Analysis of Neural Network Models for Visual Captioning System

A DISSERTATION

Submitted in the partial fulfilment
Of the requirements for the award of the degree of

Bachelor & Master of Technology in Information Technology
With Specialization in Robotics

By:

Amol Upreti
(IRM2013010)

Under the Guidance of:
Dr. Vrijendra Singh
Associate Professor
(HOD IT)
IIIT-Allahabad

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, ALLAHABAD

(A UNIVERSITY ESTABLISHED UNDER SEC.3 OF UGC ACT, 1956 VIDE NOTIFICATION NO.F.9-4/99-U.3 DATED 04.08.2000 OF THE GOVT. OF INDIA)

A CENTRE OF EXCELLENCE IN INFORMATION TECHNOLOGY ESTABLISHED BY Ministry of H.R.D., GOVT. OF INDIA

CANDIDATE'S DECLARATION

I do hereby declare that the work presented in this thesis entitled "**Analysis of Neural Network Models for Visual Captioning Systems**", submitted in the partial fulfillment of the degree of Bachelors and Masters of Technology (B.Tech + M.Tech), in Information Technology at Indian Institute of Information Technology, Allahabad, is an authentic record of my original work carried out under the guidance of **Dr. Vrijendra Singh** due acknowledgements have been made in the text of the thesis to all other material used. This thesis work was done in full compliance with the requirements and constraints of the prescribed curriculum.

Place: AMOL UPRETI
Date: IRM2013010

CERTIFICATE FROM SUPERVISOR

I do hereby recommend that the thesis work prepared under my supervision by Amol Upreti titled "**Analysis of Neural Network Models for Visual Captioning Systems**" be accepted in the partial fulfillment of the requirements of the degree of Bachelor and Master of Technology in Information Technology Engineering for Examination.

Date:

Place: Allahabad

Dr. Vrijendra Singh

HOD (I.T.), IIITA

Associate Professor

Countersigned by Dean (A) _____

CERTIFICATE OF APPROVAL

The forgoing thesis is hereby approved as a credible study in the area of Information Technology Engineering and its allied areas carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it is submitted.

Signature & Name of the Committee members _____

On final examination and approval of the thesis _____

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompany the successful completion of any project work would be impossible without the mention of the folks who made it possible and whose constant guidance and encouragement crown all the efforts. This project was not only an endeavor but also an interesting learning experience for me and it bears the imprint of a number of persons who directly or indirectly were a source of help and constant encouragement.

I would like to express my sincere and heartily thanks to my mentor Dr. Vrijendra Singh for his continuous motivation and guidance. His valuable suggestions, comments and support were an immense help for me. I am grateful to him for taking out time from his busy schedule and being very supportive in guiding my work. I am glad to be a part of Robotics Lab, a place that offers excellent environment for research.

Special thanks to my parents & my seniors Miss. Anjali Gautam and Miss. Swarnima Singh Gautam for their efforts & cooperation. The interesting and informative discussions we had together, greatly contributes to the completion of this work.

Place: Allahabad
Date:

Amol Upreti
Dual Degree Final Year, IIITA

ABSTRACT

Recurrent Neural Networks are being exploited for Image captioning task for quite a time now. In this thesis work, the two variations of Recurrent Neural Networks are proposed. First version is almost same as the proposed in most of the literature work present already where an image along with its caption is fed into recurrent neural Network for generation of captions. On the other hand, the second version consists of the generation of text using encoded version of text and images using neural networks. Both the version have been tested in this work and it is found that the second version outperforms the first one which suggests that the image of the Recurrent Neural Networks should be presented as the encoders instead of generation module.

In this work, the ensemble of two types of visual features have been used to encode the better information. Both the version of recurrent neural network are exploited for this ensemble feature representation. On experimenting, it is observed that thing ensemble feature is not producing the good results.

Recent work on attention mechanism have shown that attention model works better in context based caption generation. In this work, we have discussed how attention mechanism works on the feature vector and the visualization of those feature vectors and what information are they storing.

Table of Contents

List of Figures

List of Tables

1. Introduction

a. Overview

Generation of short and concise textual captions on the basis of the given image is a rudimentary problem in the machine learning and it attracts the folks from both Computer Vision and Natural Language Processing. Human mind, just by looking at some scene, can describe the visual of the scenery while computers on the other hand faces problems while doing such judgement and producing human level description. Image to text generation doesn't only include the objects present in the image but it also includes the action being performed in the image, relation between different objects, context and background in which that activity is happening and a lot more and then generation of the sentences on the basis of these attributes in the image. In layman terms, it is the problem translating a given set of pixels to the sentences, conditioned on those pixels.

b. Motivation

In almost all literature, Visual Captioning Systems depends on the two different neural networks, where first image if fed into convolutional neural network for extraction of the visual features and then these features are put as input into the recurrent neural networks which are later on used to generate the sentences or captions. These Captioning systems uses recurrent neural network to input the visual feature and output the sentences which are biased towards the words from the vocabulary of the given captions in such a way their sequence provides the captions pertaining to that image. In short, recurrent neural network are being used for the sentence/ caption generation while convolutional neural networks are being used for the visual feature extractor. However, it the role of recurrent neural network can also be seen as the encoder for the generating sequence rather than for generating sequence itself.

c. Problem Definition

The problem of visual captioning specifies that for given Image, it is required to sample the description of the objects, their actions and relations. In layman terms, for a given image, the corresponding captions needs to be generated. Following two architectures needs to be tested for this purpose.

- Implant Mode

  Here, the image and word are fed into recurrent neural network together. The mode is named as "Implant Mode", i.e. image is implanted into the recurrent neural network.

Figure 1.1: Implant Model

- Merge Mode

  Here, the word embedding is passed into the recurrent neural network and then the encoded sequence together with the encoded visual features are given input to feed forward layer to generate the captions. This mode is named as "Merge Model".

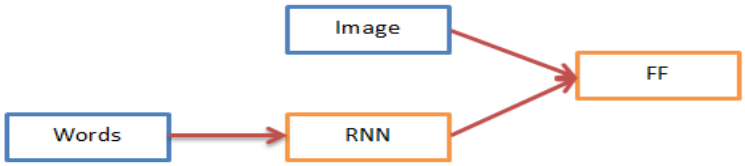


Figure 1.2 Merge Model

d. Applications of Visual captioning Systems

The solution to this problem can be helpful for several application domains. One of the domain is image search on the basis of caption which are first generated with this system and then the reverse searching of these images using the keywords and the activity in the image can be helpful for creating a big Content based Image Retrieval System.
Visually impaired people can use this system for understanding better some sort of visual scene of surroundings.
These systems can also be exploited as the story teller for albums uploaded to social media or image clouds.
Furthermore, application will be exploited with the more research work.

e. Organization of Thesis

The work in this thesis is comprised of seven chapters. The brief description and summary of these chapters are as follows.

Chapter 1. Introduction.

Chapter 2. Literature Survey.

This chapter contains information about all the previous work that has been done in similar research field and it helps building the foundation for the work presented ahead.

Chapter 3. Hardware and Software Development.

This chapter contains the information regarding the dataset used in the research work, tool and software which are exploited for faster computation.

Chapter 4. Methodology.

This chapter contains the information regarding how the proposed work is carried out with all the given resources and modules. A very short discussion on the attention model and feature visualization is performed.

Chapter 5. Testing and Analysis.

This chapter contains the information about all the testing and analysis that we have performed in this thesis work. These experiments include the comparison of the different caption size of the embedding and its effect on the accuracy of the generated captions, combination of visual features and its comparison with existing one and the comparison between the merge and implant architecture.

Chapter 6. Conclusions.

This chapter contains the information about results obtained after the experimenting through the different models of the visual captioning systems.

Chapter 7. Recommendation and Future Work.

This chapter contains the information of the study of what is done and what else can be done on the basis of this research further. It also includes a discussion on the future scope of this work.

2. Literature Survey

a. Approaches of Visual Captioning System

The process of Caption generation from images (Bernadi et al. 2016) is said to be the most important task for the solution of our problem definition which asks for the (Roy, 2005) retrieving of the textual information in conceptual data. The main focus of the most of the Visual captioning system is to generate text based captions on the basis of the elements present in the image. In recent research, folks are working on the (Antol et al. 2015) visual questioning and solution analysis and (Huang et al. 2016) story teller like narration based on the images.

There are mainly three types of Image Captioning techniques (Bernardi et al. 2016) that are discusses in the literature:

- Templated Version: This method using computer vision techniques of object detection and later visual feature extractions from the image for the visual captioning learning models. (Mitchell et al., 2012; Kulkarni et al., 2011) After extractions these visual features are used for the caption generation. The caption generation modules uses the power of Natural Language processing pipeline.

- Retrieval Version: These Visual captioning system considers the task as retrieval problem. In these Visual captioning systems, text or its sections are calculated by analyzing the importance of the text or sub-text in the training set for a picture. This is done by using either (Ordonez et al., 2011) single modal space or a (Socher et al., 2011) multiple modal. These Content based information retrieval techniques are based on the Convolutional neural networks and recurrent neural networks to handle both image and linguistic features.

- Neural Network Version: The last approach is the important approach which is exploited in this work. In this version, Recurrent Neural Network are being exploited for the generation of the highly accurate captions. The visual features are extracted using the convolutional neural network. The extracted visual features are fed into the Long Short Term Memory so that network can be allowed to generate the sub sentence from the vocabulary such the generated description is related to the image.

  This version creates the base of this thesis work and the whole focus was based on this work. The Visual feature extraction is done using the Convolution Neural

Network. However, recurrent neural networks are being used as the text generation machine, while in this work we are experimenting the use of recurrent neural network as the encoder for the textual sequence. Also, we are giving visual feature along with the linguistic feature as the input to the Long Short Term Memory.

b. Related Work

In the domains of computer vision as well as Natural language processing, automated captioning has brought a lot of attention which connects the machine learning. The task related to this domain may seem cumbersome for some naïve folks in the field but a lot of research has been done in Machine learning especially Neural Networks and its subdomain using the power of both natural language processing and the computer vision clubbing with state of the art Machine Learning Techniques. The state of the art machines like Gated Recurrent Unit, Long Short Term Memories and Convolution Neural networks are being exploited by the researchers now a days. Moreover, the big datasets from the famous competitions like ImageNet, Microsoft COCO etc. are exploited for the knowledge discovery for such systems.

The Involvement of Neural Machines classifies the process into two major models that the researchers have exploited in their work.

- Pipeline Model
  In this model, two separate models work separately, the linguistic model and the visual feature extractor. The visual feature extractor first extract the features and then the linguistic model do the work of generating sentences.

- End-to-End Architecture
  Only single neural network is responsible for the visual feature extraction and the sentence generation. So this basically it is an encapsulation of the both of the models.

A descriptive literature survey for image captioning is done the research papers related to this domain. This has been encapsulated below with each research papers and the methods that have been used those.

