Image Captioning using CNN and LSTM

Sri Lakshmi Polavarapu

PSU ID: 914559661

Abstract

Image captioning is the challenging task in the field of data science for generating normal language captions by seeing an image. For generating a caption to an image, we must have very deep understanding of content behind the image as well as context. The two approaches used in this project, Convolutional Neural Network (CNN) is used for extracting visual representative features from pictures and Long Short-Term Memory (LSTM), which is a method in Recurrent Neural Network (RNN) is mainly used for generating data that matches the image with that caption. In this project I have used a combination of CNN and LSTM where one will extract features from images and the other will generate descriptive captions for those images.

Keywords: Image Captioning, CNN, RNN, LSTM, Flicker8k dataset

Introduction

Where there is a huge load of images on the web, there are methods to describe a image very descriptively. One of the most advanced models in artificial intelligence and neural networks is image captioning. Image captioning is used in various ways like for visual impairments, for visual assistances. Deep Learning has recently gained lots of attention because of its speed and accuracy when compared to other machine learning or neural network models.

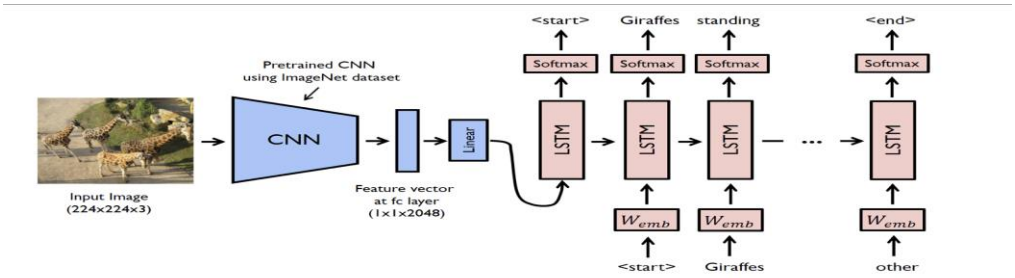
First, we classify the Flicker8k training dataset using various extractors. As we know the two components used in this project are CNN and LSTM. CNN here will deeply identify and analyze the image and extract important features from the image. The features extracted are fed into LSTM which is part of Regional Convolutional Neural Network (RNN) is used to give descriptive captions word by word to the image. A VGG16 model is used to extract image features. By combining these two features not only gives us descriptive images but also it gives us conceptual captions which are related to the input image.

Basic definitions of CNN RNN and LSTM:

1. **CNN**: Convolutional Neural Network is also known as ConvNets, which falls under the category of deep learning neural networks. It is primary build to handle for processing grid like structured data such as images, objects. CNN performs efficiently where there are tasks involving image identification and object detection. In real times CNN is growing widely in real world applications and has become a primary part of Computer Vision. Key components in CNN are convolutional layers, polling layers, activation functions and fully connected layer. All these are explain below. Few examples of CNN are VGG16, ResNet, AlexNet. But here in this project VGG16 architecture is used.

2. **RNN**: Recurrent Neural Networks are a class of neural networks which are designed to process the data by maintaining a hidden state that gets the information form past inputs. RNN has a unique architecture which is well suited for some of the tasks like text prediction, speech prediction, language processing prediction. Key components of RNN are hidden state, activation function and recurrent connections.

3. **LSTM**: long Short-Term Memory network is one of the most advanced architectures in RNN. LSTM is used when there is a huge dataset, and we need to capture long-range data. Same as RNN, LSTM is very useful in speech and text recognition, image processing and language processing. It has a unique memory mechanism where it allows it to selectively remember or forget data. Key components of LSTM are memory cells, forget gates, input and output gates.



Dataset:

The Flickr8k dataset is the most popular dataset, which is mostly used in the field of image captioning and computer vision. This dataset consists of a huge number of data's each image paired with descriptive caption associated with the image. As the images present in the dataset are visually clear we can develop a very narrative caption to them. The size of the dataset consists of 8000 images and with 5 textual descriptions for each image. The image in this dataset consists of a wide variety of scenes, objects, backgrounds and human activities.

