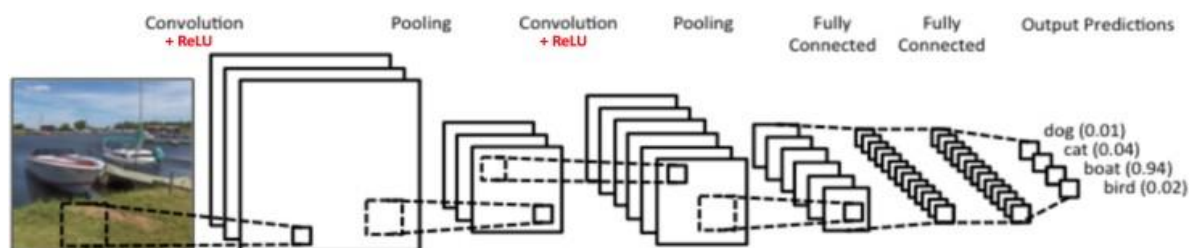


Fig. 7: A simple convnet architecture

The entire architecture of a convnet can be explained using four main operations namely,

1. Convolution
2. Non- Linearity(ReLU)
3. Pooling or Subsampling
4. Classification (Fully Connected Layer)

These operations are the basic building blocks of every Convolutional Neural Network, so understanding how these work is an important step to developing a sound understanding of ConvNets. We will discuss each of these operations in detail below.

Essentially, every image can be represented as a matrix of pixel values. An image from a standard digital camera will have three channels—red, green and blue—you can imagine those as three 2d-matrices stacked over each other (one for each color), each having pixel values in the range 0 to255.

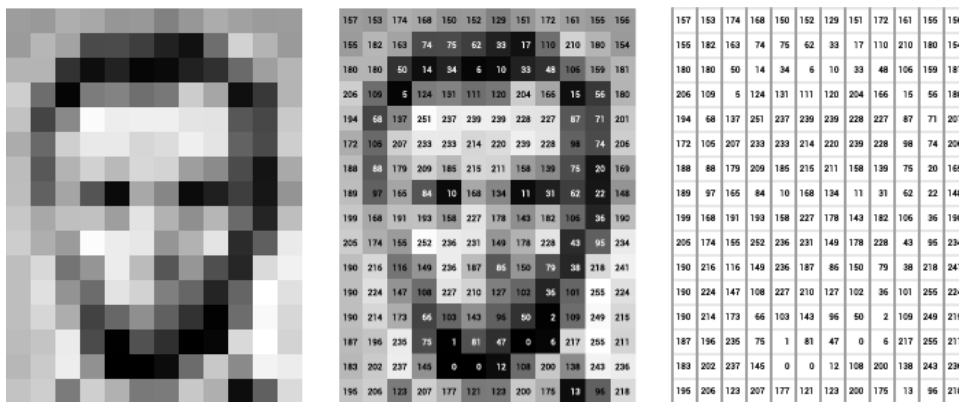


Fig. 8: A grayscale image as matrix of numbers

4.1.1 Convolution Operator

The purpose of convolution operation is to extract features from an image.

We consider filters of size smaller than the dimensions of image. The entire operation of convolution can be understood with the example below.

Consider a small 2-dimensional 5*5 image with binary pixel values.

Consider another 3*3 matrix shown in Fig. 9.

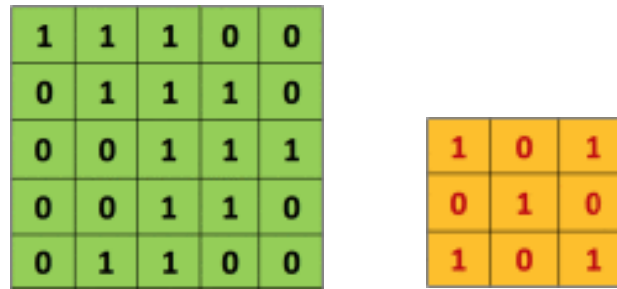


Fig. 9: Image (in green) and Filter (in orange)

We slide this orange 3*3 matrix over the original image by 1 pixel and calculate element-wise multiplication of the orange matrix with the sub-matrix of the original image and add the final multiplication outputs to get the final integer which forms a single element of the output matrix which is shown in the Fig. 10 by the pink matrix.

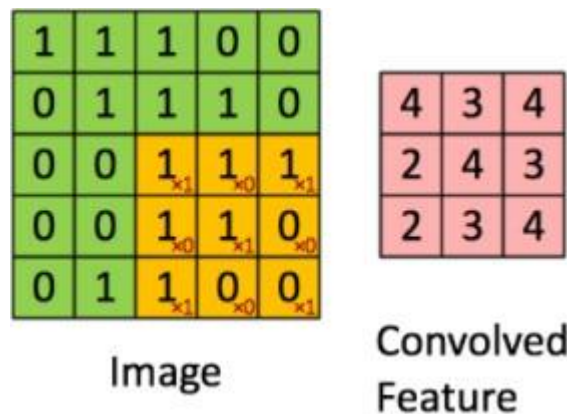


Fig. 10: Convolution operation

The 3*3 matrix is called a filter or kernel or feature detector and the matrix formed by sliding the filter over the image and computing the dot product is called the Convolved Feature or Activation Map or the Feature Map. The number of pixels by which we slide the filter over the original image is known as stride.

4.1.2 Introducing Non-Linearity

An additional operation is applied after every convolution operation. The most commonly used non-linear function for images is the ReLU which stands for

Rectified Linear Unit. The ReLU operation is an element-wise operation which replaces the negative pixels in the images with a zero

Since most of the operations in real-life relate to non-linear data but the output of convolution operation is linear because the operation applied is elementwise multiplication and addition. The output of the ReLU operation is shown in the figure below.