- [6] Show and Tell: A Neural Image Caption Generator
  The architecture discussed in this paper is more close to end to end pipeline architecture and it is more intuitive. The maximizing term here is the log likelihood of the probability distribution of the sequence of the linguistic text conditioned on the picture as written formally in following equation which calculates the cost function of the given problem.

$$C^* = arg\ max_C \sum logp(S|I;C)$$

The work in this direction started by taking motivation from the recent research in the Natural language processing area. In NLP, machine translation in particular, uses recurrent neural networks such as Long Short Term Memory for the encoding and decoding of linguistic features. In encoding part, the input is replaced by visual features. This model is called as Neural Image caption. Beam Search is widely used for testing the accuracy of the generated captions. BLEU score was exploited for comparison and evaluation and it helped in tuning the network's hyperparameters.

- [5]Deep Visual-Semantic Alignments for Generating Image Descriptions
  Generating a sentences from the images is quite a task but which object or action in the image is related to which part in the sentence is tough job to figure. [5] Karpathy et al. modeled a machine to learn these relation between image's part and sub-sentences. It was proposed in this work that the visual feature should be aligned with linguistic features. This is done by embedding both the features in common vector space. They proposed a multimodal LSTM which extract linguistic features from the corresponding images. These linguistic features are then trained on the inferenced data and testing is performed in local based semantics.

  To find the proper relation between the generated captions and the objects or actions in the image, a rank based formula is employed which generated a tree based dependencies between the captions and regions. Bidirectional recurrent neural networks are used to generate novel captions as these machines are known to handle overfitting and context based memories. The work of this image to capture the regions which minimizes the similarities between the linguistic and visual feature representations.

- [22] DenseCap: Fully Convolutional Localization Networks for Dense Captioning
  [22]Kaparthy et al. 2015 discovered this method of dense captioning where for the detected image region, the set of text is generated. Hence, this task is similar to object detection as here the model is mapping the objects in the image to some set of captions.

  Main part of this research is that to find the important regions in the image and for extractions of the weights of those regions a dense localization layer is employed using bilinear interpolation. To extract the visual features from the image, weights of VGG-16 is used by [22]Kaparthy et al. in this work. The features are extracted from the second last layer which is of 512 units. These features are passed through the recurrent neural network.

- [23] Image Captioning with Deep Bidirectional LSTM

Cheng et al. 2016, in their work, proposed to different models. The proposed model learned hierarchical visual linguistic features whose performance was comparable to the state of the art methods available. To overcome the overfitting problem due to the very deep nature of the model, some methods such as multi cropping, multi scaling and horizontal mirroring were used for data augmentation tasks.

To learn the visual features, the weights of famous computer vision architectures like Inception and VGG-19 were used. Bidirectional LSTM is used to learn word embedding without losing the context information. Cognitve mind was the main inspiration for using bidirectional LSTMs. As mentioned in the paper that LSTM learns horizontal features that it learns those features and share the weights in horizontal direction that means they are being used many times before changing. This makes the learning of embedding slow and hence the proposed method of fast up the things is to put a Multi-layer perceptron (Neural network) in between the horizontally stacked LSTMs. The best caption is selected from the comparison from both front and the last generation.

- Show, Attend and Tell: Neural Image Caption Generation with Visual Attention[7]

  In this work [7] Kelvin et al. proposed a better method of generating the captions from the images. They have used the attention mechanism. In earlier works, on dimensional vector of visual feature was not enough to judge whether the next word is dependent on one particular part or some set of regions in the image. Attention mechanism allow the caption generation to be dependent on the different activation of the image and hence it can learn which word is related best to which activation of the image. In this work, instead of using features from second layer of the Convolutional layer, low lever feature at the high end are used which have the actual activations of the image. There are two types of attention mechanism proposed in this work: soft attention and hard attention.

  Low level features at the high end were collected from famous deep convolutional neural networks like VGG-19 or InceptionV3. These features are the activation features and each feature can be seen as the attention to some special part in the image which other feature doesn't. So each feature part if mutually exclusive to other in terms of the objects or actions. This attention model works better that all other models discussed above.

From the whole literature survey of these pre-existing research there it can concluded that for Visual captioning systems, Long Short Term Memory can be used as both implant and merge modules. Also, it can be concluded that the Long Short Term memories, on one hand word as Caption generation machine conditioned on the corresponding visual features, on the other hand, they work as the encoder for the embedding the sequences along with the visual features to generate the captions with the feed forward neural network.

3. Hardware and Software Development

Dataset

We have used Flickr8k and Flickr30k dataset for Images and Captions. Flickr8k dataset includes images/pictures from Flickr website. There are around 8091 images in the dataset and for each image there are 5 annotations/captions.Flickr30k dataset is the extension of the Flickr8k which has around 31783 images and here also for each image, we have corresponding 5 captions/annotations. There are around 158k captions.

The folder structure of Flickr 8k dataset is as follows:

```
Flickr8K/
    Flickr8K_dataset/
    Flickr8K_text/
        Flickr8K.token.txt
        Flickr8K.lemma.txt
        Flickr8K.trainImages.txt
        Flickr8K.devImages.txt
        Flickr8K.testImages.txt
        ExpertAnnotations.txt
        CrowdFlowerAnnotations.txt
```

- Flickr8K_dataset/

  This directory contains all the pictures collected from the website. There are around 8091 images in this folder.

- Flickr8k.lemma.txt

  This file contains lemmatized captions for users to apply their learning model on this version.

- Flickr8k.token.txt

  All the captions corresponding to the images in the folder above are written in this file. The format of the caption is:

  Image id + '#' +[Caption No] + caption

  Example: "103718903471bn1y91b.jpg#2 A white dog is running through the snow."

- Flickr8k.trainImages.txt

Flickr8k has information about the partition of the dataset into the test, train and validation sets for the development purposes. This folder has around 6000 train images.

- Flickr8k.devImages.txt

  This file contains the validation or dev set from the above partition. It has around 1000 images info.

- Flickr_8k.testImages.txt

  This file contains the test set from the above partition. It has around 1000 images info.

- ExpertAnnotations.txt

  Experts from all over the world have given their judgements on these images and it has been included in this file. Each of the expert has given their rating on each caption from 1 to 5. 1 being the best rating which means that the given caption best describes the sentence and 5 being the worst.

- CrowdFlowerAnnotations.txt

  CrowdFlower is as ML and AI Company which exploits human intelligence to perform simple basic tasks such as provide explanation of images and writing texts. This file contains judgement of each image-caption pair as three columns, first being the percentage of truth, second being the sun of truth and third being the sun of false. If the final verdict for particular caption is truth, it is said to be acceptable for that particular image otherwise it is neglected.

SENTENCES

1. A white dog is running through the snow .
2. A dog running through deep snow pack .
3. A dog is playing in the deep snow .
4. A dog runs through the deep snow .
5. White dog running through snow

1. A group of Asian students pose for a picture with a Star Wars character in the center .
2. A large group of Asian people posing for a picture with a storm trooper
3. Asian students wearing uniforms collect around a Star Wars icon .
4. A big group of students surrounding a storm trooper .
5. Asian class posing for picture with Stormtrooper

1. a musician plays a strange pipe instrument whilst standing next to a drummer on a stage .
2. A man blows into a tube while standing in front of a man at the drumset on stage .
3. A man blows into an electrical instrument by a microphone .
4. A man plays an instrument next to a drummer .
5. Two men perform a song together on stage .

Figure 3.1: Sample images and captions from Flickr8k and Flickr30k dataset[24]

Due to the possibility that each image can have a different meaning for different users which results in high bias, more than one captions are used per image.

We have also used pre-trained weights of VGG19, AlexNet, and InceptionV3.

Software and Hardware Requirements

We have used some Machine learning libraries to speed up the work flow. Along with the libraries, hardware requirements are also mentioned below.

- GPU

  A GPU (graphical processing unit) is specialized electronic unit which is used to faster manipulate and change memory to accelerate the formation of images in the frame buffer mainly for output to display. It performs very high computation tasks on vector multiplication and convolutional operation and these operations are quite common when it comes to any ML model. We have used NVIDIA Quadro-K6000 which is equipped with 16 cores and 257838MB device memory.

- Tensorflow

  Google Brain team built tensorflow library for internal google use. It was released on November 9th, 2015 under Apache 2.0 open source license. DeepDream, an automated caption generation software made by google was made on Tensorflow.

- Keras

  It is a high level python API which has basic unit as Neural Nets and it runs on the top of ML frameworks such as Tensorflow, Theano, NLTK, and Microsoft Cognitive Toolkit. It allows the training and testing using Graphical Processing unit.

- Anaconda

  Anaconda is a tool built upon the Python and R for Data Science, Predictive Analysis, Artificial Intelligence, Scientific Computing, and Machine Learning work. It deploys different packages using the utility named as conda.

- NLTK

  Natural Language Tool Kit (NLTK) library is exploited in this work for handling word embedding and BLEU score. It is also used here for preprocessing the caption before feeding into the neural network.

4. Methodology

4.1 Introduction

a. Word Embedding Model

"Apple launches a new tablet, how do we know it is a company not fruit". Word embedding are the vector form of the word. The way these vectors are represented may be different.

All the machine learning models/ algorithms are known to face problem when given the raw string directly as input. These algorithms are required to perform number of task on vectors only. So to provide the vector as input, we use knowledge extraction from string data.

The main task comes as the solution to this problem is that it we need some specific method for converting the text into the vector. We use the dictionary which maps the word of character to a number according to the vocabulary. This number then converted to one-hot encoded values which then feed into the input as a word embedding. Mikolov et. al. introduced the standard version and named it word2vec.

Word2vec is a vector format of a word which contains the probabilities of each of word in the vocabulary.

b. Transfer Learning

In this approach to deep learning, we use pre-trained models as the initial point for computer vision and NLP tasks. This approach is often used when we want to abstract the knowledge from one domain and use it in some similar domain. It is also used when the dataset we are using is very small in size and hence any big network will over fit that dataset. When there is no option of increasing the size of the dataset, we often perform transfer learning.

This is mainly performed in three steps:

Source Model Selection: A pre-trained model weights are chosen (for example VGG16 in case of some computer vision task) and then loaded to the model itself.

Reusing the Model: This new model can be used as an initial point for the new task that we are interested in. It may use whole pre-trained network or it may be changed by removing some layers, depending upon the need.

Fine Tuning the Model: This task includes the retraining of the model on the new task, if necessary. This step can be skipped and directly model can be used for testing too.

Following image sumps up Transfer Learning. As we can see that the use case of transfer learning comes into play only when two factors meet the requirements are Data Similarity and Size of the Data.

