# IMAGE PARAGRAPH CAPTIONING

*A Project Report*
*Submitted to show the progress*
*of the project on*

Image Paragraph Captioning

IIT PALAKKAD

*By*

**Muhammed Favas (142102007)**
**Parvathy (142203002)**
**Shubham Kanojia (142102015)**

*under the guidance of*

**Dr. Mrinal Das**
**mrinal@iitpkd.ac.in**

April 21, 2022

# ACKNOWLEDGEMENT

The contentment and elation that accompany the successful completion of any work be incomplete without mentioning the people who made it possible. We express our sincere thanks and most heartfelt sense of gratitude to the Deep Learning course instructor **Dr. Mrinal Das**, a perfectionist for his expert guidance and connoisseur suggestion. It is with deep sense of gratitude that we acknowledge our indebtedness to all our Teaching Assistance, for having extended their helping hand at all times.

$21^{st}$ April 2022 Muhammed Favas K
IIT Palakkad

# ABSTRACT

The main intention of this project is to gain hands-on experience in deep learning techniques like Convolutional Neural Networks (CNN) and Recurrent Neural Network (RNN) through image paragraph captioning task. In this project we implement three architectures for doing the task. Merge and inject architectures differ in terms of the position where the features encoded from the image incorporated to the model. Third architecture uses the concept of the visual attention along with the basic encoder decoder model. This report contains my contributions to the project in each models.

# Contents

# 1 INTRODUCTION

Image paragraph captioning is the process of generating meaningful sentences from an image, which describe the context of the image. Every day we encounter many images from various sources like the internet and we interpret them easily without any detailed captions. But automatic image captioning learn machines to generate captions corresponding to an image which will make them capable of doing tasks like automatic image indexing. Image indexing is important for Content-Based Image Retrieval (CBIR). Similarly, social media platforms like Facebook and Twitter can use the technique for generating captions from images.

With the evolution of deep learning, several approaches are introduced for solving this task, and still, research is going on. Most of the algorithms use Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) for doing the task along with some modifications in the architecture. CNN will encode images into useful features. Decoder will learn meaningful sequences from these encoded features.

# 2 IMAGE CAPTIONING ARCHITECTURES

In this project we implemented three encoder decoder architectures for image paragraph captioning task. Two encoder decoder architectures namely merge and inject taken from the paper 'Where to put the Image in an Image Caption Generator' by Marc Tanti et al. Third architecture uses visual attention mechanism with encoder decoder architecture, which is explained in the paper 'Show, Attend and Tell: Neural Image Caption Generation with Visual Attention' by Kelvin Xu et al.

## 2.1 Merge Architecture

In this architecture images and captions are encoded differently, concatenates these encoded features and encode with feed-forward network. In this architecture image is not introduced in the RNN network. This architecture follows recursive framing, in which model generate one at a time given both image and description generated so far as output. Architectural diagram of the merge model is shown in figure1.

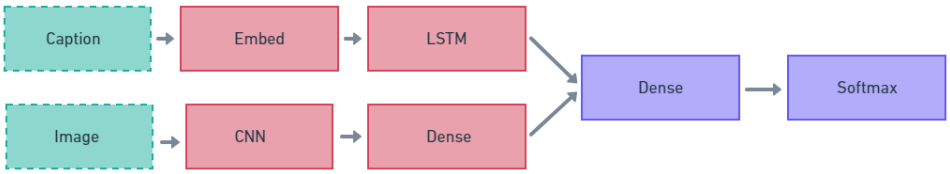



Figure 1: Merge encoder decoder architecture

**Contributions:**

- Done exploratory data analysis and found out frequent and least frequent tokens, maximum length of the captions fromthe training data.
- Extracted features from the images using VGG16 pre-trained model with second last layer as output. After extracting the features, its saved as '.pkl' file which can be retrieved during the training the encoder-decoder model which save the time.

- Implemented pipeline for training the model progressively, which will take one set of image and corresponding caption at time for training which will avoid the RAM issues.