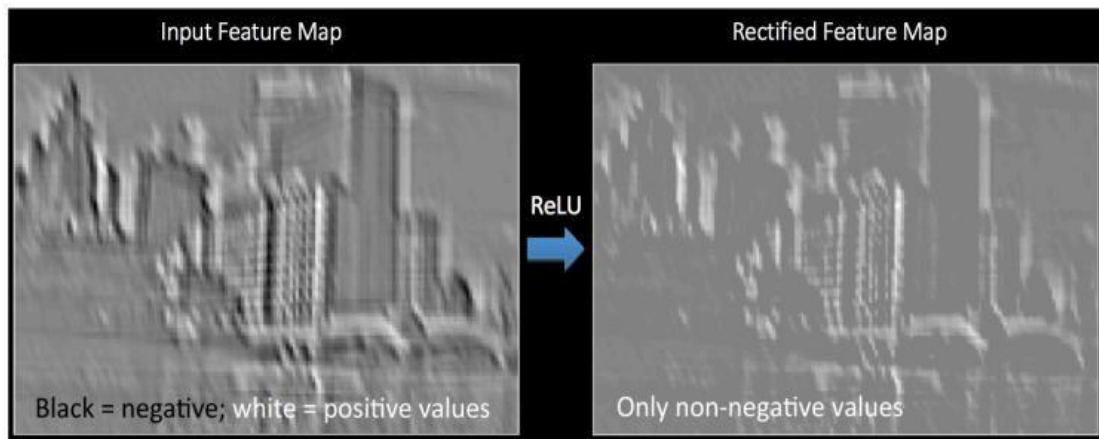


Fig. 11: Output after a ReLU operation

4.1.3 Spatial Pooling

The pooling operation reduces the dimensionality of the image but preserves the important features in the image. The most common type of pooling technique used is max pooling. In max pooling you slide a window of $n \times n$ where n is less than the side of the image and determine the maximum in that window and then shift the window with the given stride length. The complete process is specified by the fig.

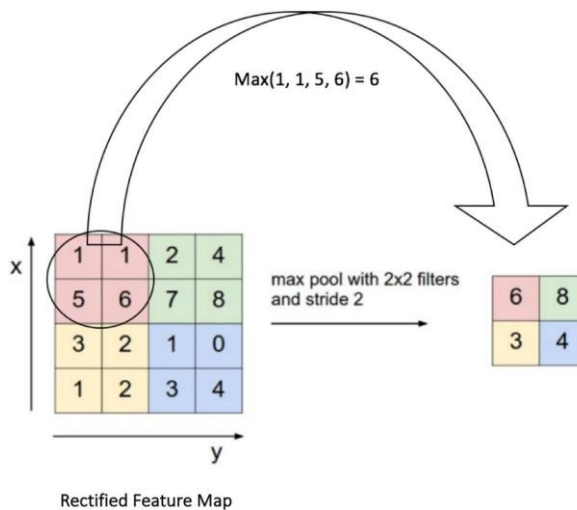


Fig. 12: Max pooling operation

4.1.4 Fully-Connected layer

The fully connected layer is the multi-layer perceptron that uses the SoftMax activation function in the output layer. The term “fully-connected” refers to the fact that all the neurons in the previous layer are connected to all the neurons of the next layer. The convolution and pooling operation generate features of an image. The task of the fully connected layer is to map these feature vectors to the classes in the training data.

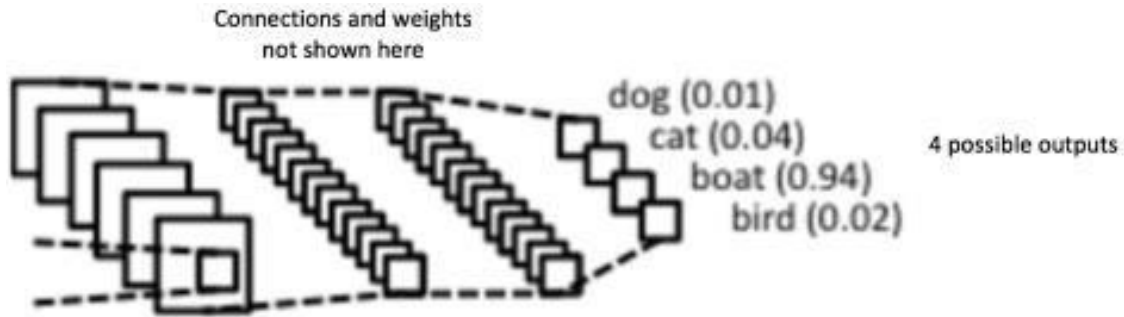


Fig. 13: An example of fully connected layer of data with 4 classes

The task of image classification on cifar-10 has shown state of the art results with the use of convnets. We use the alex net architecture proposed by Alex krizhevsky with a few tweaks. Alexnet is trained for images having 224*224 dimensions and hence need to be modified to be used for cifar-10 since the images in cifar-10 are 32*32. The model used by us has alternate layers of convolution and non-linearities. We use a fully connected layer at the end which uses softmax activation to give the scores of the 10 classes present in the cifar-10 dataset.

The dataset on these convnets yield an accuracy of 85% within around 1.5 hrs of training on gpus. The plots of loss and accuracy on test and validation set are shown in the figures below.