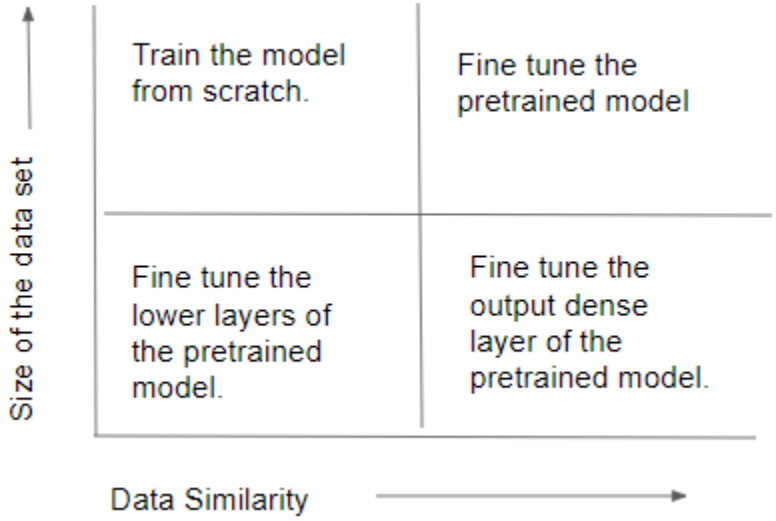


Figure 3.2 Size of Data vs Similarity Operation

c. Convolution Neural Network

It is basically a Neural Network which works on the one function "Convolution". They have weights and biases which be learnt using same backpropagation algorithm.
The Normal Neural Network, when gets bigger and deeper, doesn't scale properly, that means the number of parameters starts increasing exponentially with the size of the image and number of hidden layers. While in case of CNN, due the property of the shared weights across the image, the number of the features decrease across layers and hence even deeper network scales well.

There are mainly following types of layers in any CNN:

- Input Layer: This layer takes the input image, it can be RGB or grayscale image.

- CNN layer: It is same as convolutional layer as specified as below. Only filter in this layer is the learning parameter. In the convolution Operation:

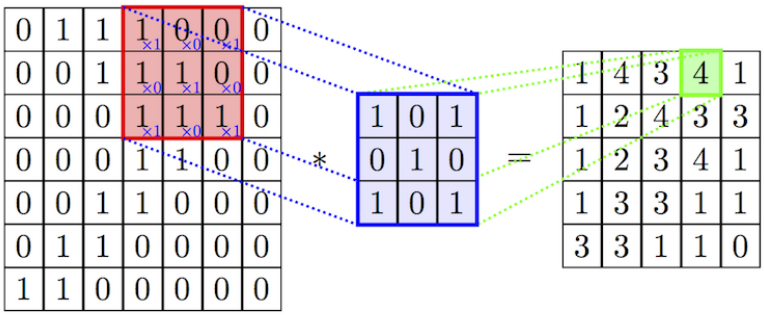


Figure 3.3 Convolution Operation

- RELU layer: This layer is used for applying the activation on the output image after convolution function.

- Pooling Layer: It is used for down sampling of the input or intermediate features.

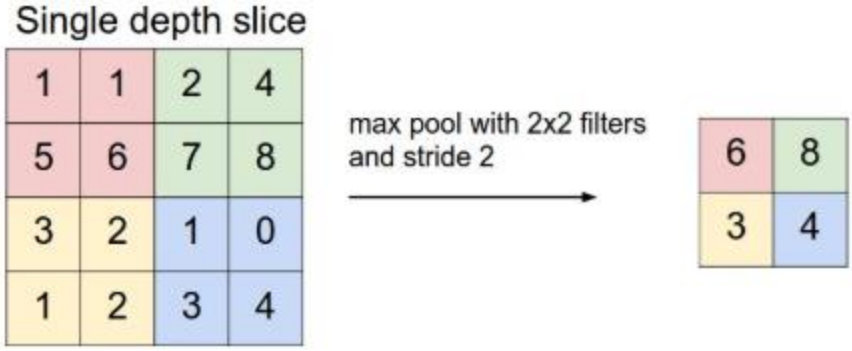


Figure 3.4 Max Pooling Operation

- Fully Connected Layer: It is same as the normal neural network layer which is mainly at the last of any CNN. Their main task is to convert the features into class scores.

d. Recurrent Neural Networks

With the other versions of neural network we have discussed above, there was the common problem i.e., all of them take inputs of constant length and output the constant length output so we cannot use them for the variable length input/output.

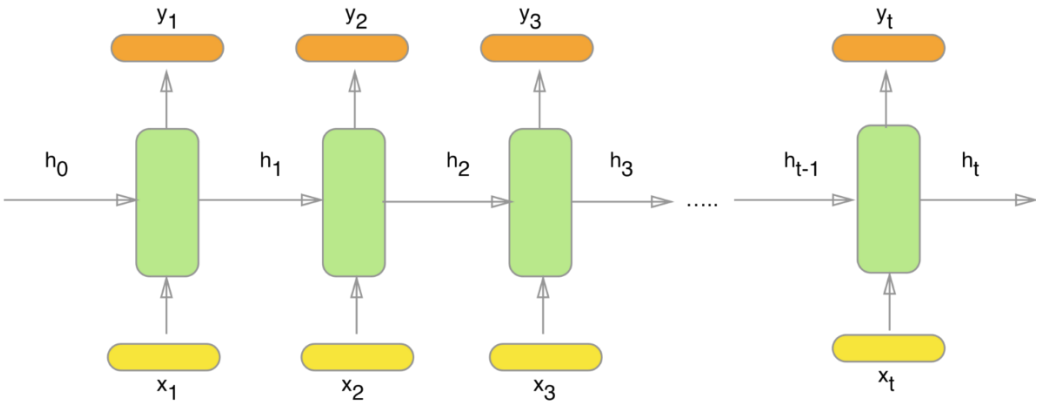


Figure 3.5 Recurrent Neural Network Module

So, to overcome that problem, we use RNN which takes input as input example and the output from the last input. The learning of this network is performed through time. For t instances of the network, we have around t by number of parameters for single network.

There are following types of RNN:

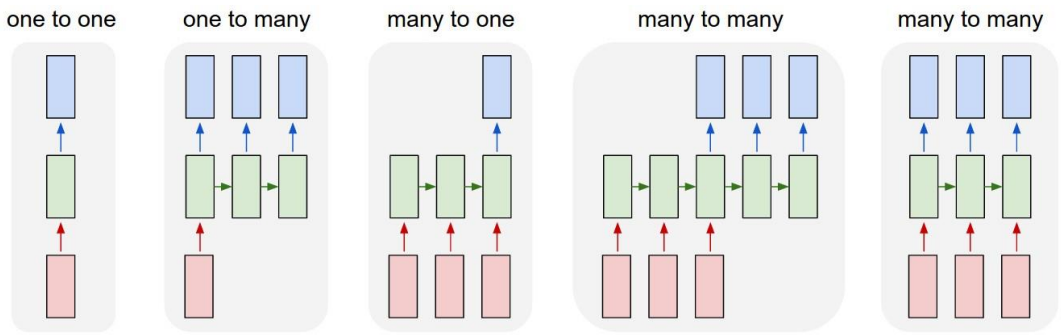


Figure 3.6: Types of Recurrent Neural Networks

e. LSTM

Following diagram explains the basic working of LSTM. We switch to LSTM/GRU because of the problem of Vanishing gradients. When the network becomes large, it nearly impossible for the gradient to travel from last layer to first without vanishing. This makes network learn slower and at the same time the performance declined. So we cannot use Simple RNN for learning large features.

The code of the LSTM is the cell state which is a horizontal chain type line on the top of each cell. It changes its value on interaction with the different gates in the path. We are using following annotations for the equations:

C_t = Memory cell

C_t' = Candidate for replacing the memory cell

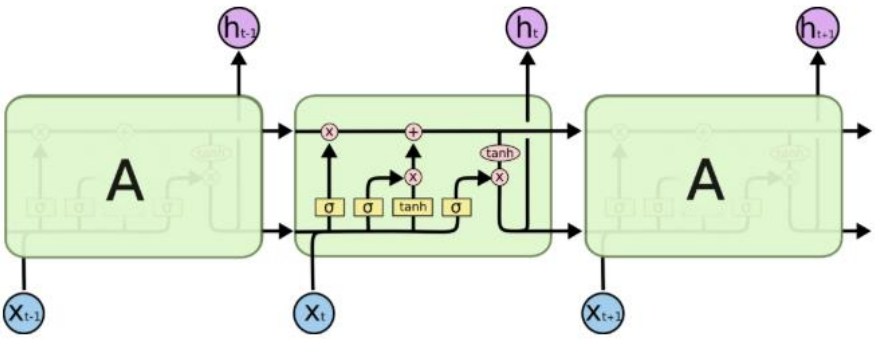
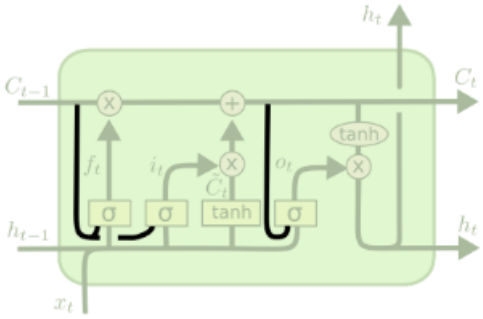


Figure 3.7: LSTM Module

There are four equations which governs the working of a basic LSTM repeating unit:



$$f_t = \sigma\left(W_f \cdot [C_{t-1}, h_{t-1}, x_t] + b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [C_{t-1}, h_{t-1}, x_t] + b_i\right)$$
$$o_t = \sigma\left(W_o \cdot [C_t, h_{t-1}, x_t] + b_o\right)$$

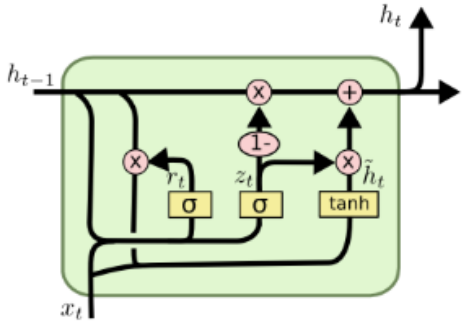
Figure 3.8 Gated Mechanism in Long Short Term Memory

f_t = Forget gate , i_t = update gate , o_t = output gate

$C_t = i_t * C_t' + f_t * C_{t-1}$

f. GRU

This is another type of RNN which solves the problem of Vanishing Gradient with slightly different mechanism. In this model, we merge the cell state and hidden stage and the resulting model is simpler as well as more robust.



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Figure 3.8: Gated Recurrent Unit

g. Attention Mechanism

Whenever we are using any sequence model, we often train the model considering one thing in mind that is each new word/output in the prediction will be the depending on only last inputs, it will not take future context in the prediction. So, the output can be absurd sometimes when such things happens.

Also, for longer sentences, it is very hard for neural network to memorize it all and then perform translation.

