International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 07 | July 2021 www.irjet.net p-ISSN: 2395-0072

Image Caption Generator using CNN-LSTM

Preksha Khant[1], Vishal Deshmukh[2], Aishwarya Kude[3], Prachi Kiraula[4]

[1-4]Department of Computer Science and Engineering, SOCSE Sandip University, Nashik, Maharashtra, India
[5]Guide: Mr. Umesh Pawar (SOCSE, Sandip University, Nashik)

---***---

Abstract - *In Artificial Intelligence, automatically describing what's there in a photograph or image has always been a context of study. This paper includes the implementation of Automatic Caption Generator using CNN and RNN-LSTM models. It combines recent studies of machine translation as well as computer vision. The datasets used were Flickr8k. For evaluation of the performance of the described model we have used BLEU scores. Through the scores, one can apart the generated captions as good captions and bad captions. Main applications of this model include usage in virtual assistants, for image indexing, for social media, for visually impaired people, recommendations in editing applications and much more.*

Key Words: **CNN, RNN-LSTM, BLEU, VGG16, Deep Learning**

1.INTRODUCTION

Image Caption Generator models is based on encoder-decoder architecture which use input vectors for generating valid and appropriate captions. This model bridges gap between natural language processing as well as computer vision. It's a task of recognizing and interpreting the context described in the image and then describing everything in natural language such as English. Our model is developed using the two main models i.e., CNN (Convolutional Neural Network) and RNN-LSTM (Recurrent Neural Networks- Long Short-Term Memory). The encoder in the derived application is CNN which is used to extract the features from the photograph or image and RNN-LSTM works as a decoder that is used in organizing the words and generating captions. Some of the major applications of the application are self-driving cars wherein it could describe the scene around the car, secondly could be an aid to the people who are blind as it could guide them in every way by converting scene to caption and then to audio, CCTV cameras where the alarms could be raised if any malicious activity is observed while describing the scene, recommendations in editing, social media posts, and many more.

2. RELATED WORK

The application is merged with two main architectures CNN and RNN which describes attributes, relationships, objects in the image and puts into words.

CNN is an extractor that extracts features from the given image.

RNN- LSTM will be fed with the output of the CNN and following it will describe and generate a caption.

CNN is a Convolutional Neural Network which process the data having the input shape similar to two-dimensional matrix. CNN model has many layers including input layer, Convo Layer, Pooling Layer, Fully-connected layers, Softmax, and Output layers. Input layer in CNN is an image. Image data is presented in form of 3D form of matrix. Convo Layer also known as feature extractor where it performs the convolutional operation and calculate the dot products. ReLU is sub layer in Convo layer that converts all negative values to zero. Pooling layer is one where the volume of the image is being reduced once the convolution layer executes. Fully-Connected layers is connection layer that connects one neuron in a layer to other neuron in other layer involving neurons, biases and weights. Softmax layer is used for multi- classification of objects where using formula the objects are classified. Output layer is last layer at CNN model and has the encoded result to be fed to LSTM model.

RNN is Recurrent Neural Network where output of previous step is fed to ongoing step. LSTM (Long Short-Term Memory) is an extended version of RNN that are used to the predict the sequence based on the previous step where in it remembers all the steps and also the predicted sequence at every step. It grasps the required information from the processing of inputs as well as forget gate and also it does remove the non-required data