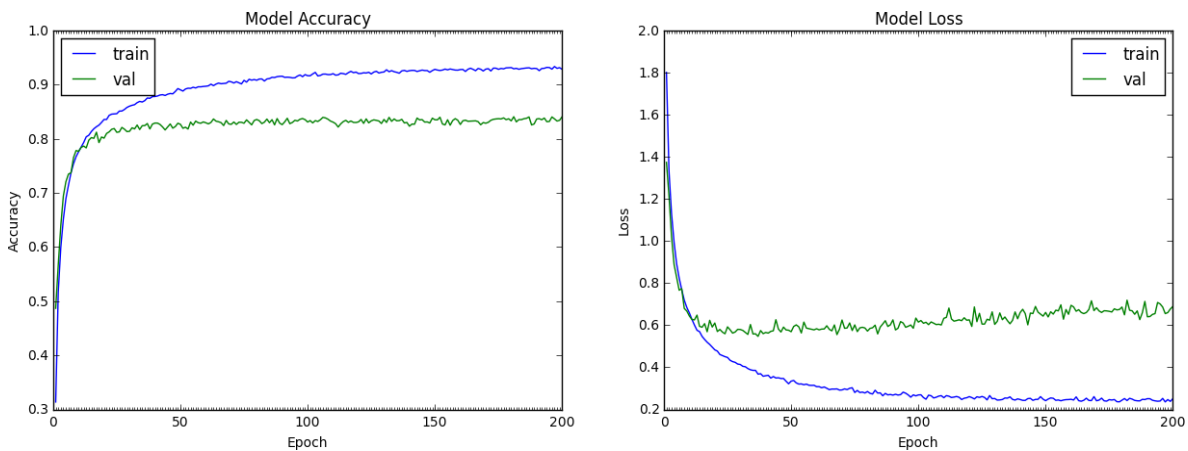


Fig. 14: Accuracy and loss plot on training and validation set

The model is built using tensor flow. Tensorflow is an open source library developed by Google brain team for machine learning. Though being a python api, most of the code of tensor flow is written in C++ and CUDA which is nvidia's programming language for GPU'S. This helps tensorflow in faster execution of code since python is slower than CPP. Also, the use of GPU enhances the performance of the code significantly.

4.3 LONG SHORT TERM MEMORY

LSTM stands for Long short term memory, they are a type of RNN (recurrent neural network) which is well suited for sequence prediction problems. Based on the previous text, we can predict what the next word will be. It has proven itself effective from the traditional RNN by overcoming the limitations of RNN which had short term memory. LSTM can carry out relevant information throughout the processing of inputs and with a forget gate, it discards non-relevant information. LSTMs are designed to overcome the vanishing gradient problem and allow them to retain information for longer periods compared to traditional RNNs. LSTMs can maintain a constant error, which allows them to continue learning over numerous time-steps and back propagate through time and layers.

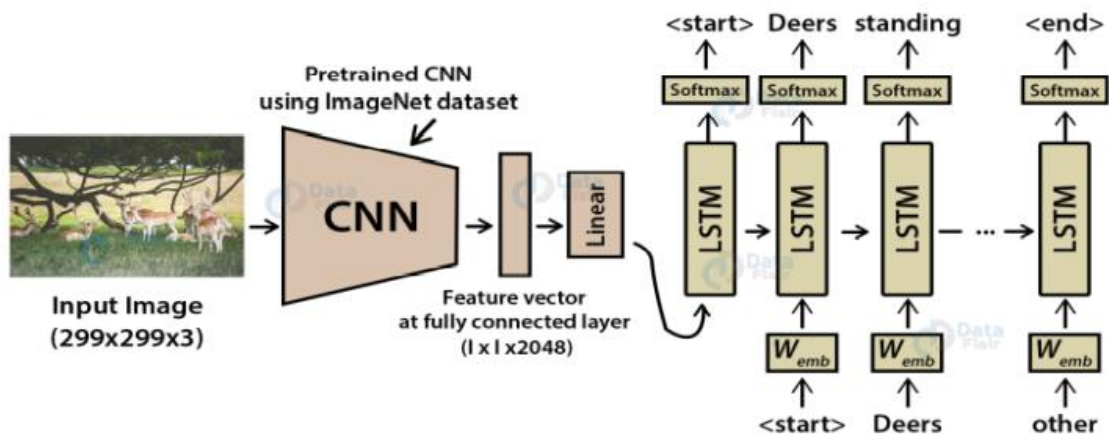


Figure 15. Model, Image Caption Generator

LSTMs use gated cells to store information outside the regular flow of the RNN. With these cells, the network can manipulate the information in many ways, including storing information in the cells and reading from them. The cells are individually capable of making decisions regarding the information and can execute these decisions by opening or closing the gates. The ability to retain information for a long period of time gives LSTM the edge over traditional RNNs in these tasks.

The chain-like architecture of LSTM allows it to contain information for longer time periods, solving challenging tasks that traditional RNNs struggle to or simply cannot solve.

The three major parts of the LSTM include:

Forget gate—removes information that is no longer necessary for the completion of the task. This step is essential to optimizing the performance of the network.

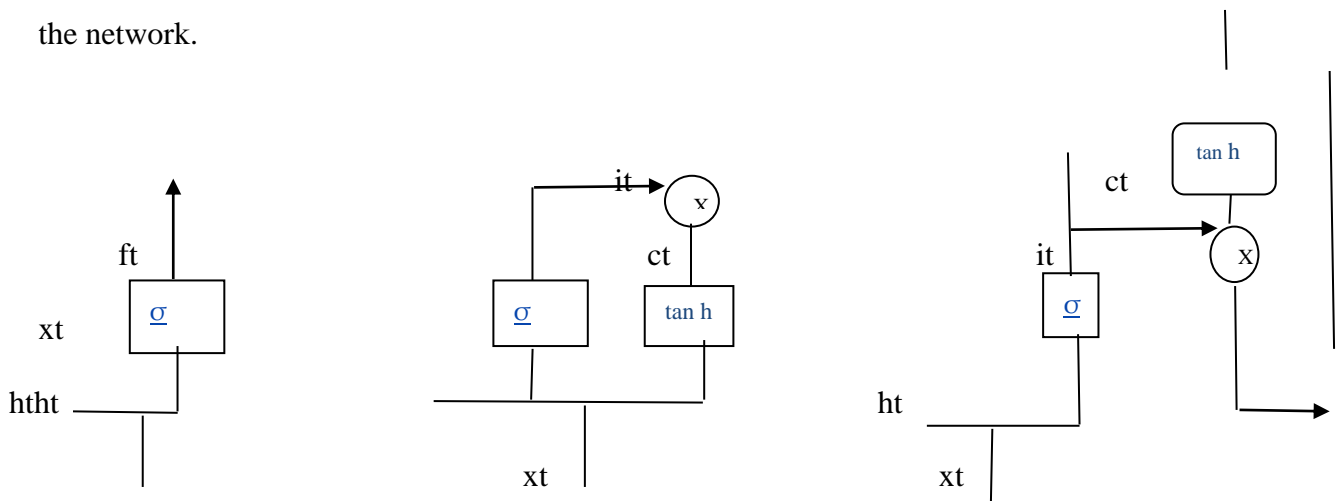


Figure.16 ForgetGate ,Input Gate, Output Gate

Input gate—responsible for adding information to the cells

Output gate—selects and outputs necessary information.