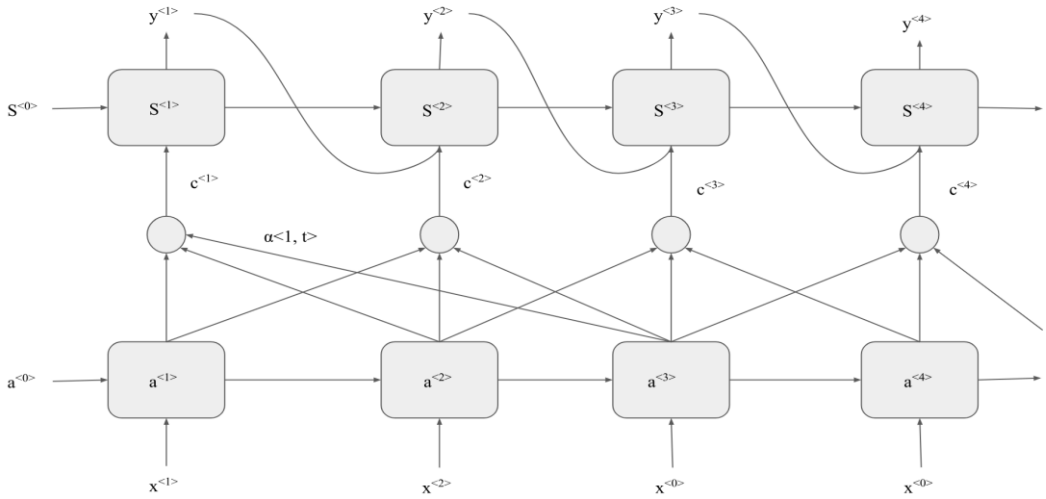


Figure 3.9: Attention Mechanism

Where,

$a^{<t'>}$ = *activations for attention layer*

$\alpha<t, t'>$ = *weights of attention layer (amount of "attention"* $y^{<t>}$ *should pay to* $a^{<t'>}$*)*

= $exp(e^{<t, t'>})/(\sum exp(e^{<t, t'>}))$, *where* $e^{<t, t'>}$ *is trained output of a neural net on* $s^{<t-1>}$ *and* $a^{<t'>}$.

$c^{<t>}$ = *context* = $\sum \alpha < t, t' > * a < t' >$

t' = *time step for input sentence*

t = *time step for output sentence*

$y^{<t>}$ = *output word at time t*

$x^{<t'>}$ = *input word at time t'*

h. BLUE-N score

When different sentences generated are true for single input, traditional beam search will not be able to decide which one to select. To overcome this problem of multi-correctness, we use BLEU (Bilingual evaluation understudy) score.

BLEU score is an evaluation method to test our generated output with the original/reference output. BLUE score value ranges from 0 to 1 (0 being the worst and 1 represents exactly same sentence as reference)

Suppose we have following two sentences:

- The bat is on the surface.
- There is a bat on the surface.

Model Output: The bat the bat on the surface.

Possible Bigrams	Count (for Model)	Count Clip (in reference)
The bat	2	1
bat the	1	0
Bat on	1	0
On the	1	1
The surface	1	1

Table 4.1: BLEU score examples

Precision = Count Clip/Count = 4/6
This is how we find BLUE score on bigrams. The NLTK library provides a package for BLEU score. Apart from Machine Translation BLEU.

4.2 Process Flow of Visual Captioning System

The following process flow is the summarize representation of the work done. A detailed discussion of each component is shown in this flow chart.

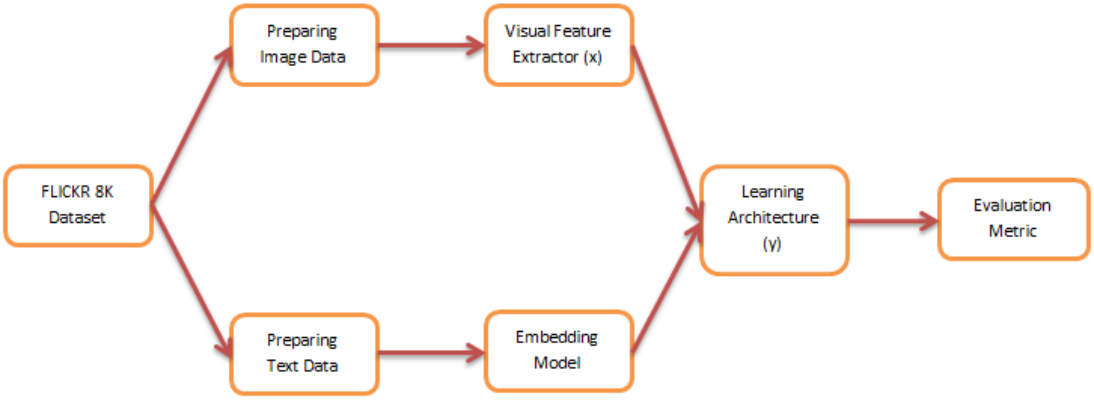


Figure 4.1: Process Flow

4.3 Preprocess Image Data

The given datasets Flickr8k and Flickr30k are collection of total 8000 and 30000 images respectively. These images are RGB images with 3 dimensions: width, height and channel. To give any image as an input to the model, it needs to be preprocessed. The preprocessing includes, edge detection, object segmentation, background extraction, etc. The way this can be done manually but it takes a lot of labor to perform the task. To ease this task, auto-encoder is exploited. What auto-encoder does is that it takes the image as input and outputs the extracted feature of that image. VGG19 and InceptionV3 without the last layer have been used to perform this feature extraction.

First the given input image is resized to [224,224] with all other channels as it is. So we will input the image of size- [224, 224, 3] to VGG-19 or InceptionV3.

Secondly, feature wise zero center is performed on each sample for a particular fixed mean. This fixed mean is taken from standard Image Net competition where it was calculated as sum of all the RGB values dividing by total width*height.

Later this image is then fed into the auto-encoder which is discussed in next section.

4.4 Extracting the Visual Feature

To extract the visual feature we have exploited 3 different Convolutional Neural Networks. These are VGG, Inception, and AlexNet. The brief discussion on these are as follows:

1. VGG
   1.1. Introduction

   This architecture is designed and trained by Oxford's famous Visual Geometry Group. It is very deep CNN with over 16 and 19 layers in VGG16 and VGG19 respectively. It was proposed by Karen et al. in their paper "*Very Deep Convolutional Networks for Large-Scale Image Recognition*". The main purpose of designing this network is to detect object in given image. It takes the image of size [224,224,3] as input and then performs the object detection. The size of weights of this model is around 528MB and this network gives 92.8% accuracy on Image Net competition which has about 1000 classes and 14 million images.

   1.2. Architecture

   There are several architectures proposed by Visual Geometry Group and they have been explained in following figure and only D and E have been used for feature extraction.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 4.4.1: Different VGG architectures

In above figure 4.4.1, "Conv3-256" is abbreviation of "256 convolutional filters of size 3x3 each", "FC-1000" is abbreviation of "Fully connected Feed forward layer of 1000 neurons", "soft-max" indicates the "Softmax layer for performing the activation" and "maxpool" points to "Max-Pooling layer".

The precise model of VGG-16 is as described in the following figure 4.4.2. The one things that should be taken care of in the following figure is that a preprocessing layer is included in the architecture to perform zero centering feature wise on each image.

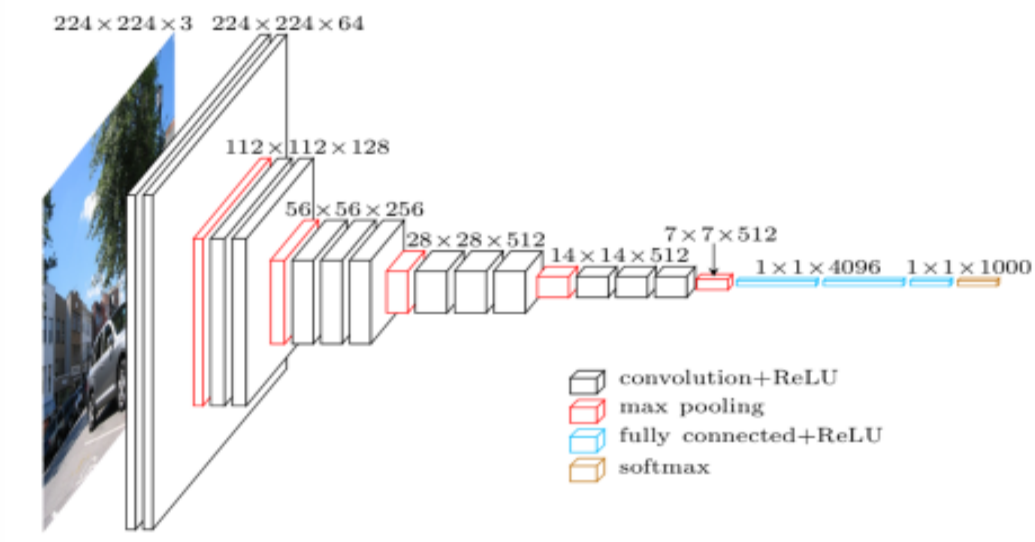


Figure 4.4.2 3D representation of VGG-19 layer network.

1.3. Application

Here in our work, VGG-19 is used as one of the visual feature extractor. The preprocessed Image section 4.3 is input to the network. The last softmax layer has been removed and now the last layer is only fully connected fed forward neural layer with 4096 neurons. The output is the 4096 sized vector.

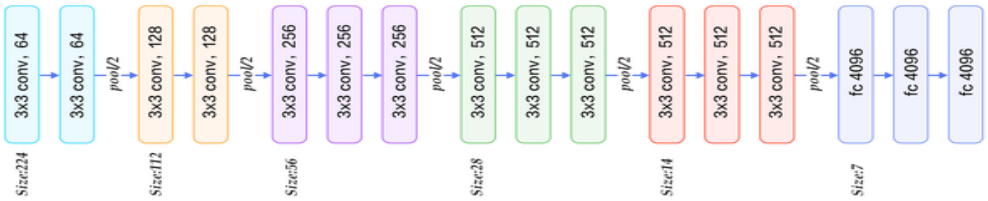


Figure 4.4.3: VGG-19 without soft-max layer.

2. InceptionV3

2.1. Introduction

Christian et. al. 2015 proposed the first inceptionV3 architecture in their paper "*Rethinking the Inception Architecture for Computer Vision*". This model was benchmarked with top-5 error rate of 3.6% on testing set at ILSVRC 2012.

2.2. Architecture

The figure 4.4.5 shows the basic inception module. This is building block of several convolutional inception network. It can be seen in the figure that there are 1x1, 3x3 and 5x5 convolutions along with a 3x3 max-pool layer. To reduce the computation cost of such a large network, the reference paper suggested that to use 1x1 convolutions.

It has been observed that when multiple features from multiple filters has been exploited, the performance of the network improves. One more point to notice here is that earlier convolutional networks don't use the 1x1 filter, which on used in inception module helps in cross channel correlation along with the spatial dimension correlation by bigger 3x3 and 5x5 filters.