## 2.2 Inject Architecture

In inject architecture encoded image features is introduced to the RNN model along with the encoded descriptions. So at every time step RNN decoder uses information from both image and captions for predicting the next word. There are three verities of inject architecture, namely init-inject, pre-inject and par-inject. architectural diagram of inject model is given in figure2. In this project I implemented pre-inject architecture where image is passed as first word into the RNN model followed by words in the description. .

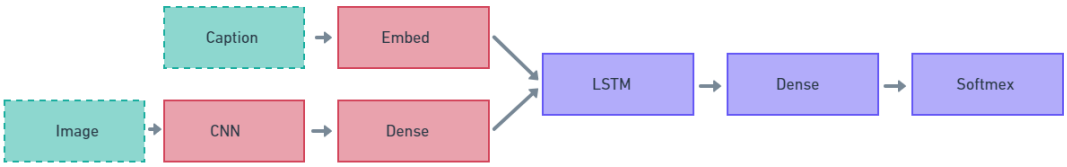



Figure 2: Inject encoder decoder architecture

**Contributions:**

- Pre-processed (lower casing and non alphanumeric character removal) the captions in the training data
- Created a vocabulary which contains all the tokens from the training captions whose frequency is greater than a frequency threshold.
- Created stoi(string to int) and itos(int to string) dictionaries which can be used to convert tokens into vectors of numbers and vice versa. The dictionaries includes special tokens SOS (represents start of sentence), EOS (represents end of sentence), PAD (represents padded position) and UNK (represents unknown words).
- Converted all captions into numerical vectors with the help of stoi dictionary.
- Applied padding for all descriptions for making them uniform size.
- Implemented the data loader pipeline which load data in batches. It get images and corresponding captions from the directory. Apply augmentation like random crop, horizontal flip, normalization and resize on the image. Convert descriptions into numerical value tokens.
- Encoder module consist of pre-trained inception-v3 along with one linear and dropout layers. Encoder will take image as input and encode image in to a vector of size mentioned in the hyper-parameters.
- Decoder will take captions and apply embedding layer on it. Concatenate encoded image features on the top of the these encoded vectors and input to an LSTM layer followed by a linear and dropout layer.
- Trained the model for 50 epochs and saved states of the encoder and decoder if validation loss is decreased.
- In the inference implemented a function which take image as input and generate captions from it.

## 2.3 Encoder Decoder with Visual Attention

Visual attention is based on the working of human eye. The attention models try to focus on important features from the image where it will get most useful information than the other regions. So incorporating attention technique along with basic encoder decoder model will produce accurate captions. The architectural diagram of the attention model is given in the figure 3. The architecture consist of three blocks encoder, attention and decoder.

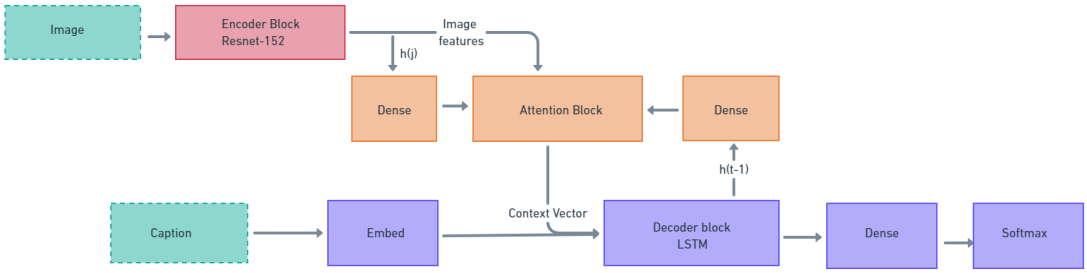



Figure 3: Encoder decoder architecture with visual attention

Encoder block uses Resnet-152 as feature extractor, which get the augmented images of size 224x224 and extract 2048 features with size 7x7.
In the attention block we used Behadanau soft attention which will take extracted features from the encoded block and outputs a context vector which is the weighted sum of the encoder features. The weights are calculated by score function which will take encoded features and previous hidden states from decoder block. after applying linear layer on both inputs it added together, applied tanh activation and then input to a linear layer followed by softmax activation which will gives the scores for each feature maps. The mathematical equation of score function is given below.