The CNN LSTM architecture involves using Convolutional Neural Network (CNN) layers for feature extraction on input data combined with LSTMs to support sequence prediction. This architecture was originally referred to as a Long-term Recurrent Convolutional Network or LRCN model, although we will use the more generic name “CNN LSTM” to refer to LSTMs that use a CNN as a front end in this lesson.

This architecture is used for the task of generating textual descriptions of images. Key is the use of a CNN that is pre-trained on a challenging image classification task that is re-purposed as a feature extractor for the caption generating problem.

5. SYSTEM DESIGN

This project requires a dataset which have both images and their caption. The dataset should be able to train the image captioning model.

5.1 FLICKR8K DATASET

Flickr8k dataset is a public benchmark dataset for image to sentence description. This dataset consists of 8000 images with five captions for each image. These images are extracted from diverse groups in Flickr website. Each caption provides a clear description of entities and events present in the image. The dataset depicts a variety of events and scenarios and doesn't include images containing well-known people and places which makes the dataset more generic. The dataset has 6000 images in training dataset, 1000 images in development dataset and 1000 images in test dataset.

Features of the dataset making it suitable for this project are:

- Multiple captions mapped for a single image makes the model generic and avoids over fitting of the model.
- Diverse category of training images can make the image captioning model to work for multiple categories of images and hence can make the model more robust.

5.2 IMAGE DATA PREPARATION

The image should be converted to suitable features so that they can be trained into a deep learning model. Feature extraction is a mandatory step to train any image in deep learning model. The features are extracted using Convolutional Neural Network (CNN) with Visual Geometry Group (VGG-16) model. This model also won Image Net Large Scale Visual Recognition Challenge in 2015 to classify the images into one among the 1000 classes given in the challenge. Hence, this model is ideal to use for this project as image captioning requires identification of images. In VGG-16, there are 16 weight layers in the network and the deeper number of layers help in better feature extraction from images.

The VGG-16 network uses 3*3 convolutional layers making its architecture simple and uses max pooling layer in between to reduce volume size of the image. The last layer of the image which predicts the classification is removed and the internal representation of image just before classification is returned as feature. The dimension of the input image should be 224*224 and this model extracts features of the image and returns a 1- dimensional 4096 element vector.

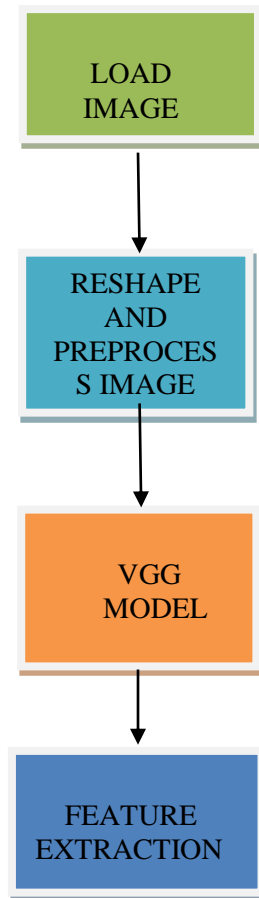


Figure 17 : Feature Extraction in Image using VGG

5.3 CAPTION DATA PREPARATION

Flickr8k dataset contains multiple descriptions described for a single image. In the data preparation phase, each image id is taken as key and its corresponding captions are stored as values in a dictionary.

5.3.1 DATA CLEANING

In order to make the text dataset work in machine learning or deep learning models, raw text should be converted to a usable format.

The following text cleaning steps are done before using it for the project:

- Removal of punctuations.
- Removal of numbers.
- Removal of single length words.
- Conversion of uppercase to lowercase characters.

Stop words are not removed from the text data as it will hinder the generation of a grammatically complete caption which is needed for this project. Table 1 shows samples of captions after data cleaning.

Original Captions	Captions after Data cleaning
Two people are at the edge of a lake, facing the water and the city skyline.	Two people are at the edge of lake facing the water and the city skyline
A little girl rides in a child's swing.	Little girl rides in child swing
Two boys posing in blue shirts and khaki shorts.	Two boys posing in blue shirts and khaki shorts

Table : Data cleaning of caption

6. IMPLEMENTATION

6.1 PRE-REQUISITES

This project requires good knowledge of Deep learning, Python, working on Jupyter notebooks, Keras library, Numpy, and *Natural language processing*.

Make sure you have installed all the following necessary libraries:

- tensorflow
- keras
- pillow
- numpy
- tqdm
- jupyterlab
- Google collab

6.2 PROJECT FILESTRUCTURE

Downloaded from dataset:

- **Flicker8k_Dataset** – Dataset folder which contains 8091 images.
- **Flickr_8k_text** – Dataset folder which contains text files and captions of images. The below files will be created by us while making the project.
- **Models** – It will contain our trained models.
- **Descriptions.txt** – This text file contains all image names and their captions after preprocessing.
- **Features.p** – Pickle object that contains an image and their feature vector extracted from the Xception pre-trained CNN model.
- **Tokenizer.p** – Contains tokens mapped with an index value.
- **Model.png** – Visual representation of dimensions of our project.