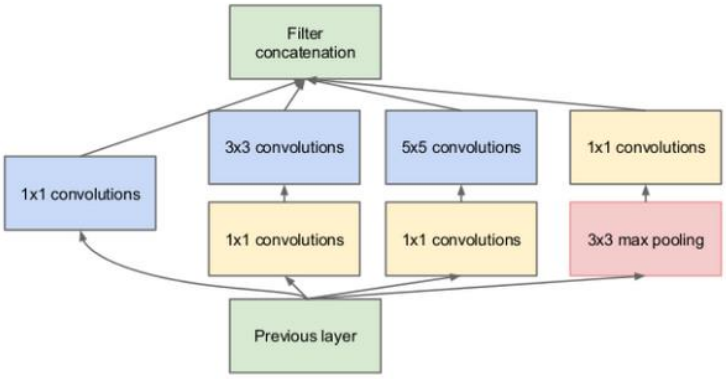


Figure 4.4.5: Basic Inception Module

2.3. Application

In our work, InceptionV3 is used as another visual feature extractor just like VGG-19. Again the image of the shape [224, 224, 3] is given as input in this inception module. So the preprocessed image of shape [224, 224, 3] was put into the Inception network and the resulting vector as the output of sixe 2048 and applying global average pooling on that, as shown in figure 4.4.6.

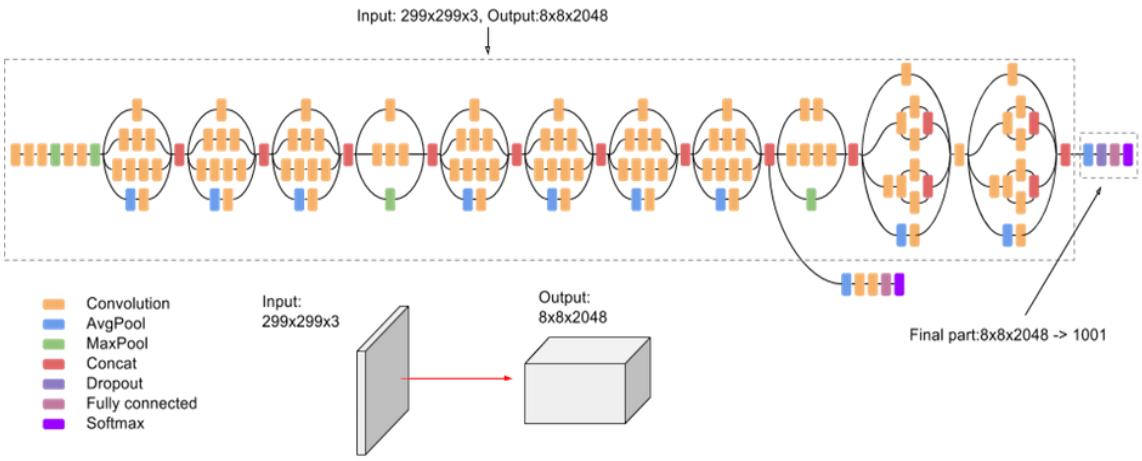


Figure 4.4.6: InceptionV3 Architecture

4.5 Preprocessing the Text Data

Textual data are preprocessed first before inputting into the embedding model. Following steps are taken for this preprocessing task.

- Punctuation removal
- Words less than certain frequencies are removed
- All text to lower case
- Words with numbers like age, address, street no., telephone no etc are removed.
- Stop word have important role in generating fluent captions so they are not removed.

This preprocessed text needs to passed from LSTM/GRU/RNN via embedding model but before that we need to perform one more operation to whole sequence that we have till now. A starting and an ending tag needs to be added to the sequence to specify the starting and the ending of the caption.

For Example:

- Original text: A woman is riding on a "British Horse"
- Preprocessed Text: a woman is riding on a horse.
- After Adding tags: [startseq, a, woman, is , riding, on , a, horse, endseq]

Now at the time of inputting this whole sequence into the embedding model, we will have startseq as the first word. But embedding model need fixed size input. Therefore, we add padding to the input sequence to each of the word such that maximum length of the padding is equal to the sum of startseq, endseq and caption. Each time step of how input and output should be there for embedding model is shown in the figure 4.5.1.

INPUT	OUTPUT
startseq [pad maxlen-1 times]	a
startseq a [pad maxlen-2 times]	man
startseq a man [pad maxlen-3 times]	is
startseq a man is [pad maxlen-4 times]	riding
startseq a man is riding [pad maxlen-5 times]	on
startseq a man is riding on [pad maxlen-6 times]	a
startseq a man is riding on a [pad maxlen-7 times]	horse
startseq a man is riding on a horse [pad maxlen-8 times]	endseq

Figure 4.5.1: Example of Embedding model Input and Output.

4.6 Embedding Model

A dense vector that represents a word is called as word embedding. Differences and similarities between the words can be obtained using word embedding. In this thesis work, the embedding model is trained on whole training set. We have used 3 word vector length [64,128 and 256]. So for a caption of length 34 which is actual maximum length of the caption, then on passing through the embedding model, the output is generated matrix will have size of 34*64.

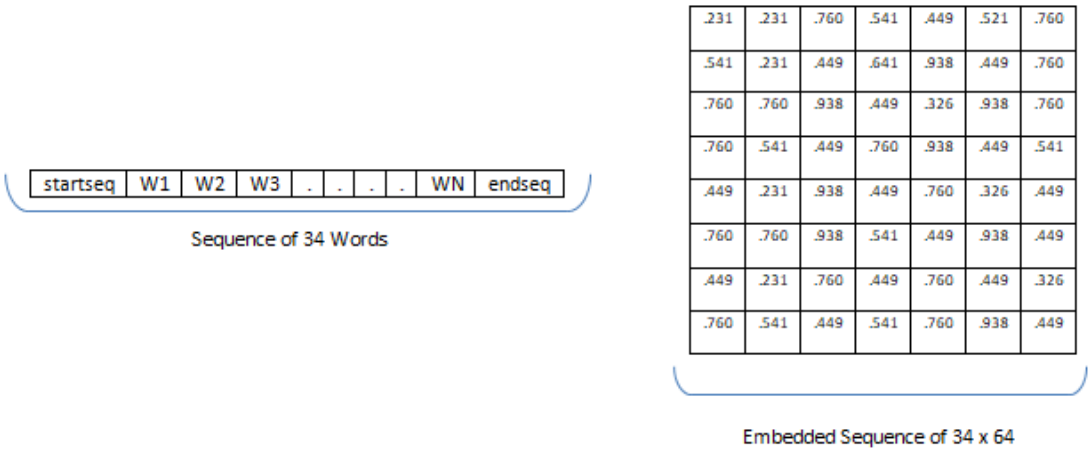


Figure 4.6.1: Embedded vector word example

4.7 Learning Models

1. Encoder Decoder Model
   In this thesis work, captioning system follows the standard model architecture also known as encoder and decoder architecture. It has two components, encoder and decoder. Encoder, is a CNN network which takes the input and encode it into feature which is fixed length vector type representation. Decoder, on the other hand, is a network module which takes that encoded feature and generate the resultant results as outputs.

2. Implant Model
   In this model, the visual feature vector which is extracted from VGG-19/ InceptionV3/ AlexNet activation layers which contains the visual information of the images in section 4.3 are implanted into the two different RNNs namely LSTM and GRU along with the

textual embedding. Here we have treated the image as the prefix of the caption which is joined before startseq.

This model train LSTM and GRU to condition the generation of both visual and linguistic features simultaneously. In brief, the LSTM and GRU are responsible for generation of the caption with visual information as the base condition for generation of those caption. The architecture that has been used for this model is shown in the figure 4.7.1.

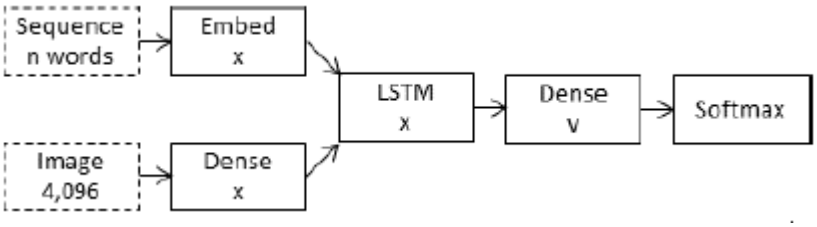


Figure 4.7.1: Implant model for captioning system.

3. Merging Model

In this merge model, LSTM and GRU only have embedded text from embedding model as input and visual features are kept away so that they can be merged at a later in the CNN. This way caption prefix is only text and no visual feature.

Merge model allows LSTM and GRU to only encode linguistic information without caring about the visual feature which are going to merge into later stage after RNN. Only at the later stage Visual feature works as the condition for the generation of the captions.

There are several ways of merging these features which are described in the literature. The basic version that we have exploited in this work is shown in the figure 4.7.2.

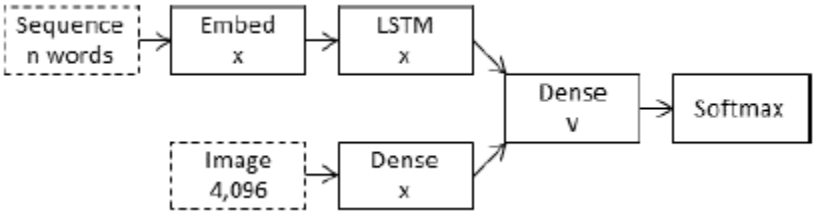


Figure 4.7.2: Merge model for captioning system

4. Ensemble Model

This work also proposes the ensemble model for the two visual features extractors which are used to encode the images. This experiment is performed to see whether the visual extractors from two different visual extractors can contribute to better accuracy for

caption generation. Here we have used the dense layered features and then merging of those features. Both merge and implant models have been tested for this ensemble to test the consistency of GRU and LSTM.

5.

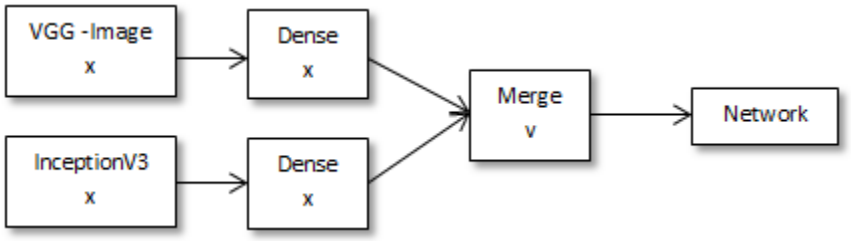


Figure 4.7.3: Ensemble model for captioning system

6. Attention Model

Normal learning model for image captioning system have been discussed in earlier sections of this work. It is shown in the figure 4.7.4. In this model, an input image is encoded into a feature (named as h here). This encoded feature, h, is then passed to LSTM or GRU to generate desired caption.

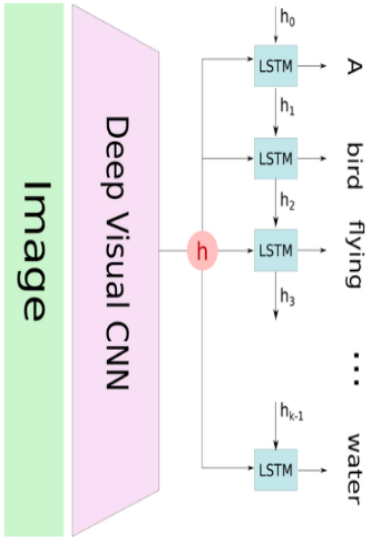


Figure 4.7.4: Image captioning using normal Captioning System

The problem with this model is that it tries to learn the caption/ text based on the feature h of the image. When the model is predicting the next word for the caption, it needs to be concerned with only with some portion of the image not whole image. Therefore model

cannot produced proper captions which are required. So, attention mechanism is used for this purpose. We have already discusses about the attention mechanism in section 4.1 (g).