Dataset link: https://www.kaggle.com/datasets/adityajn105/flickr8k

1
2
3
4
5
6
7
8
9

Proposed system

An encoder, Convolutional Neural Network (CNN) plays a primary role in encoding images to vectors. VGG16 model is used for extracting both easy and complicated features from the images, basically used for alterations. The images from the Flickr dataset are encoded by VGG16 and are passed into LSTM model, which is the specialized version of RNN. The input which we give to this system is an image of a fixed resolution. To verify whether the caption generated is like an actual caption or not, i used BLEU metric.

Proposed experiments

1. Dataset selection: for image captioning we need a huge dataset for processing, training the model along with descriptive captions. Then we input the data with images.
2. Data preprocessing: preprocess the data that is images by resizing the image into a consistent same size and same format. For captions, preprocess them by tokenizing which means converting them to either uppercase or lowercase, removing extra spaces or any punctuation.
3. Model architecture: we need to implement the proposed architecture for giving descriptive sentences to images and for that we need to combine Convolutional Neural Network (CNN) with Recurrent Neural Network (RNN). For RNN, for this project i have used Long Short-Term Memory (LSTM). We need to explore some CNN architectures like VGG or ResNet to test and extract the features of the images. Apart from that we need to explore variations of LSTM like hidden layers, total number of layers and performance.

4. Training: I trained the model by using the extracted features we got after performing CNN. For training the captions, we need to preprocess the captions. I trained my model using gradient descent technique and backpropagation. Optimized my model using loss function, for that i have used categorical cross-entropy along with that I used an Adam optimizer. The reason i used Adam optimizer is to minimize the loss function during my training period.
5. Evaluation: I evaluated the given captions by the trained model. The evaluation metrics I used was Bilingual Evaluation Understudy (BELU). After evaluating my caption using BELU, i crosschecked my output of the generated caption with the actual caption.
6. Analysis: by observing the evaluation score, we need to analyze the performance
7. Testing: after we train the model, we need to pass some unseen images to the model to get the captions. Based on the results we decide how the model works.

Proposed method

Proposed method for image captioning is the combination of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) to extract features from image and generate descriptive sentences as captions.

1. **CNN**: used for image extraction of features. For this process we give a pre-trained CNN model, in here i have given VGG16 to extract some high-level features from images which are given as input. CNN here is used to encode the given image that is change the raw pixel image data into an image where computer will easily understand and classify, identifies the image clearly. The output which we get after passing it through CNN model is a vector.

1
2
3
4
5
6
7
8
9

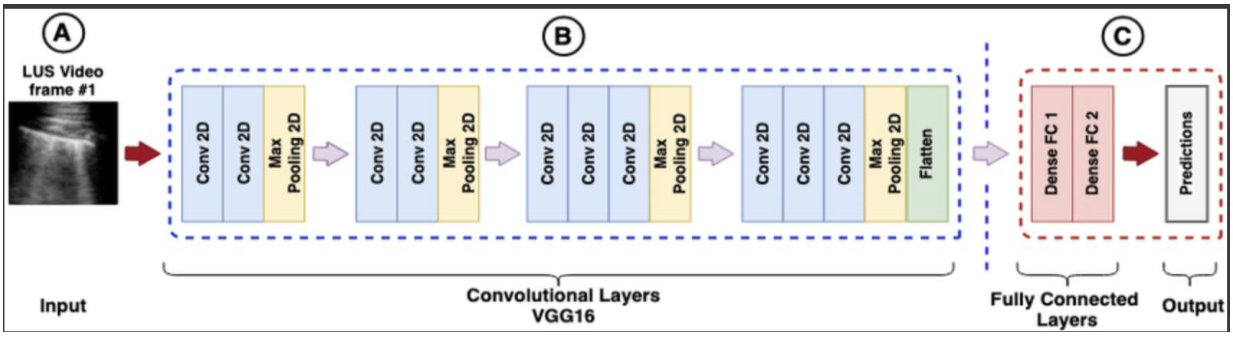
2. **LSTM**: the vector output which we got during CNN extraction is given as input to LSTM model. The LSTM takes the vector along with a token as a input and then generates a sequence of words which forms a caption. LSTM predicts the next word to form a complete sense of caption.