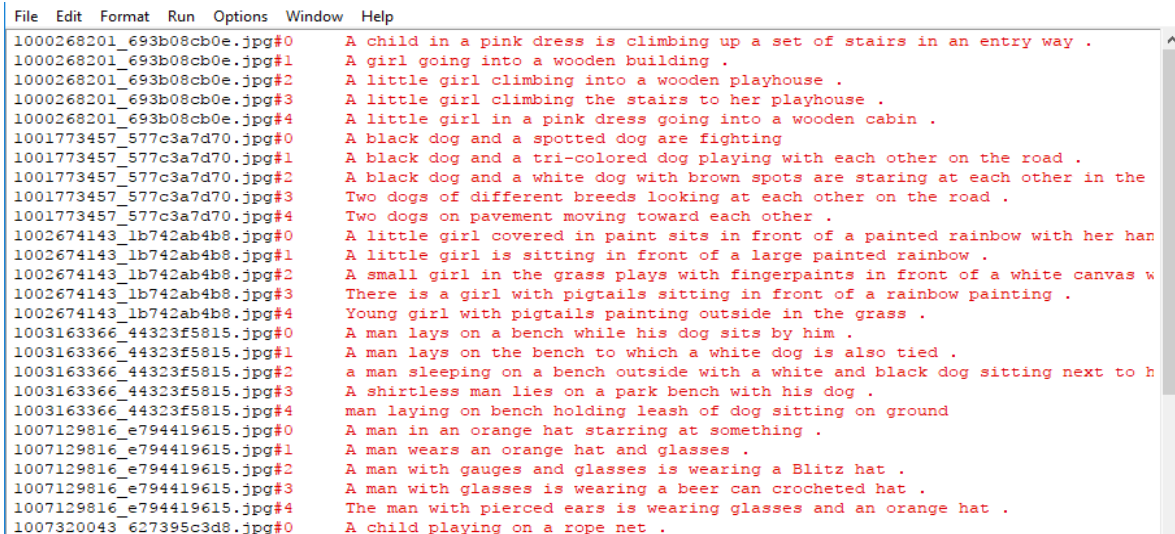
- **Testing_caption_generator.py** – Python file for generating a caption of any image.
- **Training_caption_generator.ipynb** – Jupyter notebook in which we train and build our image caption generator.

6.3 BUILDING THE PYTHON BASED PROJECT

Let's start by initializing the jupyter notebook server by typing `jupyter lab` in the console of your project folder. It will open up the interactive Python notebook where you can run your code. Create a Python3 notebook and name it `training_caption_generator.ipynb`.

6.3.1 GETTING AND PERFORMING DATA CLEANING

The main text file which contains all image captions is `Flickr8k.token` in our `Flickr_8k_text` folder.



```
File Edit Format Run Options Window Help
1000268201_693b08cb0e.jpg#0 A child in a pink dress is climbing up a set of stairs in an entry way .
1000268201_693b08cb0e.jpg#1 A girl going into a wooden building .
1000268201_693b08cb0e.jpg#2 A little girl climbing into a wooden playhouse .
1000268201_693b08cb0e.jpg#3 A little girl climbing the stairs to her playhouse .
1000268201_693b08cb0e.jpg#4 A little girl in a pink dress going into a wooden cabin .
1001773457_577c3a7d70.jpg#0 A black dog and a spotted dog are fighting
1001773457_577c3a7d70.jpg#1 A black dog and a tri-colored dog playing with each other on the road .
1001773457_577c3a7d70.jpg#2 A black dog and a white dog with brown spots are staring at each other in the
1001773457_577c3a7d70.jpg#3 Two dogs of different breeds looking at each other on the road .
1001773457_577c3a7d70.jpg#4 Two dogs on pavement moving toward each other .
1002674143_1b742ab4b8.jpg#0 A little girl covered in paint sits in front of a painted rainbow with her han
1002674143_1b742ab4b8.jpg#1 A little girl is sitting in front of a large painted rainbow .
1002674143_1b742ab4b8.jpg#2 A small girl in the grass plays with fingerpaints in front of a white canvas w
1002674143_1b742ab4b8.jpg#3 There is a girl with pigtails sitting in front of a rainbow painting .
1002674143_1b742ab4b8.jpg#4 Young girl with pigtails painting outside in the grass .
1003163366_44323f5815.jpg#0 A man lays on a bench while his dog sits by him .
1003163366_44323f5815.jpg#1 A man lays on the bench to which a white dog is also tied .
1003163366_44323f5815.jpg#2 a man sleeping on a bench outside with a white and black dog sitting next to h
1003163366_44323f5815.jpg#3 A shirtless man lies on a park bench with his dog .
1003163366_44323f5815.jpg#4 man laying on bench holding leash of dog sitting on ground
1007129816_e794419615.jpg#0 A man in an orange hat starring at something .
1007129816_e794419615.jpg#1 A man wears an orange hat and glasses .
1007129816_e794419615.jpg#2 A man with gauges and glasses is wearing a Blitz hat .
1007129816_e794419615.jpg#3 A man with glasses is wearing a beer can crocheted hat .
1007129816_e794419615.jpg#4 The man with pierced ears is wearing glasses and an orange hat .
1007320043_627395c3d8.jpg#0 A child playing on a rope net .
```

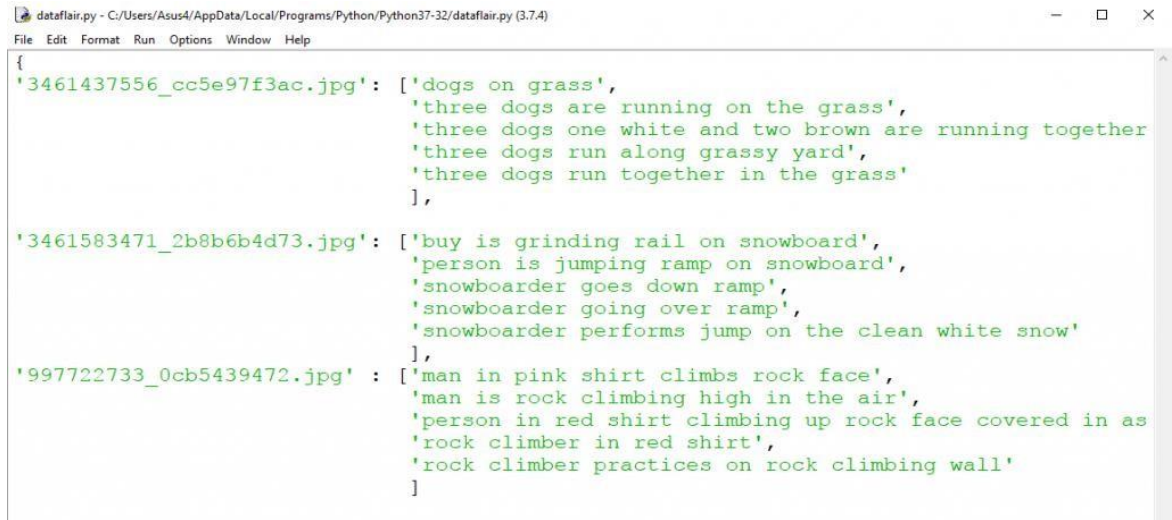
Figure.18. Flickr Dataset text format

The format of our file is image and caption separated by a new line (“\n”).