Image is divided into N parts for applying the attention mechanism. Then using convolutional neural network we learn the feature representation as $[y_1, y_2,\ldots,y_N]$. Now, with this mechanism recurrent neural network can put its attention on one part of the image which is relevant and hence the quality of caption increases.

The figure 4.7.5 shows the exact working of the attention mechanism. The hidden state of RNN will be h_i if I words have been predicted. The Z_i , output of the attention model is the representation of the filtered image where the main scenes of the concerned image are important and it is used as the input for LSTM for predicting the new hidden state h_{i+1} and new word.

The visual features which have been used to exploit the power of attention mechanism have been discusses in the next section.

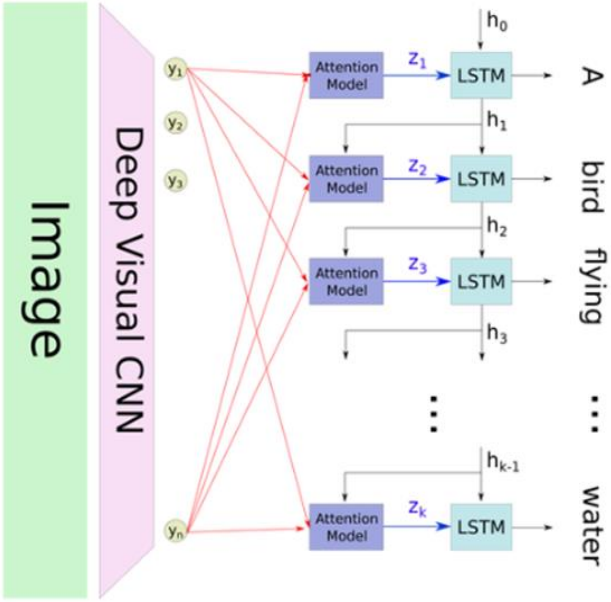


Figure 4.7.5: Attention mechanism for captioning system

4.8 Visual Features Visualizations

To visualize what type of features are used by attention mechanism, we have extracted 512x14x14 are extracted from VGG-19. The visualization is shown in the figure 4.8.1. It is to be noted that only few features have been visualized out of all 512 feature maps of size 14x14.

These selected feature maps are contains some special information regarding the image like background, foreground, person, tree, path etc. These information are helpful for the attention model to judge that which feature to exploit during training of the captioning system.

Figure 4.8.1 (a): Visualization of feature map

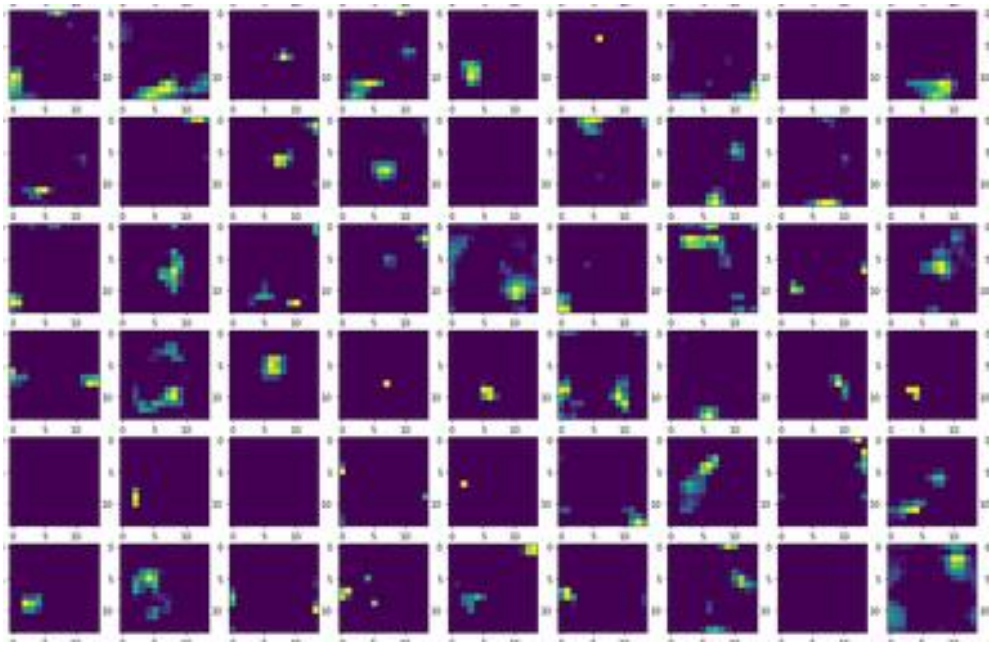


Figure 4.8.1 (b): Visualization of feature map

5. Testing and Analysis

a. Comparison on Merge and Implant Architecture

Architecture of both views that is used to perform the experiments is very basic. There is no hyper-parameter tuning involved and the techniques such as regularization are not considered. This is done to avoid the bias from both architectures so that one finely tune architecture cannot overwhelm another. For testing purposes, these architectures are run 3 times and then mean of the score is noted.

Implant architecture is shown in figure 5.1. First two boxes at the top are input layers. One is sequence that we get from text pre-processing and another visual representation of image extracted from convolutional networks. The sequence goes to embedding model to dense layer and then a dropout is applied. Visual feature vector goes to dense layer to dropout. Then both the encodings of language and image are combined and implanted into the LSTM. After that a dense and then a soft-max layer is introduced. Dropout rate is fifty percentage fixed.

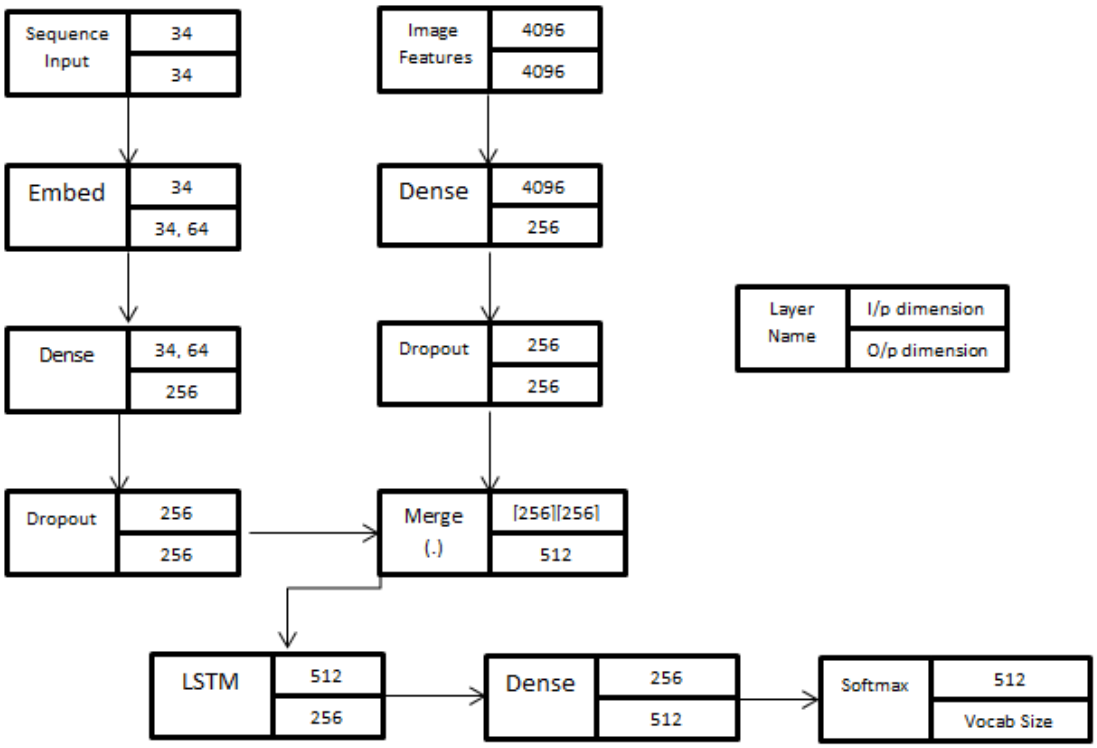


Figure 5.1: Implant Architecture

Merge architecture is shown in figure 5.2. Every box represents a neural layer. First two boxes at the top are input layers. One is sequence that we get from text pre-processing and another visual representation of image extracted from convolutional networks. The sequence goes to embed to dropout to LSTM. Image features goes to dense to dropout. After that both extracted features are merged together on concatenation. Then concatenated vector goes to dense layer and then to softmax layer. Softmax layer is a fully connected layer with softmax activation. Total number of neurons in softmax layer is equal to size of trained vocabulary. This layer will predict the most probable word.

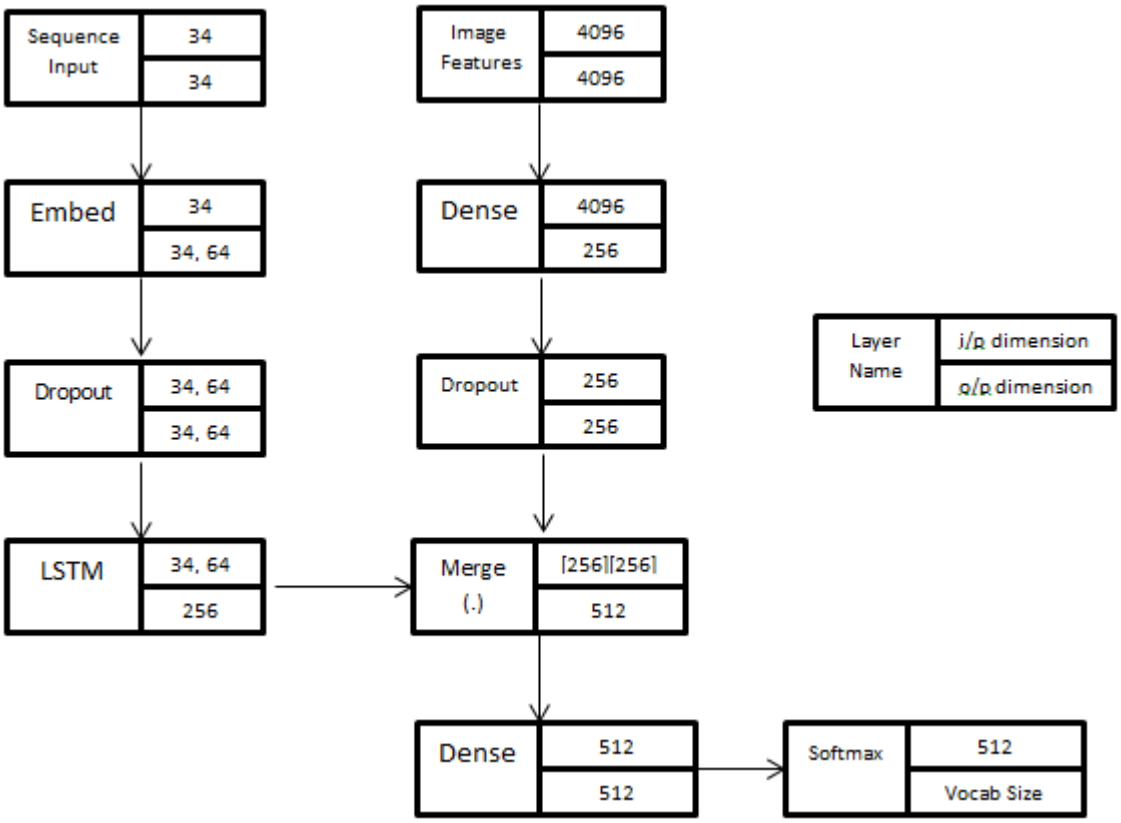


Figure 5.2: Merge Architecture

On both architectures, data is train and tested. Parameters on which these experiments are performed are embedded word vector length and the size of LSTM state. The evaluation metrics score and their analysis is performed in *"Results and Analysis"* section.