3. DATA FLOW DIAGRAM

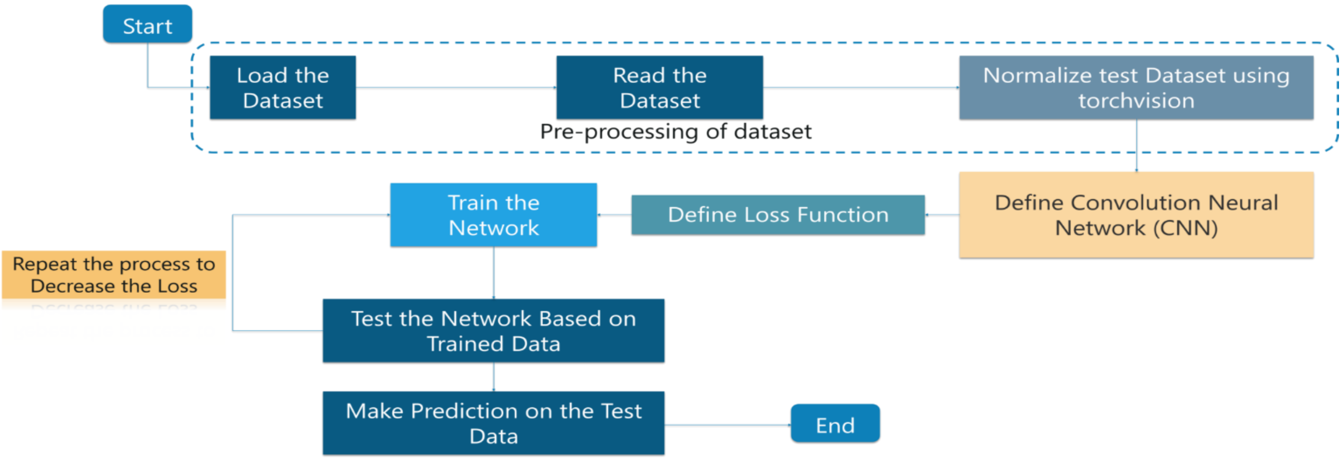

Fig -1: DFD Diagram

4. PROPOSED ARCHITECTURE

4.1 Three Phases of the application

- Features Extraction: The extraction of the features from images are being extracted. It creates vector features also known as embeddings. The CNN model extract features from original images after which they are compressed to smaller and RNN compatible feature vector. Thus, it's also known as Encoder.
- Tokenization: The next phase in the application is RNN, that decodes the feature vectors that were fed to it from CNN. Here the sequence of the words is predicted and however the captions are generated.
- Prediction: After the tokenization, the last step is Prediction. Here the vectors are decoded and the final output is being generated using get_prediction() function.

4.2 Flow of the project

- Importing the libraries
- Configuring the GPU memory to be used for training purposes
- Importing the image dataset and its respective captions
- Plotting few images and their captions from the dataset
- Cleaning captions for further analysis
- Cleaning the captions for further processing
- Plotting the top 50 words that appear in the cleaned dataset
- Loading VGG16 model and weights to extract features
- Extracting features
- Plotting similar images from the dataset
- Tokenizing the captions for further processing
- Processing the captions and images as per the requires shape by the model
- Building the LSTM model
- Training the LSTM model
- Plotting the loss value
- Generating captions
- Evaluating the performance using BLEU scores

5. LITERATURE SURVEY

Table -1:

No.	Authors	Research Paper	Publication Year	Dataset and methodology	Conclusions
1	Pranay Mathur, Aman Gill, Nand Kumar Bansode, Anurag Mishra	Camera2Caption: A real-time image caption generator	2017	Dataset: MS COCO Method: Advanced deep reinforcement learning based on NLP	The model proposed generates the real time environment high quality captions with the help of

				and Computer vision	tenserflow.
2	Simao Herdade, Armin Kappeler, Kofi Boakye, Joao Soares	Automatic Image Captioning using Convolution Neural Networks and LSTM	2019	Dataset: MS COCO Method: architecture model using CNN as well as NLP techniques	Using CNN and LSTM models the image's caption is generated.
3	Manish Raypurkar, Abhishek Supe, Pratik Bhumkar, Pravin Borse, Dr. Shabnam Sayyad	Deep learning-based image caption generator	2021	Dataset: Flickr_8k Method: CNN and LSTM model to extract features and sequence the words and finally generating captions.	Proposed model is based on multi label Neural networks
4	B.Krishnakumar,K.Kousalya, S.Gokul,R.Karthikeyan, D.Kaviyarasu	Image caption Generator using Deep Learning	2020	Method: Deep learning-based model using CNN to identify featured objects with the help of OpenCv.	Proposed model could generate captions successfully in Jupyter Notebook using keras as well as tenserflow
5	R. Subash	Automatic Image Captioning using Convolution Neural Networks and LSTM	2019	Dataset: MS COCO Method: NLP and CNN-LSTM based model	Using CNN-LSTM and NLP techniques the model for image captioning is generated

6. Datasets

- For the application of image caption generator, we used the dataset named Flickr_8k dataset.
- This dataset contains a wide range of images that has many different types of situations and scenes.
- Flickr_8k dataset has 8000 images and every image has 5 captions.
- We divided the entire dataset of 8000 images as 6000, 1000 and 1000 as training, validation and testing sets respectively
- Every image has different dimensions.

