Automated Caption Generator Using Beam Search: A Deep Learning Process

Ashna Rahim[1], Asst.Professor. Suranya.G[2]

[1]M-Tech, Applied Electronics, Electronics and Communication Engineering, Ilahia College of Engineering and Technology, Kerala, India

[2]Electronics and Communication Engineering, Ilahia College of Engineering and Technology, Kerala, India

---***---

Abstract— *Image Caption Generator deals with generating captions for a given image. The semantic meaning in the image is captured and converted into a natural language. The capturing mechanism involves a tedious task that collaborates both image processing and computer vision. The mechanism must detect and establish relationships between objects, people, and animals. The aim of this paper is to detect, recognize and generate worthwhile captions for a given image using beam search. The proposed method focuses on deep learning to further improve upon the existing image caption generator system. Experiments are conducted on the Flickr 8k dataset using python language to demonstrate the proposed method.*

Keywords—Image, capturing, generator, beam search, CNN, RNN, deep learning

I.INTRODUCTION

Making a computer system to detect the image and produce a description using natural language processing is an exigent task, which is called an image caption generator system. Generating a caption for an image involves various tasks such as understanding the higher levels of semantics and describing the semantics in a sentence by which human can understand. In order to understand the higher levels of semantics, the computer system must learn the relationships between the objects in a given image. Usually, communication in human beings occurs with the help of natural language, so developing a system that produces descriptions that can be understandable by human beings is a challenging goal. There are several steps to generate captions, such as understanding visual representation of objects, establishing relationships among the objects and generating captions both linguistically and semantically correct. It consists of object detection, feature extraction, Convolution Neural Network (CNN) for feature extraction and for scene classification, Recurrent Neural Network (RNN) for human and objects attributes, RNN encoder and a fixed length RNN decoder system.

1.1Deep Learning

Deep learning methods aim at learning feature hierarchies with features from higher levels of the hierarchy formed by the composition of lower level features.

Automatically learning features at multiple levels of abstraction allow a system to learn complex functions mapping the input to the output directly from data, without depending completely on human-crafted features.

Deep learning algorithms seek to exploit the unknown structure in the input distribution in order to discover good representations, often at multiple levels, with higher-level learned features defined in terms of lower- level features

Fig -1: Deep Learning

The hierarchy of concepts allows the computer to learn complicated concepts by building them out of simpler ones. If we draw a graph showing how these concepts are built on top of each other, the graph is deep, with many layers. For this reason, we call this approach to AI deep learning. Deep learning excels on problem domains where the inputs (and even output) are analog. Meaning, they are not a few quantities in a tabular format but instead are images of pixel data, documents of text data or files of audio data. Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction.

1.2 TensnsorFlow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google, TensorFlow is Google Brain's second- generation system. Version 1.0.0 was released on February 11, While the reference implementation runs on single devices, TensorFlow can run on multiple CPUs and GPUs (with optional CUDA and SYCL extensions for general-purpose computing on graphics processing units).

Tensor Flow is available on 64-bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS. Its flexible architecture allows for the easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices.

1.3 Keras

Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation and developer guides. Keras contains numerous implementations of commonly used neural network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code. The code is hosted on GitHub, and community support forums include the GitHub issues page, and a Slack channel. Keras is a minimalist Python library for deep learning that can run on top of Theano or Tensor Flow. It was developed to make implementing deep learning models as fast and easy as possible for research and development. It runs on Python 2.7 or 3.5 and can seamlessly execute on GPUs and CPUs given the underlying frameworks. It is released under the permissive MIT license.

Beam Search

Beam search is a search algorithm that explores a graph by expanding the most promising node in a limited set. Beam search is an optimization of best-first search that reduces its memory requirements. Best-first search is a graph search which orders all partial solutions (states) according to some heuristic. But in beam search, only a predetermined number of best partial solutions are kept as candidates.

2. PROPOSED METHODOLOGY

The Proposed methodology for generating captions with the detection and recognition of objects using beam search deep learning is shown in Fig.2. It consists of Input image, feature extraction, Convolution Neural Network (CNN) for feature extraction, Recurrent Neural Network (RNN) encoder and a fixed length RNN decoder system, Beam Search for generating captions.