b. Comparison on Caption Numbers

To understand the working of Long Short Term Memory Networks, an experiment is also performed where constraints are put on number of captions. By using same implant and merge architectures mentioned above, this experiment is performed. In this experiment, number of captions that are provided to the network is in the range of one, three and five.

c. Ensemble Models

This is the last experiment performed in this work. The ensemble model based on use of two visual feature extractors or convolutional neural networks is exploited. Two convolutional neural networks are VGG-16 and InceptionV3. The architecture of this model is shown in figure 5.3 and figure 5.4. It can be seen that feature vectors extracted from both models are first passed through dense layer separately and then they are merged using concatenation both. Rest architecture is the same as it is for previous implant and merge architectures.

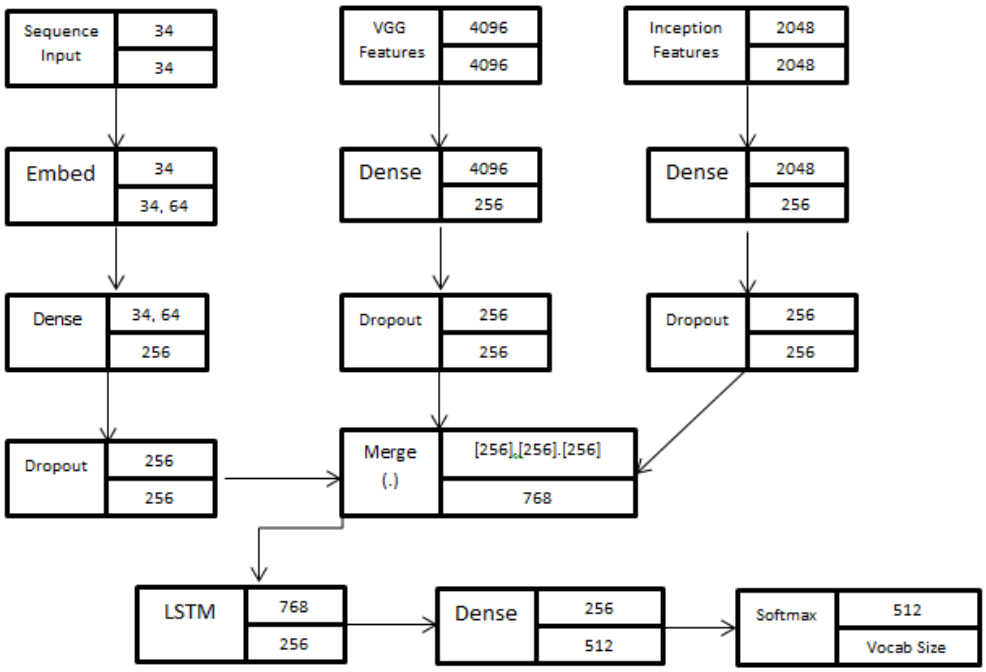


Figure 5.3: Ensemble Implant

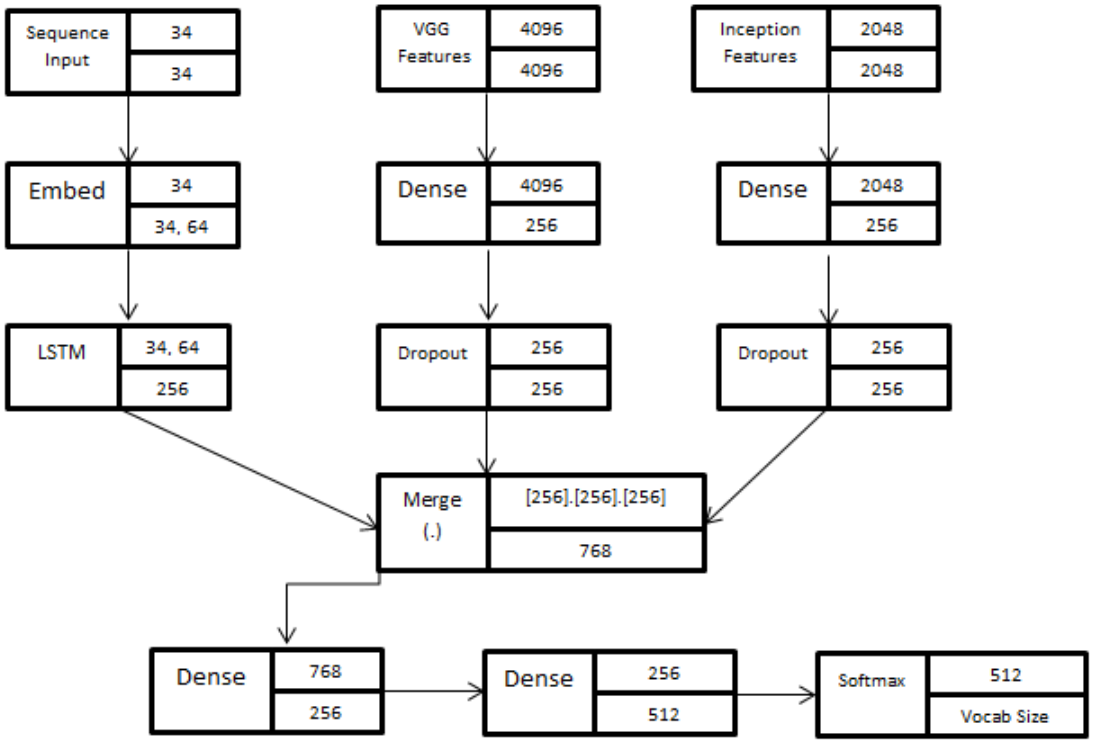


Figure 5.4: Ensemble Merge

6. Conclusions

- **Implant vs. Merge**

  Both implant and merge architecture are tested. In order to keep the flow of experiments in fast pace, models are tested on one caption. Although architectures were evaluated three times each to keep the experimentation consistent and then the average of scores was taken. Parameter taken into consideration is the word vector embedding length. To maintain the consistency state size of 256 is fixed in these architectures. These scores are given in table 6.1. Abbreviations used are "I-E64" = Implant Architecture with Embedding Word Vector Length 64 and "M-E64" = Merge Architecture with Embedding Word Vector Length 64.

BLEU Scores on One Caption				
	BLEU-1	BLEU-2	BLEU-3	BLEU-4
I-E64	0.2468	0.1212	0.0863	0.0448
I-E128	**0.3179**	**0.1544**	**0.1102**	**0.0496**
I-E256	0.3260	0.1509	0.1133	0.0544
M-E64	**0.3659**	**0.1774**	**0.1297**	**0.0613**
M-E128	0.3410	0.1721	0.1248	0.0582
M-E256	0.2959	0.1263	0.1059	0.0484

Table 6.1: BLEU Scores

From the given architectures and table 6.1 various observations can be made which are pointed out as such.

1. Consistent Advantage

   Above table 6.1 suggests that late merging of visual features with the textual information is more beneficial. Although the deviation in results of implant and merge is really narrow yet it performs consistently during three separate runs and has an advantage over implant. Merge architecture performs better than implant architecture and can generate captions of good quality even with smaller layers.

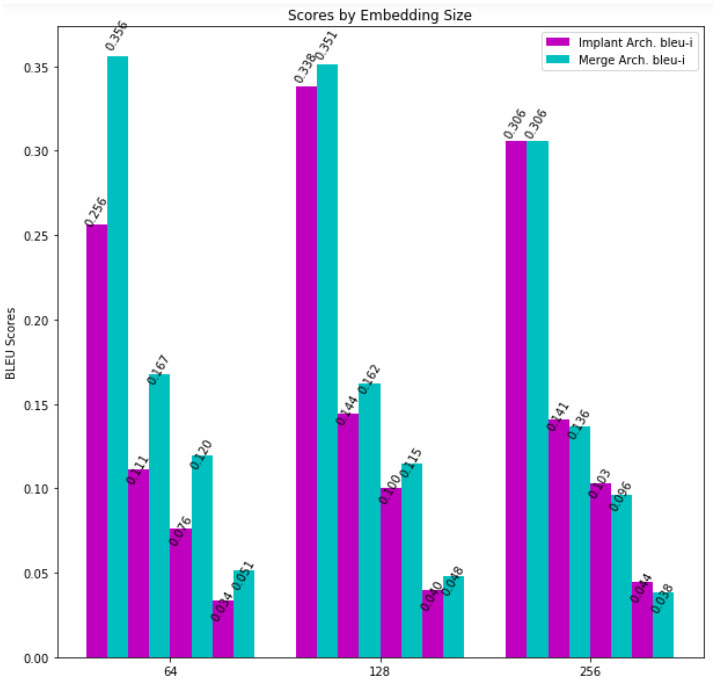


Figure 6.1: BLEU scores Graphs

2. Parameters Handled

   There is one more observation about the parameter handling of both architectures. On one hand implant architecture uses recurrent network to train on both visual and linguistic encodings at the same time thus there is a great increase in the vocabulary handled by recurrent network whereas merge architecture uses only textual encodings for recurrent network providing it smaller vocabulary on expense of more parameters at the point of merging of image and text in the network.

3. Training Loss

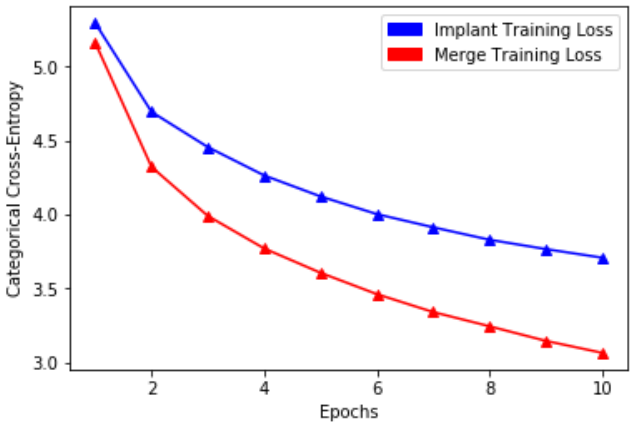


Figure 6.2: Training loss

As shown in figure 6.2, the training rate of implant architecture is rather slow than merge architecture. Merge architecture learns much faster and better than the implant architectures.

