CAPTIONING IMAGES USING CNN-LSTM DEEP NEURAL NETWORKS AND FLASK

Rutvik Bhalekar [#1], Ajaykumar Gupta [#2], Sanket Bhangale[#3], Shrikant Sanas[#4]

[1]Student, *Information Technology Department, Padmabhushan VasantDada Patil College of Engineering*
[2]Student, *Information Technology Department, Padmabhushan VasantDada Patil College of Engineering*
[3]Student, *Information Technology Department, Padmabhushan VasantDada Patil College of Engineering*
[4]Professor, *Information Technology Department, Padmabhushan VasantDada Patil College Of Engineering*

Abstract - *Captioning images automatically is one of the hearts of the human visual system. There are various advantages if there is an application which automatically caption the scenes surrounded by them and revert back the caption as a plain message. In this project, we present a model based on CNN-LSTM neural networks which automatically detects the objects in the images and generations descriptions for the images. This model can perform two operations. The first one is to detect objects in the image using convolutional neural networks and the other is to caption the images using RNN based LSTM. Interface of the model is developed using flask rest API, which is a web development framework of python. The main use case of this project is to help visually impaired to understand the surrounding environment and act according to that.*

Key Words: *Captions, Image, CNN,RNN.*

1. Introduction

When we see a picture our brain can easily tell what the image is about, but can a computer tell what the image represents? Computer vision researchers have worked hard on this and consider it impossible until now! With the development of deep learning reading strategies, access to large databases and computing power, we can create models that can make picture captions.

Captioning images automatically is one of the hearts of the human visual system. There are various advantages if there is an application which automatically caption the scenes surrounded by them and revert back the caption as a plain message. An image caption generator is a function that incorporates computer vision and natural language processing concepts to visualize the image and interpret it in the native language such as English. In this project, we present a model based on CNN-LSTM neural networks which automatically detects the objects in the images and generations descriptions for the images. This model can perform two operations. The first one is to detect objects in the image using convolutional neural networks and the other is to caption the images using RNN based LSTM.

2. Survey Existing System

The first paper is Deep Learning based Automatic Image Caption Generation. This paper aims to generate automated captions by reading the content of the image. At present images are defined by human intervention and become an impossible task in big business details. The image website is provided as an integral part of the Convolutional Neural Network (CNN) to create a "thought vector " that removes the features and nuances from the image and the RNN (Recurrent Neural Network) decoder is used to translate the features and features provided by the image to find the sequence, a logical description of the image. In this paper, they systematically analyze the various methods of image depth with network-based networks and fictional models to conclude with a highly efficient model with good optimization. The demerit of this approach is the GRU used for sequence generation.

The Second paper is Transfer learning-based Object Detection by using Convolutional Neural Networks.In tis paper, they use transfer learning model on CUB 200-211 dataset to detect object from the image.CNN for image classification works as it takes input in the

form of image and gives the output as a category of that image. The CNN convolves already learned feature along with the input data,
and it uses a two-dimensional convolution layer.That is ideal for the processing of the two dimensional pictures.They use vgg16,alexnet,GoogleNet,RestNet,Xception transfer learning model.

The Third paper is Automatic Indonesian Image Caption Generation using CNN-LSTM Model and FEEH-ID Dataset. Image capture is a challenge in computer vision research. This paper expands the study on the default generation of photo captions to the size of Indonesia. An Indonesian sentence description is made with unlabeled pictures. The database used is FEEH-ID, this is the first Indonesian image caption database. This study is important due to the unavailability of a caption of photographic captions in Indonesian. This paper will compare the test results in the FEEHID database with English, Chinese and Japanese data using CNN and LSTM models. The performance of the proposed model on the test set gives a promising result of 50.0 BLEU-1 results and 23.9 BLEU-3 results, which is more than the average of the Bleu test results in other information. The integration model between CNN and LSTM shows the positive effects of the FEEH-ID database.

3.Objective.

1. Develop a deep learning model using commodity hardware.
2. Train the CNN-LSTM model to predict the caption for a given photo .
3. Allow individuals to get the caption of their specified photo.

4. Proposed System

4.1 Dataset

The dataset that we used is the Flickr 8k consisting of 8,000 images that are each paired with five different captions which provide clear descriptions of the salient entities and events. The Flickr8k_text contains the text information.

4.2 Data Pre Processing

Before we trained our model we performed data preprocessing on our data. We used transfer leaning in this project so that we can used CNN for the features extraction of the 8000 images and after that we created a description file which stored all the captions of the 8000 images but we removed all the single A's and punctuation from the captions before saving them in the description file.We used 6000 images for the training part. The below figure is of the data preprocessing of the data.To get a caption, we needed to define a size for the caption so that the model would come to understand that it is the max length which should not be exceeded so therefore we defined a vocabulary size of 32 as well.

```
Dataset: 6000
Descriptions: train=6000
Photos: train=6000
{'1000268201_693b08cb0e': ['startseq child in pink dress is climbing up set of stairs in an entry way endseq',
  'startseq girl going into wooden building endseq',
  'startseq little girl climbing into wooden playhouse endseq',
  'startseq little girl climbing the stairs to her playhouse endseq',
  'startseq little girl in pink dress going into wooden cabin endseq'],
 '1001773457_577c3a7d70': ['startseq black dog and spotted dog are fighting endseq',
  'startseq black dog and tricolored dog playing with each other on the road endseq',
  'startseq black dog and white dog with brown spots are staring at each other in the street endseq',
  'startseq two dogs of different breeds looking at each other on the road endseq',
  'startseq two dogs on pavement moving toward each other endseq'],
```