Architecture:

The architecture used in this project is VGG16, which consists of 16 weight layers. It's used everywhere in image classification problems because of its simple and effectiveness. It is mainly used for extracting features in various tasks like image classification, detecting an object and in image captioning.

VGG16 Architecture

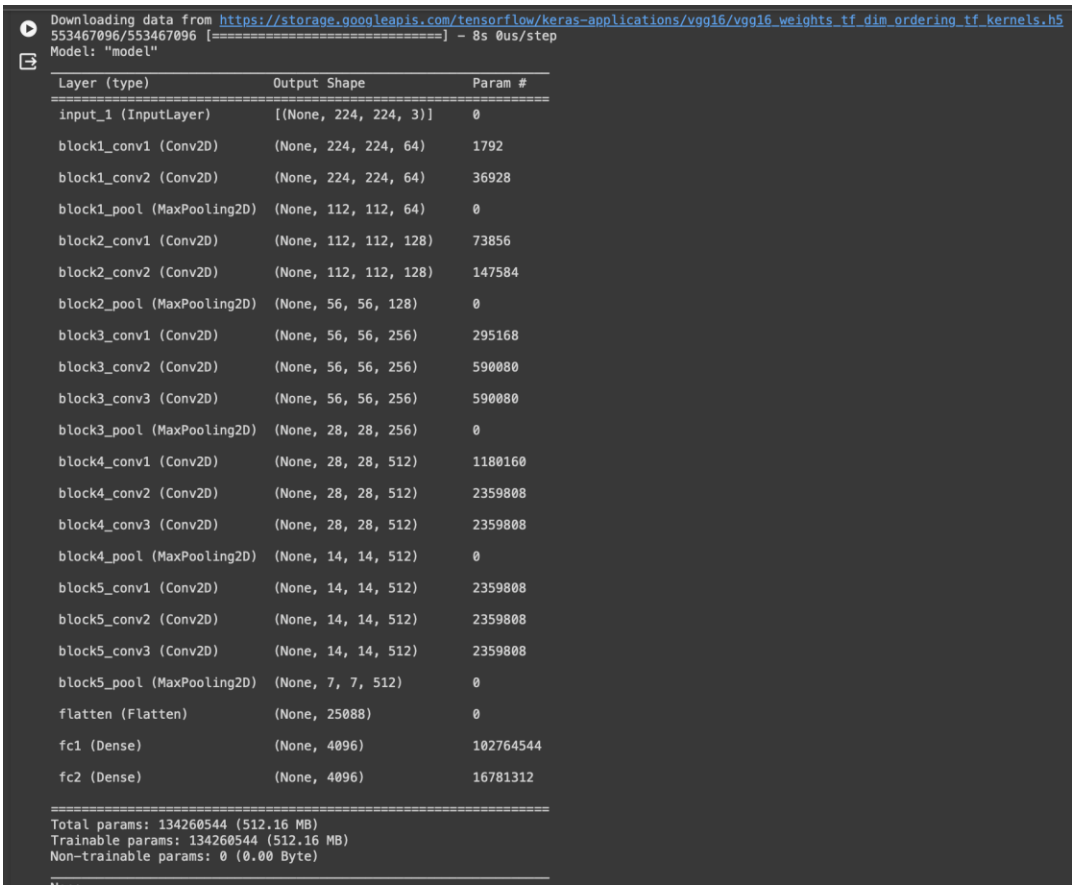
1. Input layer: the image is given as an input to the model
2. Convolutional layers: 13 convolutional layers are used which are divided into 5 blocks each block is followed by max polling. Each block has many convolutional layers with a activation function called ReLU activation. The function of max-polling layers is to decrease the dimensions of features.
3. Fully connected layers: after convolutional blocks, VGG16 has three fully connected layers. Each fully connected layer is followed by activation function except the output layer.