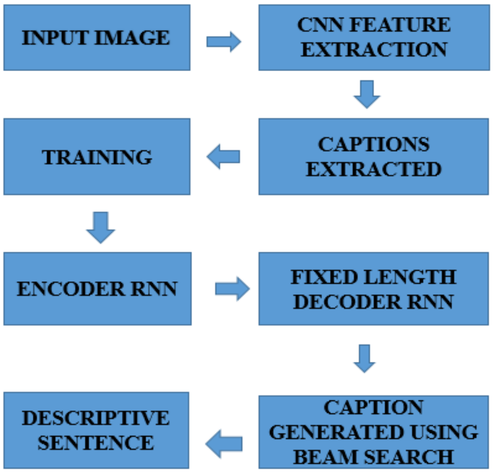


Fig. 2

The steps for generating captions using beam search is as follows.

Step 1: Input Image

In this step, the input image is provided.

Step 2: Feature Extraction

In this step, the features in the image are extracted using CNN Inception v3.

Step 3: Caption Extraction

In this step, the captions are extracted and saved into caption.txt with image id.

Step 4: Encoder and Decoder

In this step, the label strings were subjected to an encoder RNN for encoding the label strings to a proper format, and the resultant variable length string is subjected to a fixed length decoder for converting to a fixed length descriptive sentence.Here RNN brings maximum number of partial solutions.

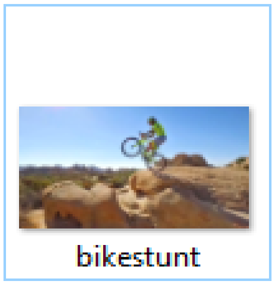
Step 5: Beam Search

In this step, best captions are generated by beam search with less memory requirement.

3. EXPERIMENTAL ANALYSIS

The aim of this paper is to propose a deep learning method for generating captions using beam search. The dataset details have been described in this section. The experimental evaluation of the proposed methodology is done by Flickr 8k dataset obtained from [11], from 8000 images, just for simplicity, only two images were subjected to the proposed methodology and the results were obtained. Fig. 3 represents the input image to which the caption needs to be generated. Fig. 4 describes the caption generation process, first, the input image is subject to the feature extraction using the feature_extraction command to extract the features in that image and captions are generated using beam search. The proposed model accurately generated a caption that a man on a bike jumps a ramp Fig. 4. The model is also evaluated with Fig. 5, the model accurately generated caption as shown in Fig. 6.

Fig. 3. Put Image-1



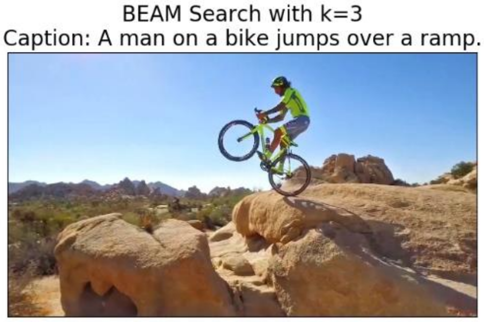

BEAM Search with k=3
Caption: A man on a bike jumps over a ramp.

Fig. 4. Output: Caption generated

Fig. 5. Input Image-2

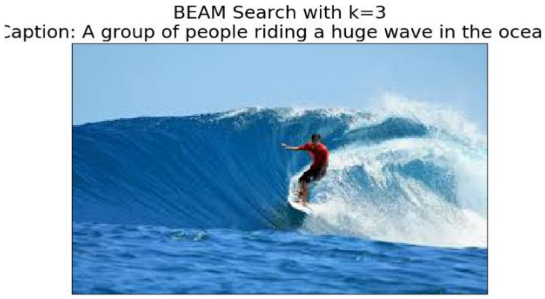

BEAM Search with k=3
Caption: A group of people riding a huge wave in the ocea

Fig.6.Output:Caption Generated

1.Import all the necessary packages

2.Getting and performing data cleaning

3.Extracting the feature vector from all images

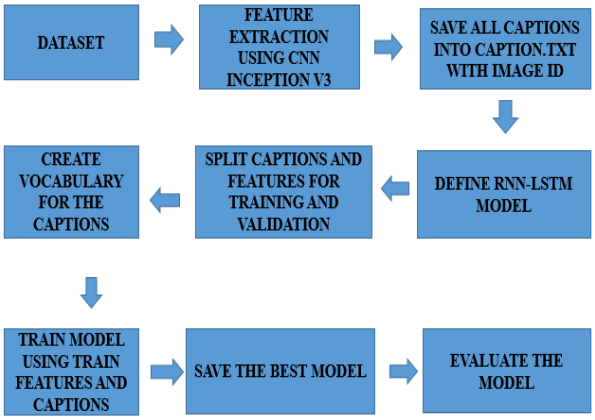
4.Loading dataset for Training the model