Each image has 5 captions and we can see that #(0 to 5)number is assigned

for each caption. We will define 5 functions:

- **load_doc(filename)** – For loading the document file and reading the contents inside the file into a string.
- **all_img_captions(filename)** – This function will create a **descriptions** dictionary that maps images with a list of 5 captions. The descriptions dictionary will look something like the Figure.



```
dataflair.py - C:/Users/Asus4/AppData/Local/Programs/Python/Python37-32/dataflair.py (3,7,4)
File Edit Format Run Options Window Help
{
  '3461437556_cc5e97f3ac.jpg': ['dogs on grass',
                                'three dogs are running on the grass',
                                'three dogs one white and two brown are running together',
                                'three dogs run along grassy yard',
                                'three dogs run together in the grass'
                                ],
  '3461583471_2b8b6b4d73.jpg': ['boy is grinding rail on snowboard',
                                'person is jumping ramp on snowboard',
                                'snowboarder goes down ramp',
                                'snowboarder going over ramp',
                                'snowboarder performs jump on the clean white snow'
                                ],
  '997722733_0cb5439472.jpg': ['man in pink shirt climbs rock face',
                                'man is rock climbing high in the air',
                                'person in red shirt climbing up rock face covered in as',
                                'rock climber in red shirt',
                                'rock climber practices on rock climbing wall'
                                ]
}
```

Figure 19. Flickr Dataset Python File

- **cleaning_text(descriptions)** – This function takes all descriptions and performs data cleaning. This is an important step when we work with textual data, according to our goal, we decide what type of cleaning we want to perform on the text. In our case, we will be removing punctuations, converting all text to lowercase and removing words that contain numbers. So, a caption like “A man riding on a three-wheeled wheelchair” will be transformed into “man riding on three wheeled wheelchair”
- **text_vocabulary(descriptions)** – This is a simple function that will separate all the unique words and create the vocabulary from all the descriptions.

- **save_descriptions(descriptions, filename)** – This function will create a list of all the descriptions that have been preprocessed and store them into a file. We will create a descriptions.txt file to store all the captions. It will look something likethis:

```

File Edit Format Run Options Window Help
1000268201_693b08cb0e.jpg child in pink dress is climbing up set of stairs ir
1000268201_693b08cb0e.jpg girl going into wooden building
1000268201_693b08cb0e.jpg little girl climbing into wooden playhouse
1000268201_693b08cb0e.jpg little girl climbing the stairs to her playhouse
1000268201_693b08cb0e.jpg little girl in pink dress going into wooden cabin
1001773457_577c3a7d70.jpg black dog and spotted dog are fighting
1001773457_577c3a7d70.jpg black dog and tricolored dog playing with each othe
1001773457_577c3a7d70.jpg black dog and white dog with brown spots are starir
1001773457_577c3a7d70.jpg two dogs of different breeds looking at each other
1001773457_577c3a7d70.jpg two dogs on pavement moving toward each other
1002674143_lb742ab4b8.jpg little girl covered in paint sits in front of paint
1002674143_lb742ab4b8.jpg little girl is sitting in front of large painted re
1002674143_lb742ab4b8.jpg small girl in the grass plays with fingerpaints in
1002674143_lb742ab4b8.jpg there is girl with pigtails sitting in front of rai
1002674143_lb742ab4b8.jpg young girl with pigtails painting outside in the gr
1003163366_44323f5815.jpg man lays on bench while his dog sits by him

```

Figure.20. Description of Images

6.3.2 EXTRACTING THE FEATURE VECTOR FROM ALLIMAGES

This technique is also called transfer learning, we don “have to do everything on our own, we use the pre-trained model that have been already trained on large datasets and extract the features from these models and use them for our tasks. We are using the Xception model which has been trained on imagenet dataset that had 1000 different classes to classify. We can directly import this model from the keras.applications . Make sure you are connected to the internet as the weights get automatically downloaded. Since the Exception model was originally built for image net, we will do little changes for integrating with our model. One thing to notice is that the Exception model takes 299*299*3 image size as input. We will remove the last classification layer and get the 2048 feature vector.

```
model =Exception( include_top=False, pooling="avg" )
```

The function **extract_features()** will extract features for all images and we will

map image names with their respective feature array. Then we will dump the features dictionary into a “features.p” pickle file.

This process can take a lot of time depending on your system. I am using an Nvidia 1050 GPU for training purpose so it took me around 7 minutes for performing this task. However, if you are using CPU then this process might take 1-2 hours. You can comment out the code and directly load the features **from our** pickle file.

6.3.3 LOADING DATASET FOR TRAINING THE MODEL

In our **Flickr_8k_test** folder, we have **Flickr_8k.trainImages.txt** file that contains a list of 6000 image names that we will use for training.

For loading the training dataset, we need more functions:

- **load_photos (filename)** – This will load the text file in a string and will return the list of image names.
- **load_clean_descriptions (filename, photos)** – This function will create a dictionary that contains captions for each photo from the list of photos. We also append the <start> and <end> identifier for each caption. We need this so that our LSTM model can identify the starting and ending of the caption.
- **load_features (photos)** – This function will give us the dictionary for image names and their feature vector which we have previously extracted from the exception model.