7. SYSTEM REQUIREMENTS

OS: Windows 7 and above, Recommended: Windows 10.

CPU: Intel processor with 64-bit support

Disk Storage: 8GB of free disk space.

For Execution: Anaconda Framework in Python.

For Deployment: Python

7.1 Libraries Used:

- Tensorflow: Its an open-source library that supports deep learning using Python etc frameworks.
- Keras: Its an open-source Python library that allows to evaluate the deep learning models.
- Pillow: Pillow is a Python Imaging Library (PIL), that adds support for opening, manipulating, and saving images.
- Numpy: To work with arrays, Numpy library is used.
- Matpolib: Library to create static and animated visuaizations in Python framework.

8. RESULTS

1	A	B	C	D	E	F	G	Output
2	DEEP LEARNING MODEL	ACTIVATION FUNCTION	COST FUNCTION	EPOCHS	GRADIENT ESTIMATION	NETWORK ARCHITECTURE	NETWORK INITIALIZATION	Mean BLEU score
3	**Gradient Estimation**							
4	1	ReLU	Cross-Entropy	5	Adam	3 layer, 256 nodes, LSTM, vgg16	default	0.37
5	2	ReLU	Cross-Entropy	6	Adam	3 layer, 256 nodes, LSTM, vgg16	default	0.351
6	3	ReLU	Cross-Entropy	5	Adagrad	3 layer, 256 nodes, LSTM, vgg16	default	0.404
7	4	ReLU	Cross-Entropy	5	RMSProp	3 layer, 256 nodes, LSTM, vgg16	default	0.374
8	5	ReLU	Cross-Entropy	5	Adadelta	3 layer, 256 nodes, LSTM, vgg16	default	0.353
9	6	ReLU	Cross-Entropy	5	Nadam	3 layer, 256 nodes, LSTM, vgg16	default	0.353
10	7	ReLU	Cross-Entropy	5	SGD	3 layer, 256 nodes, LSTM, vgg16	default	0.028
11	**Cost Function**							
12	1	ReLU	mean_squared_error	5	Adam	3 layer, 256 nodes, LSTM, vgg16	default	0.215
13	2	ReLU	hinge	5	Adam	3 layer, 256 nodes, LSTM, vgg16	default	0
14	3	ReLU	kullback_leibler_divergence	5	Adam	3 layer, 256 nodes, LSTM, vgg16	default	0.373
15	4	ReLU	cosine_proximity	5	Adam	3 layer, 256 nodes, LSTM, vgg16	default	0
16	**Network Initialization**							
17	1	ReLU	Cross-Entropy	5	Adam	3 layer, 256 nodes, LSTM, vgg16	glorot_uniform	0.381
18	2	ReLU	Cross-Entropy	5	Adam	3 layer, 256 nodes, LSTM, vgg16	random_uniform	0.388
19	3	ReLU	Cross-Entropy	5	Adam	3 layer, 256 nodes, LSTM, vgg16	lecun_uniform	0.367
20	4	ReLU	Cross-Entropy	5	Adam	3 layer, 256 nodes, LSTM, vgg16	he_uniform	0.389
21	5	ReLU	Cross-Entropy	5	Adam	3 layer, 256 nodes, LSTM, vgg16	glorot_normal	0.398
22	**Activation Function**							
23	1	ReLU	Cross-Entropy	5	Adam	3 layer, 256 nodes, LSTM, vgg16	default	0.374
24	2	tanh	Cross-Entropy	5	Adam	3 layer, 256 nodes, LSTM, vgg16	default	0.384
25	3	elu	Cross-Entropy	5	Adam	3 layer, 256 nodes, LSTM, vgg16	default	0.392
26	4	selu	Cross-Entropy	5	Adam	3 layer, 256 nodes, LSTM, vgg16	default	0.363
27	5	linear	Cross-Entropy	5	Adam	3 layer, 256 nodes, LSTM, vgg16	default	0.192
28	6	sigmoid	Cross-Entropy	5	Adam	3 layer, 256 nodes, LSTM, vgg16	default	0.375
29	7	softsign	Cross-Entropy	5	Adam	3 layer, 256 nodes, LSTM, vgg16	default	0.396
30	8	softplus	Cross-Entropy	5	Adam	3 layer, 256 nodes, LSTM, vgg16	default	0.381
31	**Epochs**							
32	1	ReLU	Cross-Entropy	3	Adam	3 layers, 256 nodes each	default	0.429
33	2	ReLU	Cross-Entropy	4	Adam	3 layers, 256 nodes each	default	0.394
34	3	ReLU	Cross-Entropy	5	Adam	3 layers, 256 nodes each	default	0.408
35	4	ReLU	Cross-Entropy	6	Adam	3 layers, 256 nodes each	default	0.38
36	5	ReLU	Cross-Entropy	7	Adam	3 layers, 256 nodes each	default	0.405
37	**Network Architecture**							
38	1	ReLU	Cross-Entropy	5	Adam	3 layers, 256 nodes each	default	0.407
39	2	ReLU	Cross-Entropy	5	Adam	3 layers, 128 nodes each	default	0.405
40	3	ReLU	Cross-Entropy	5	Adam	3 layers, 512 nodes each	default	0.394
41	4	ReLU	Cross-Entropy	5	Adam	4 layers, 256 nodes each	default	0.406
42	5	ReLU	Cross-Entropy	5	Adam	4 layers, 128 nodes each	default	0.386