There are three layers in VGG16

1. Convolutional layer: in total there are 13 convolutional layers, which have different channels in each layer. Each layer has an activation function, ReLU for non-linearity.
2. Max-polling layer: after every two convolutional layers there is a max-polling layer, which decreases the dimensions of the features by reducing its height and weight to half.
3. Fully connected layers: this layer is used at the end of the network, where it performs classification based on the output of convolutional layers.

1
2
3
4
5
6
7
8
9



```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels.h5
553467096/553467096 [==============================] - 8s 0us/step
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 224, 224, 3)]     0

block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792

block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928

block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0

block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856

block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584

block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0

block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168

block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080

block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080

block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0

block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160

block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808

block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808

block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0

block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808

block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808

block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808

block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0

flatten (Flatten)            (None, 25088)             0

fc1 (Dense)                  (None, 4096)              102764544

fc2 (Dense)                  (None, 4096)              16781312

=================================================================
Total params: 134260544 (512.16 MB)
Trainable params: 134260544 (512.16 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

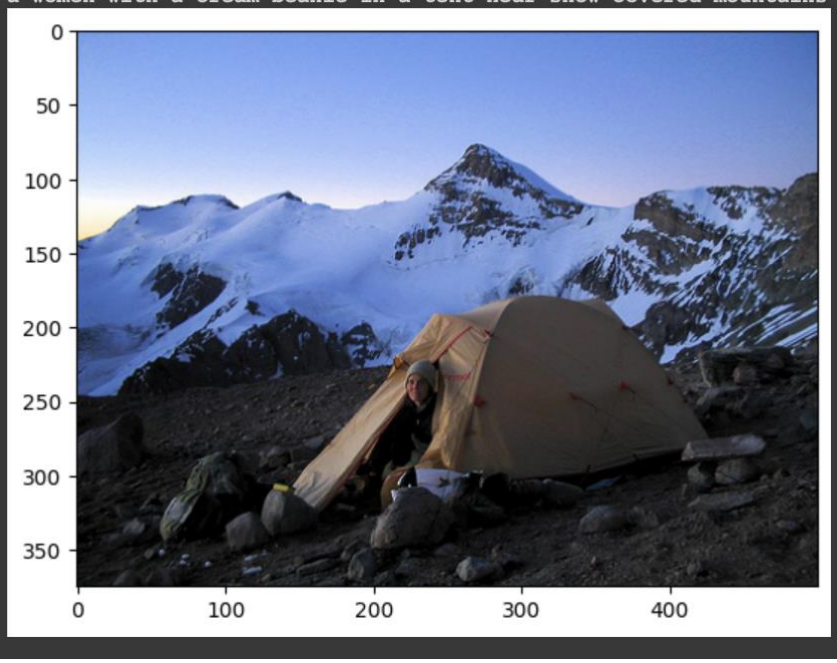
Results:

1
2
3
4
5
6
7
8
9

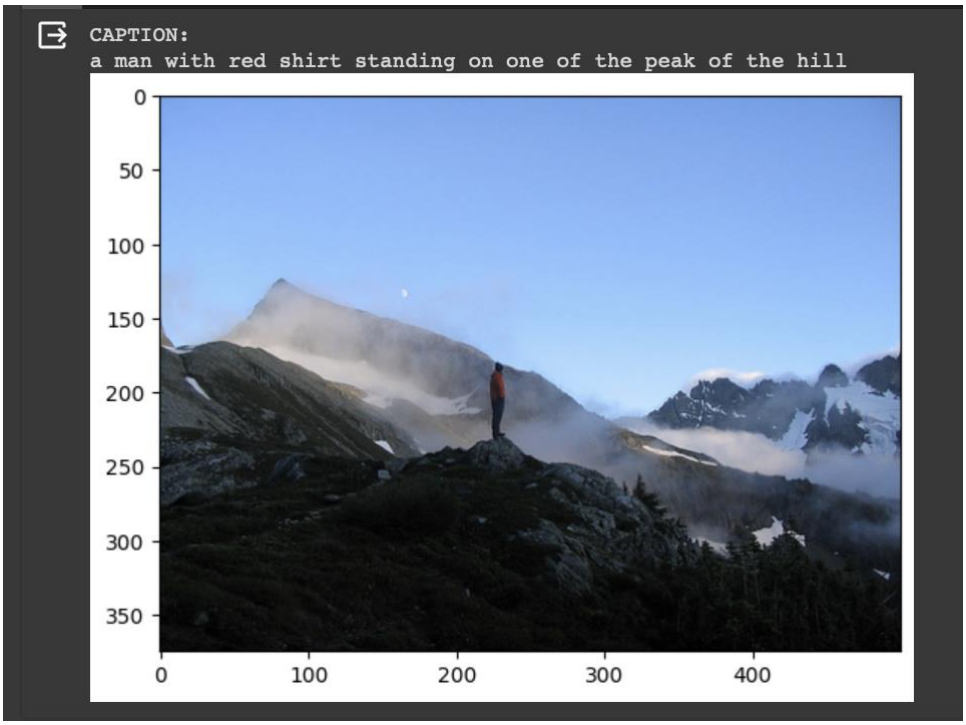
CAPTION:
a man with red jacket and a beanie is on top of the snow mountain

1
2
3
4
5
6
7
8
9