6.3.4 TOKENIZING THE VOCABULARY

Computers don't understand English words, for computers, we will have to represent them with numbers. So, we will map each word of the vocabulary with a unique index value. Keras library provides us with the tokenizer function that we will use to create tokens from our vocabulary and save them to a “tokenizer.p” pickle file.

Our vocabulary contains 7577 words. We calculate the maximum length of the descriptions. This is important for deciding the model structure parameters. Max_length of description is 32.

6.3.5 Create Data generator

Let us first see how the input and output of our model will look like. To make this task into a supervised learning task, we have to provide input and output to the model for training. We have to train our model on 6000 images and each image will contain 2048 length feature vector and caption is also represented as numbers. This amount of data for 6000 images is not possible to hold into memory so we will be using a generator method that will yield batches.

For example:

The input to our model is [x1, x2] and the output will be y, where x1 is the 2048 feature vector of that image, x2 is the input text sequence and y is the output text sequence that the model has to predict.

x1(feature vector)	x2(Text sequence)	y(word to predict)
feature	start,	two
feature	start, two	dogs
feature	start, two, dogs	drink
feature	start, two, dogs, drink	water
feature	start, two, dogs, drink, water	end

Table . Word Prediction Generation Step By Step

6.3.6 Defining the CNN-RNN model

To define the structure of the model, we will be using the Keras Model from Functional API. It will consist of three major parts:

- **Feature Extractor** – The feature extracted from the image has a size of 2048, with a dense layer, we will reduce the dimensions to 256 nodes.
- **Sequence Processor** – An embedding layer will handle the textual input, followed by the LSTM layer.
- **Decoder** – By merging the output from the above two layers, we will process by the dense layer to make the final prediction. The final layer will contain the number of nodes equal to our vocabulary size.

Visual representation of the final model is given in the figure

6.3.7 Training the model

To train the model, we will be using the 6000 training images by generating the input and output sequences in batches and fitting them to the model using `model.fit_generator()` method. We also save the model to our models folder. This will take some time depending on your system capability.

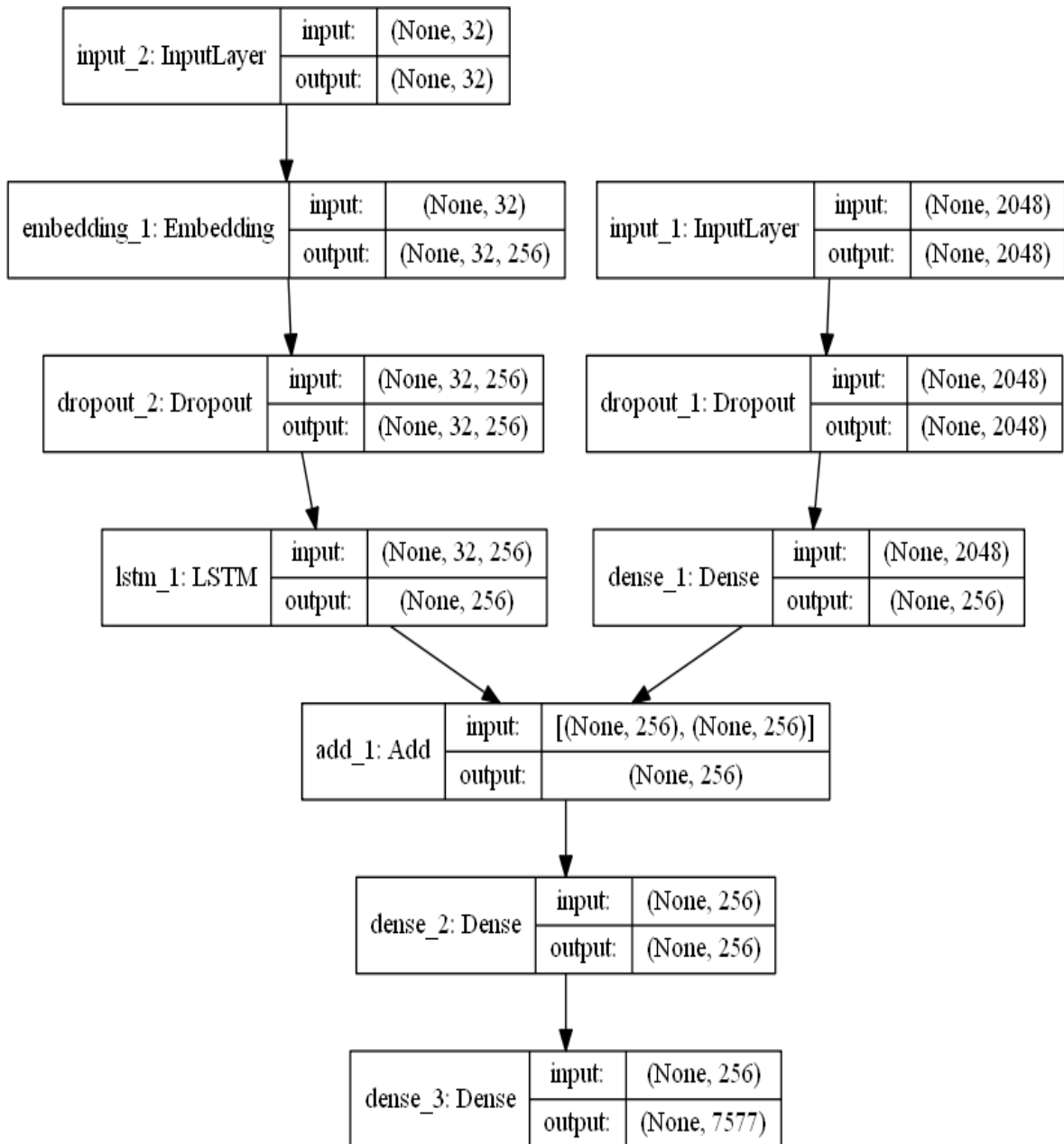


Figure.21. Final Model Structure

6.3.8 Testing the model

The model has been trained, now, we will make a separate file `testing_caption_generator.py` which will load the model and generate predictions. The predictions contain the max length of index values so we will use the same `tokenizer.p` pickle file to get the words from their index values.



```
!python3 '/content/drive/MyDrive/Colab Notebooks/testing_caption_generator.py .py' -i '/content/drive/MyDrive/Colab Not
# !python3 '/content/drive/MyDrive/ML/testing_caption_generator.py' -i '/content/drive/MyDrive/ML/Flicker8k_Dataset/373
2021-12-04 18:42:17.597697: E tensorflow/stream_executor/cuda/cuda_driver.cc:271] failed call to cuInit: CUDA_ERROR_NO_
Downloading data from 

start man is climbing rock face end