4. Performance over Embedding Word Vector Length
   Merge architecture tends to decrease in performance as embedding size grows. In this case embedding size of 64 is the best case scenario for merge whereas embedding size of 256 is the worst case scenario. On other side implant architecture shows inconsistent performance on variations in the length of embedded word vector.

5. Training Vocabulary Used
   On test captions, the amount of English words used for generation process is very low. Overall maximum vocabulary used is only 16% of the training data. It inferred that neural model finds it difficult to use infrequent words. It clearly suggests reducing the size of vocabulary will have minimal loss in performance.

6. Overall Inference
   The final observation is that late merging of image features with linguistic information gives better results than the one shown in implant architecture. Therefore, it concludes that if recurrent network should better be viewed as an encoder to encode the textual data rather than a generator in image captioning systems because if that was the case it would need an image in order to know what to extract from the image and generate as caption but that does not prove to be beneficial.

- **One versus Three versus Five Captions**
  As shown in experiments chapter, an effort to understand the working of LSTM is made. Here results are noted with variations in number of captions. These results are noted only for "M-E64" architecture since it performs better than the others and thus can tell us more about the way LSTM works. Table 6.2, represents the BLEU scores in the experiment performed. Abbreviation "M-E64-1" means merge architecture with word embedding length of 64 trained on one caption.

BLEU Scores				
	BLEU-1	BLEU-2	BLEU-3	BLEU-4
M-E64-1	0.3659	0.1774	0.1297	0.0613
M-E64-3	0.4138	0.2317	0.1639	0.0773
M-E64-5	**0.5978**	**0.3388**	**0.2501**	**0.1234**

Table 6.2: BLEU Scores variation with caption number

Table 6.2 clearly dictates that Long Short Term Memory Networks works better in terms of more number of captions. As we can see, results of model M-E64-5 are way better than the remaining two. One plausible reasoning for such a performance is long short term memory network are able to learn long term dependencies. More captions provide better context to the LSTM networks for the caption generation. Therefore such a score is observed.

- **Ensemble Models**

  From the experiment performed on ensemble model of visual feature extractors, it is observed that the model gives poor results. This model was indeed implemented in both views and evaluated on a single caption. The BLEU score observed on the ensemble model are shown in table 6.3. Abbreviation E-I-E64 means ensemble model in implant view trained with embedding word vector of length 64.

BLEU Scores				
	BLEU-1	BLEU-2	BLEU-3	BLEU-4
E-I-E64	0.3482	0.1507	0.1087	0.0513
E-M-E64	0.3560	0.1762	0.1299	0.0631

Table 6.3: Ensemble Models BLEU Scores

As clearly visible in table 6.3, these results show a slight variation with predefined merge and implant architecture. One plausible explanation is that both the visual feature extractor VGG-16 and InceptionV3 are trained on a same dataset. In the ensemble model, their last layer feature vector which represents the most important feature of the image is exploited. So somehow both architectures learnt the same features and these same features if combined are just a copy of each other.

- **Tested Samples**

  To conclude this chapter, some samples are presented which are tested on the overall best model so far proposed in this work. These samples are shown in figure 6.3.

	**dogs are running on grass**
	**man in red shirt riding bike**
	**girl standing with horse**
	**a girl and a boy in white shirt buying fruits**

Figure 6.3: Tested Samples

7. Recommendations and Future Work

In this work, two views of recurrent neural network in an image captioning systems are proposed. These views consider recurrent neural network as a generator and encoder respectively.

Recurrent neural network if considered as a component to generate captions, image access is necessary so that recurrent neural network will know what to extract from the image and generate as caption. But this view of taking recurrent neural network as a generator component does not give beneficiary result, so this is not the case seemingly.

However if another view of recurrent neural network is considered which is to encode the textual data instead of being viewed as a generative component, it makes sense because implant architecture does suffer great loss in performance than its competitor merge architecture. One possible explanation of this happening can be the prefix size. When textual data is provided as input to the recurrent network then at each time step prefix size is increased where there is addition of one word in the corresponding prefix. Each such prefix is framed in a fixed size vector.

In implant architecture, to encode is much difficult because of image features that are included with the textual features. Indeed implant architecture follow the basic rule of caption generation where every word in the caption is concatenated with the visual features. There are two reasons for this difficulty. First is the requirement of compressing the prefixes of the captions with the visual features in a vector of limited size or fixed size. Second being growth. There is a huge increase in the vocabulary size handled by recurrent neural network because each prefix is a sum of image and word. In merge architecture, this problem is removed since recurrent network task is to encode text rather than encoding image and text. Although there is a load on the layer in which both the image and encoded textual features are merged and fed into.

In general, implant architecture shows worse performance than the merge architecture. Therefore, recurrent neural network should be better viewed as an encoder or a learner of language representations which can further fed into the neural network that is predicting the caption based on previous predictions. If recurrent neural network was the primary component of generation module of image captioning systems than it would need the visual features but this thinking cause performance loss.

So, to conclude everything if merging of features is needed in neural network architecture then it would be better to first encode both the representations and then feed into a multimodal layer rather than putting everything into the same recurrent neural network via separate input pipeline. To this point, best view of recurrent neural network will be to learn linguistic representations.

A quite different point of view is also experimented and proposed in this work. This view is the combination of two visual feature extractors where both of them are convolutional neural networks and winners of ImageNet Competition. First features are extracted from both of the feature extractors and then both features are compressed into a fixed size vector separately. By passing through a feed forward layer and then combining with recurrent neural network everything goes through a soft-max layer that samples the most probable word required in the caption conditioned on the previously generated sequence.

Now this architecture after extracting visual features also follow both views named implant and merge discussed previously. Although two feature extractors are combined together yet there is no significant increase in performance. One plausible explanation of this can be that both feature extractors are designed to learn the same thing that is both were generated based on ImageNet classification and visual features were extracted from the last layer of both the networks that is the layer which possess the most important features of the scene or image. That is why these architectures are not able to attain the performance comparable the state of the art.

If both architectures are compared that is if comparison of implant and merge architecture is considered in terms of combined feature extractors' model, results are same. Merge still shows better performance even when two visual feature extractors are used.

One more thing to notice is that on increasing number of captions long short term memory networks that are preferred recurrent neural network in this work performs better. The performance of model is directly proportional to the increment in number of captions. The reason is to do with the working of long short term memory networks.

In this work, a discussion is also performed on the working of attention model and the visualization of feature vectors. This discussion shows that on applying attention to such type of learning architectures where focus on different regions is required model performance can be improvised. Attention brings out the activations of the regions other than the most important one thus playing a major role in such type of learning architectures. Not only that attention can also be applied in machine translation problems, sequence to sequence problems like question-answering, chat bots etc.

The inferences discovered in this work give invitation to more research on the applicability of merge architecture in pool of different domains. The science of transfer learning can leverage from the merge view of image captioning systems where the recurrent neural network that is used for captioning can be replaced with the general corpus trained neural language model. However, implant architecture cannot have such advantage because it needs both the image as well as the text to do its learning. Future researches can be done to see how a transfer learning based neural language model trained on a general corpus if transferred to image captioning systems in place of the recurrent neural network that is used to perform the encoding of the linguistics performs in the caption generator.

References

[1] Hodosh, Micah, Peter Young, and Julia Hockenmaier. "Framing image description as a ranking task: Data, models and evaluation metrics." *Journal of Artificial Intelligence Research* 47 (2013): 853-899.

[2] Rashtchian, Cyrus, et al. "Collecting image annotations using Amazon's Mechanical Turk." *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*. Association for Computational Linguistics, 2010.

[3] Bernardi, Raffaella, et al. "Automatic description generation from images: A survey of models, datasets, and evaluation measures." *Journal of Artificial Intelligence Research* 55 (2016): 409-442.

[4] Karpathy, Andrej. *Connecting Images and Natural Language*. Diss. Stanford University, 2016.

[5] Karpathy, Andrej, and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

[6] Vinyals, Oriol, et al. "Show and tell: A neural image caption generator." *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2015.

[7] Xu, Kelvin, et al. "Show, attend and tell: Neural image caption generation with visual attention." *International Conference on Machine Learning*. 2015.

[8] Oord, Aaron van den, Nal Kalchbrenner, and Koray Kavukcuoglu. "Pixel recurrent neural networks." *arXiv preprint arXiv:1601.06759* (2016).

[9] Kiros, Ryan, Ruslan Salakhutdinov, and Richard S. Zemel. "Unifying visual-semantic embeddings with multimodal neural language models." *arXiv preprint arXiv:1411.2539* (2014).

[10] Fang, Hao, et al. "From captions to visual concepts and back." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

[11] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

[12] Plummer, Bryan A., et al. "Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models." *Computer Vision (ICCV), 2015 IEEE International Conference on*. IEEE, 2015.

[13] Mao, Junhua, et al. "Deep captioning with multimodal recurrent neural networks (m-rnn)." *arXiv preprint arXiv:1412.6632* (2014).

[14] Banerjee, Satanjeev, and Alon Lavie. "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments." *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. 2005.

[15] 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.

[16] Bengio, Yoshua, et al. "A neural probabilistic language model." *Journal of machine learning research* 3.Feb (2003): 1137-1155.

[17] Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009.

[18] Szegedy, Christian, et al. "Rethinking the inception architecture for computer vision." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[19] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).

[20] Pan, Sinno Jialin, and Qiang Yang. "A survey on transfer learning." *IEEE Transactions on knowledge and data engineering* 22.10 (2010): 1345-1359.

[21] Papineni, Kishore, et al. "BLEU: a method for automatic evaluation of machine translation." *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002.

[22] Johnson, Justin, Andrej Karpathy, and Li Fei-Fei. "Densecap: Fully convolutional localization networks for dense captioning." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[23] Wang, Cheng, et al. "Image captioning with deep bidirectional LSTMs." *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016.

[24] Plummer, Bryan A., et al. "Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models." *Computer Vision (ICCV), 2015 IEEE International Conference on*. IEEE, 2015.

Plagiarism Report

ORIGINALITY REPORT

13%	8%	6%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Indian Institute of Information Technology, Allahabad Student Paper	6%
2	Johannesma, P. I. M., and H. F. P. van den Boogaard. "Stochastic Formulation of Neural Interaction", Mathematics of Biology, 1985. Publication	1%
3	webdocs.cs.ualberta.ca Internet Source	1%
4	arxiv.org Internet Source	1%
5	espace.curtin.edu.au Internet Source	<1%
6	"Computer Vision – ACCV 2016", Springer Nature, 2017 Publication	<1%
7	docplayer.net Internet Source	<1%
	www.mdpi.com	