```
----------------------Caption----------------------
a women with a cream beanie in a tent near snow covered mountains
```



1
2
3
4
5
6
7
8
9



Challenges:

The challenges which I faced while doing this project were:

1. Constructing and running the architecture. As it is a deep neural network which involves many complicated models. I could not finely tune the hyperparameters, which affected my performance.
2. My second problem was the CNN and LSTM model was very time consuming. Running 100 epochs took me 30 minutes.

Limitations:

1. Sometime the model might not correctly scan the image which might lead to undescriptive captions.
2. The performance of the model purely depends on training quality and quantity data. If the data is not trained properly, we might not get the correct result.
3. When the images are not properly understood by the model, it might face difficulty in generating captions.

Conclusions:

Here in this project for image captioning I have used a combination of CNN and LSTM for image extraction and for generating captions for those input images. The experiments i conducted will evaluate the effective performance of both CNN and LSTM in giving descriptive sentences for images. Our combinational approach of CNN and LSTM demonstrates very strong performance in generating captions which accurately describe the content of images which are present in the dataset. The architectural choices I made for CNN as well as LSTM have increased the performance of the model. Fine tuning of images has increased the probability of caption accuracy rate. The evaluation metric which I used, BLEU, has improved the quality of captions. After using this metric, the caption that the model gave was similar to that of human given captions.

So overall, the project contributes to the advancement of image captioning by giving a efficient captions to the input images by using CNN and LSTM model. This project is a deep learning technique which bridges a gap between visual and textual descriptions.

Future scope:

As image captioning is very helpful for people to identify the image properly, this project can be developed into a website using some front-end languages and can add voice while describing the image. By using python, we can develop voice assistance on the website, which will help people who have problems with visibility. This website then can help who are disable either visually or if they can't hear properly, they can read the descriptive sentence provided by the model and can understand. Instead of CNN, we can try using different models like ResNet or GoogleNet to get more accurate results.

References:

[1] Oriol, Alexander, Samy, Dumitru, "Show and Tell: A Neural Image Caption Generator", IEEE paper on Computer Vision and Pattern Recognition, 2015

[2] Swarnim, Ravi, "Image Caption Generator Using CNN and LSTM", ITEE Journal paper on Deep Learning, 2022

[3] Himanshu, Anand, "Incorporating external knowledge for image captioning using CNN and LSTM", paper on Natural Language Processing, 2020

[4] Rohit, Omini, Rutuja, "Image Caption Generator using CNN and LSTM – GUI Application", IJARIIE paper on Deep Learning, 2022

[5] Cheng, Haojin, Christian, Christoph, "Image Captioning with Deep Bidirectional LSTMs", ACM international conference, 2016