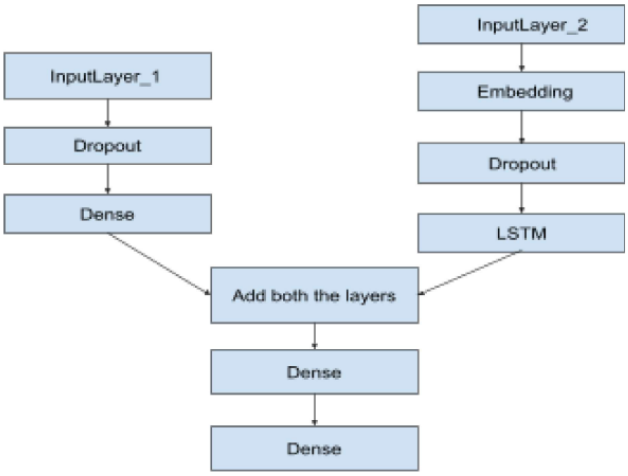
4.3Model Architecture

Model Architecture defines the logical connections of various functions used in the model creation and training.We will be using a generator method which yields the result in batches , we are using this because we don't have memory to store 2048features of each image that we will train. Defining the CNN-RNN model: To define the structure of the model, we will be using the Keras Model from Functional API. It will consist of three major parts:

Extractor – The feature extracted from the image has a size of 2048, with a dense layer, we will reduce the dimensions to 256 nodes.It will be used as input1.

Sequences – An embedding layer contains the captions as input2, followed by the LSTM layer.

Decoder – By combining the output from the two layers above, we will process the dense layer to make the final prediction. The final layer will contain a number of nodes equal to our vocab size.



The above diagram shows the layers that are used in our project

Our model architecture is as follows:

Consider inputs{input_1,input_2}as inputs to our model and output{out_0}, where input_1 is the 2048 feature vector of that image, input_2 is the caption sequence and out_0 is the output of the model that it predicted. For example:

Input_1	Input_2	Out_0
feature	start	person
feature	start,person	wearing
feature	start,person,wearing	skis
feature	start,person,wearing, skis	looking
feature	start,person,wearing, skis,looking	at
feature	start,person,wearing, skis,looking,at	framed
feature	start,person,wearing, skis,looking,at,framed	pictures
feature	start,person,wearing, skis,looking,at,framed,pictures	set
feature	start,person,wearing, skis,looking,at,framed,pictures,set	up
feature	start,person,wearing, skis,looking,at,framed,pictures,set,up	in
feature	start,person,wearing, skis,looking,at,framed,pictures,set,up,in	the
feature	start,person,wearing, skis,looking,at,framed,pictures,set,up,in,the	snow
feature	start,person,wearing, skis,looking,at,framed,pictures,set,up,in,the,snow	end

After the model is trained then we will make a flask file so that we can deploy the model on the internet.

5. Hardware and Software Requirements

a) Hardware:

- CPU- Intel Core i3 5th gen/ AMD Ryzen 3 1300 or above.
- GPU- Nvidia GTX 1050 or above.
- RAM- 4 GB or above.
- HDD- 2GB free space.

b) Software:

- Windows 7/8/8.1/10 (32 bit/64 bit) or Ubuntu Linux or Mac OS.
- Python 3.1 with flask, Keras & Tensor flow libraries installed

6.Result and Discussion

TEST	SCORE
BLEU-1	0.53
BLEU-2	0.28
BLEU-3	0.18
BLEU-4	0.08

BLEU metric scales translate from 0 to 1 scale, but are usually expressed as a percentage. The closer you are to 1, the more relevant the translation becomes. Simply put, BLEU metric estimates how many words go beyond a given translation compared to a reference translation, giving high scores on consecutive words.There are 4-ngrams in BLEU which basically means that it compares the prediction which model made and the actual caption of the photo and gives a accuracy for it. A good BLEu score is above 0.50 and as you can see in the table above we have achieved that through this model.After the model was made we created a flask file and tested our model on the web using a local host.

7. CONCLUSIONS

We have successfully created an image caption generator.For this project, we have used the CNN-RNN model by building an image caption generator. Another important point to note is that our model is data-dependent, and therefore, cannot predict words outside its vocabulary. We used a small database with 8000 images. For production-level models, we need larger dataset that can produce better accuracy models.

8.REFERENCES

[1] V. Kesavan, V. Muley and M. Kolhekar, "Deep Learning based Automatic Image Caption Generation," *2019 Global Conference for Advancement in Technology (GCAT)*, Bangalore, India, 2019, pp. 1-6, doi: 10.1109/GCAT47503.2019.8978293.

[2] B. Bamne, N. Shrivastava, L. Parashar and U. Singh, "Transfer learning-based Object Detection by using Convolutional Neural Networks," *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Coimbatore, India, 2020, pp. 328-332, doi: 10.1109/ICESC48915.2020.9156060.

[3] E. Mulyanto, E. I. Setiawan, E. M. Yuniarno and M. H. Purnomo, "Automatic Indonesian Image Caption Generation using CNN-LSTM Model and FEEH-ID Dataset," 2019 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), Tianjin, China, 2019, pp. 1-5, doi: 10.1109/CIVEMSA45640.2019.9071632.

[4] François Chollet, Python Keras Documentation, *https://keras.io*