$$\alpha_{tj} = softmax(V_a \times tanh(U_a \times h_{t-1} + W_a \times h_j)) \tag{1}$$

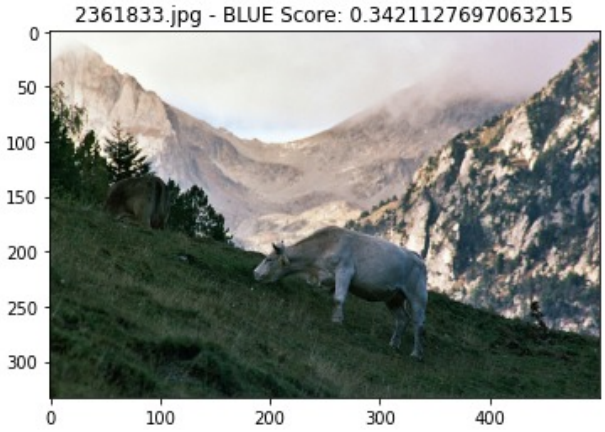
Where:
$\alpha_{tj}$ = weight of feature j at time step t
$V_a, U_a, W_a$ = weight matrix of linear layers
$h_{t-1}$ = decoder hidden state at time $t_1$
$h_j = j^{th}$ encoder feature map

Decoder block gets caption vectors and apply embedding layer on it. at each time step it will get context vector from the attention block and it will concatenate with current state input word (embedded vector). It will then pass in to an LSTM layer followed by a dropout layer and fully connected layer which produce next word in the sequence.
**Contributions:**

- Used captions pre-processing, vocabulary making and data loading pipelines same as that in the inject model.
- Implemented all the encoder, attention and decoder blocks.
- Implemented training and validation pipelines and saved the states of encoder and decoder blocks when validation loss is decreased.
- Implemented data inference pipeline for generating the captions in greedy search. Greedy search consider argmax or most probable word in each time step for generating the next word from the output probabilities.

# 3 RESULTS AND INFERENCE

Some the generated captions are shown in the figure4 and figure 5. BLUE score is used for measuring performance.
BLUE-N = Number of correct predicted N-grams / Number of total predicted N-grams.




```
 a cow is standing in the middle of a field. the cow is a light brown color and ha
s a lot of wool on it. the cow is standing on a field of grass. the cow is standin
g on a dirt path. the grass is green and lush. there is a large rock on the hill.
the sky above is blue with white clouds. the sky above the sheep is hazy and cloud
y.
```

Figure 4: Generated caption with BLUE score for image-1


```
 a stop sign is standing tilted upward to the top of the road. the stop sign is at
tached to a pole. the stop sign is attached to a pole. the sign is on a pole. the
sign is white with white letters. the sky is cloudy and overcast. the sky is blue
with white clouds. the sky is blue with white clouds.
```

Figure 5: Generated caption with BLUE score for image-2

# 4 CONCLUSIONS

Image captioning is an interesting topic in deep leaning which requires both CNN and RNN. Most of the image captioning algorithms are evolved by the modification of sequence sequence generators. In this project we tried three architectures, in which merge architecture gives good results. If train more epochs encoder decoder with visual attention may produce accurate results.

# References


[1] *Where to put the Image in an Image Caption Generator*. Marc Tanti, Albert Gatt, Kenneth P. Camilleri University of Malta, 2018.

[2] *Show, Attend and Tell: Neural Image Caption Generation with Visual Attention*, Kelvin Xu Jimmy Lei Ba Ryan Kiros Kyunghyun Cho Aaron Courville Ruslan Salakhutdinov Richard S. Yoshua Bengio, 2016