```



```
!python3 '/content/drive/MyDrive/Colab Notebooks/testing_caption_generator.py .py' -i '/content/drive/MyDrive/Colab
# !python3 '/content/drive/MyDrive/ML/testing_caption_generator.py' -i '/content/drive/MyDrive/ML/Flicker8k_Dataset
```

```
2021-12-04 18:58:54.517921: E tensorflow/stream_executor/cuda/cuda_driver.cc:271] failed call to cuInit: CUDA_ERROR
```

start two dogs are playing with ball end

Conclusion

In this chapter we have thrown some light on the conclusion of our project. We have also underlined the limitation of our methodology. There is a huge possibility in this field, as we have discussed in the future scope section of this chapter.

In this paper, we have reviewed deep learning-based image captioning methods. We have given a taxonomy of image captioning techniques, shown generic block diagram of the major groups and highlighted their pros and cons. We discussed different evaluation metrics and datasets with their strengths and weaknesses. A brief summary of experimental results is also given. We briefly outlined potential research directions in this area. Although deep learning-based image captioning methods have achieved a remarkable progress in recent years, a robust image captioning method that is able to generate high quality captions for nearly all images is yet to be achieved. With the advent of novel deep learning network architectures, automatic image captioning will remain an active research area for some time. We have used Flickr_8k dataset which includes nearly 8000 images, and the corresponding captions are also stored in the text file. Although deep learning -based image captioning methods have achieved a remarkable progress in recent years, a robust image captioning method that is able to generate high quality captions for nearly all images is yet to be achieved. With the advent of novel deep learning network architectures, automatic image captioning will remain an active research area for sometime. The scope of image-captioning is very vast in the future as the users are increasing day by day on social media and most of them would post photos. So this project will help them to a greater extent

Our end-to-end system neural network system is capable of viewing an image and generating a reasonable description in English depending on the words in its dictionary generated on the basis of tokens in the captions of train images. The model has a convolutional neural network encoder and a LSTM decoder that helps in generation of sentences. The purpose of the model is to maximize the likelihood of the sentence given the image.

LIMITATIONS

The neural image caption generator gives a useful framework for learning to map from images to human-level image captions. By training on large numbers of image-caption pairs, the model learns to capture relevant semantic information from visual features. However, with a static image, embedding our caption generator will focus on features of our images useful for image classification and not necessarily features useful for caption generation. To improve the amount of task-relevant information contained in each feature, we can train the image embedding model (the VGG-16 network used to encode features) as a piece of the caption generation model, allowing us to fine-tune the image encoder to better fit the role of generating captions. Also, if we actually look closely at the captions generated, we notice that they are rather mundane and commonplace. Take this possible image-caption pair for instance:

Future Prospects

Future work Image captioning has become an important problem in recent days due to the exponential growth of images in social media and the internet. This report discusses the various research in image retrieval used in the past and it also highlights the various techniques and methodology used in the research. As feature extraction and similarity calculation in images are challenging in this domain, there is a tremendous scope of possible research in the future. Current image retrieval systems use similarity calculation by making use of features such as color, tags, IMAGE RETRIEVAL USING IMAGE CAPTIONING 54 histogram, etc. There cannot be completely accurate results as these methodologies do not depend on the context of the image. Hence, a complete research in image retrieval making use of context of the images such as image captioning will facilitate to solve this problem in the future. This project can be further enhanced in future to improve the identification of classes which has a lower precision by training it with more image captioning datasets. This methodology can also be combined with previous image retrieval methods such as histogram, shapes, etc. and can be checked if the image retrieval results get better.

References

1. Vedantam, Ramakrishna, C. Lawrence Zitnick, and Devi Parikh. "Cider: Consensus-based image description evaluation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
2. Architectures for Image Captioning. 2021 5th International Conference on Computing Methodologies and Communication (ICCMC) _ 10.1109_ICCMC51019.2021.9418234
3. Sharma, H., & Jalal, A. S. (2020). Incorporating external knowledge for image captioning using CNN and LSTM. Modern Physics Letters B 2050315. doi:10.1142/s0217984920503157
4. K. Simonyan and A. Zisserman, in Int. Conf. on Learning Representations (ICLR), 2015.
1. Sci-Hub _ Comparison of Image Encoder
5. S. Hochreiter and J. Schmidhuber, Neural Comput. 9(8) (1997) 1735.
6. Qassim, H., Verma, A., & Feinzimer, D. (2018). Compressed residual-VGG16 CNN model for big data places image recognition. 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC). doi:10.1109/ccwc.2018.8301729
7. Wang, C., Yang, H., Bartz, C., & Meinel, C. (2016). Image Captioning with Deep Bidirectional LSTMs. Proceedings of the 2016
8. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In International Conference on Learning Representations (ICLR).
9. Shuang Bai and Shan An. 2018. A Survey on Automatic Image Caption Generation. Neuro computing .ACM Computing Surveys, Vol. 0, No. 0, Article 0. Acceptance Date: October 2018. 0:30 Hossain et al.
10. Satyanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, Vol. 29. 65–72.
11. Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling

ling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*. 1171– 1179.

12. Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3, Feb, 1137–1155.

13. Adam L Berger, Vincent J Della Pietra, and Stephen A Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational linguistics* 22, 1 (1996), 39–71.