5.Tokenizing the vocabulary

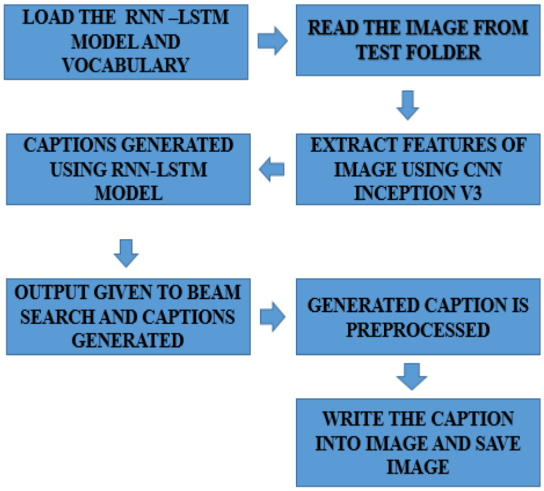
6. Create Data generator

7.Defining the CNN-RNN model

8.Training the model


DATASET
FEATURE EXTRACTION USING CNN INCEPTION V3
SAVE ALL CAPTIONS INTO CAPTION.TXT WITH IMAGE ID
DEFINE RNN-LSTM MODEL
SPLIT CAPTIONS AND FEATURES FOR TRAINING AND VALIDATION
CREATE VOCABULARY FOR THE CAPTIONS
TRAIN MODEL USING TRAIN FEATURES AND CAPTIONS
SAVE THE BEST MODEL
EVALUATE THE MODEL

9. Testing the model


LOAD THE RNN –LSTM MODEL AND VOCABULARY
READ THE IMAGE FROM TEST FOLDER
EXTRACT FEATURES OF IMAGE USING CNN INCEPTION V3
CAPTIONS GENERATED USING RNN-LSTM MODEL
OUTPUT GIVEN TO BEAM SEARCH AND CAPTIONS GENERATED
GENERATED CAPTION IS PREPROCESSED
WRITE THE CAPTION INTO IMAGE AND SAVE IMAGE

4.CONCLUSION

In this paper, a deep learning method for image caption generation using beam search is presented, the proposed method was applied to a Flickr 8k dataset. The proposed deep learning methodology generated captions with more descriptive meaning, more speed and less memory requirement than the existing image caption generators. Applying hybrid image caption generator model with location specifications can be developed in the future for more accurate captions.

ACKNOWLEDGEMENT

There are many people who have helped directly or indirectly in the successful implementation of my project. I would like to have this opportunity to thank one and all. I thank **GOD Almighty** for the blessings he had bestowed on me which resulted in finalization of my project. I would like to extend my sincere thanks to my staffs and supervisors for their immense help and contribution in this endeavour. I extend my heartfelt gratitude to Principal, HOD, project guide and coordinators for their unfailing support in this work.

REFERENCES

[1].https://ieeexplore.ieee.org/document/8728516

[2] .P. Hede, P. Moellic , J. Bourgeoys , M. Joint , C. Thomas, Automatic generation of natural language descriptions for i mages, in: Proceedings of the Recherche Dinformation Assistee Par Ordinateur, 2004.

[3].J. Donahue, Y. Jia , O. Vinyals , J. Hoffman , N. Zhang , E. Tzeng , T. Darrell , DeCAF: a deep convolutional activation feature for generic visual recognition, in: Proceedings of The Thirty First International Conference on Machine Learning, 2014, pp. 647–655.

[4]. A. Farhadi , M. Hejrati , M.A. Sadeghi , P. Young , C. Rashtchian , J. Hocken- maier , D. Forsyth , Every picture tells a story: Generating sentences from images, in: Proceedings of the European Conference on Computer Vision,,2010, pp. 15–29.

[5]. M. Hodosh, P. Young , J. Hockenmaier , Framing image description as a rank- ing task: data, models and evaluation metrics, J. Artif. Intell. Res. 47 (2013) 853–899 .

[6]. Y. Yang , C.L. Teo , H. Daume , Y. Aloimono , Corpus-guided sentence generation of natural images, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2011, pp. 4 4 4–454 .

[7]. .R. Socher , A. Karpathy , Q.V. Le , C.D. Manning , A.Y. Ng, Grounded composi- tional semantics for finding and describing images with sentences, TACL 2 (2014) 207–218

[8].O. Vinyals , A. Toshev , S. Bengio , D. Erhan , S3how and tell: a neural image cap- tion generator, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3156–3164 .