Fig-3: Model results

Table: BELU Scores

No.	Research Works i.e., Models	BELU Scores
1.	LRCN	0.669
2.	NIC	0.277
3.	VSA	0.584
4.	CNN-LSTM	0.681
5.	Our Model	0.398

8.1 Some good and bad captions

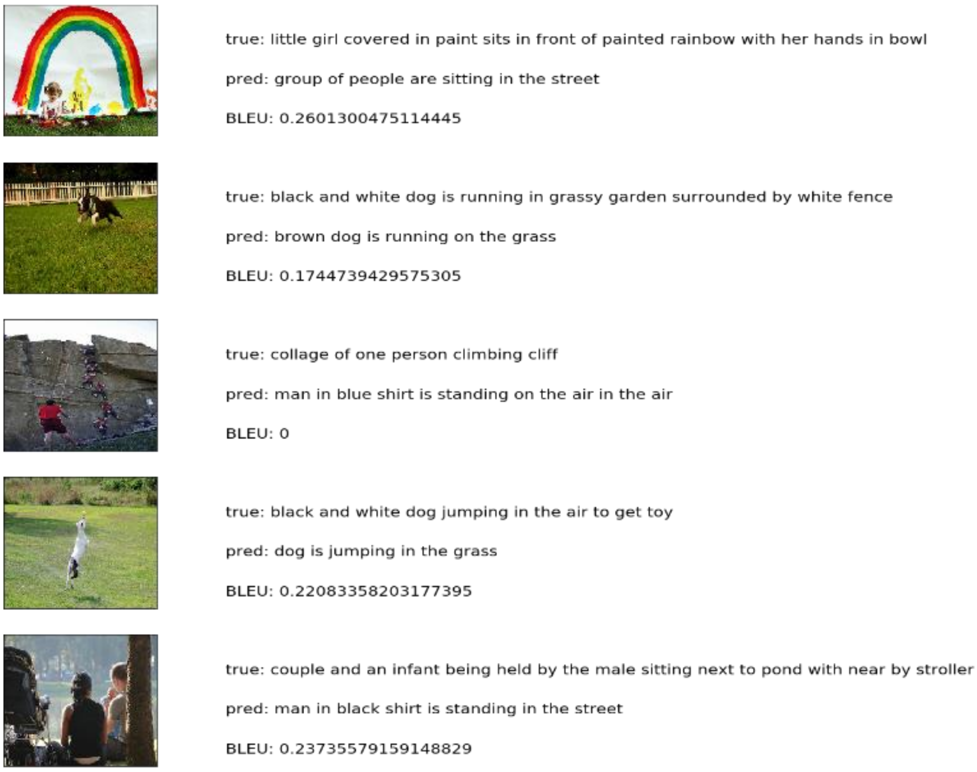

Fig-5: Bad Captions

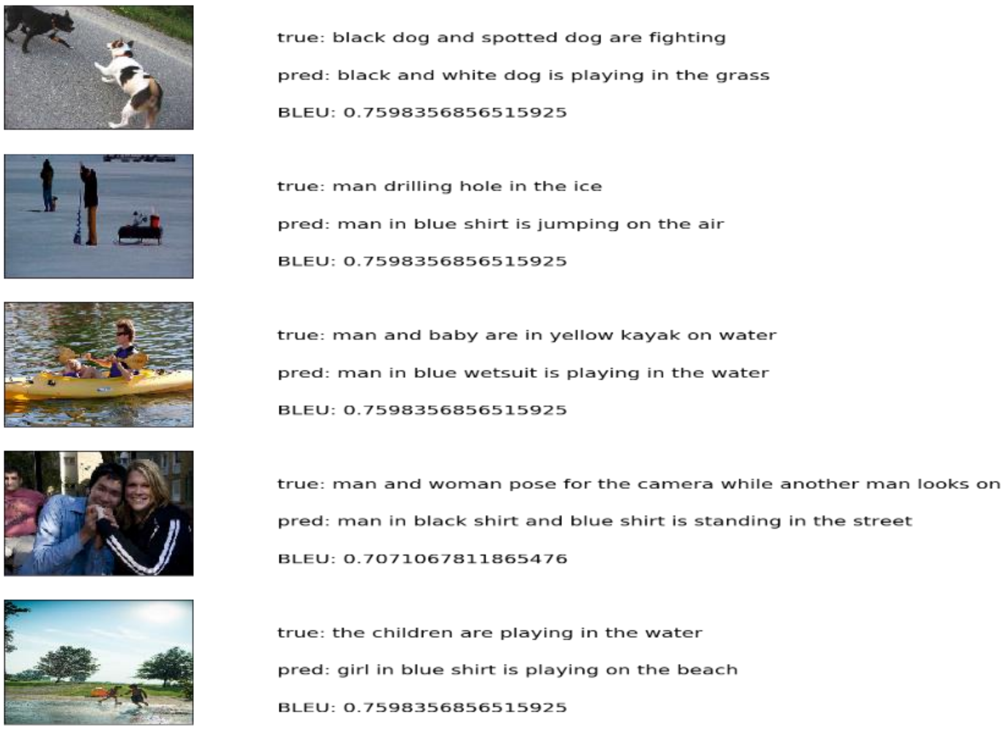

Fig-6: Good Captions

9. CONCLUSION AND FUTURE WORK

The model has been successfully trained and tested to generate the valid captions for the loaded images.

The proposed model is based on multi label classification that uses CNN-RNN approach to generate the captions where CNN works as an encoder and RNN works as a decoder. The CNN architecture used is VGG16.

Some of the future enhancements would include describing the captions based on on multiple targets. The generated caption should be in variety of languages. Training and testing the model with larger datasets and on different architectures with different CNN architectures such as LeNet, AlexNet, GoogLeNet, ResNet and more. Also, the values generated of BeLU i.e., BeLU scores must be high in order to achieve the maximum accuracy of the model.

10. REFERENCES

[1] R. Subash November 2019: Automatic Image Captioning Using Convolution Neural Networks and LSTM.

[2] Seung-Ho Han, Ho-Jin Choi (2020): Domain-Specific Image Caption Generator with Semantic Ontology.

[3] Pranay Mathur, Aman Gill, Aayush Yadav, Anurag Mishra and Nand Kumar Bansode (2017): Camera2Caption: A Real-Time Image Caption Generator

[4] Simao Herdade, Armin Kappeler, Kofi Boakye, Joao Soares (june 2019): Image Captioning: Transforming Objects intoWords.

[5] Manish Raypurkar, Abhishek Supe, Pratik Bhumkar, Pravin Borse, Dr. Shabnam Sayyad (March 2021): Deep learning-based Image Caption Generator

[6] Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan (2015): Show and Tell: A Neural Image Caption Generator

[7] Jianhui Chen, Wenqiang Dong, Minchen Li (2015): Image Caption Generator based on Deep Neural Networks