IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁒ Impact Factor 7.39 ⁒ Vol. 11, Issue 5, May 2022

DOI: 10.17148/IJARCCE.2022.115208

Image Captioning and Fact Generation

Aniruddh T S[1], Joshua A[2], Mukesh Kanna V[3], Vishnu S S[4], Dr. Tamilselvi P[5]

[1-5]School of Computer Science and Technology, Faculty of Engineering and Technology,

Jain (Deemed to be University), Bangalore, India.

Abstract: The use of machines to perform different tasks is constantly increasing in society. Providing machines with perception can lead them to perform a great variety of tasks; even very complex ones such as elderly care. Machine perception requires that machines understand their environment and the interlocutor's intention.Thus, deep learning has the potential to improve human-machine interaction because its ability to learn features will allow machines to develop perception. And by having perception, machines will potentially provide smoother responses, drastically improving the user experience.

The process of creating a textual explanation for a set of photos is known as image captioning. In the Deep Learning arena, it has been a critical and basic endeavor. Image captioning has a wide range of uses. Image captioning is a popular research field in Artificial Intelligence as it combines the 2 major fields in Artificial Intelligence i.e., Deep Learning and Natural Language Processing. This paper presents a model that combines Natural Language Processing modules (Glove Embedding and LSTM) and Deep Learning (Feature extraction from images) to generate a sentence describing an image. The model is combined with a function that generates facts based on the primary feature in the image. Given the training image, the model is trained to maximize the likelihood of the target description sentence. Also, this has been deployed using streamlit, hosted on the web.

Keywords: Deep Learning, Artificial Intelligence, Natural Language Processing, Image Captioning, Streamlit.

INTRODUCTION

Image Captioning and Facts Generation as the name suggest is a generative model which tries to generate a caption according to the image that is fed into the system with respect to the subject(s) in the image and fact generation is also more or less the same but generates facts according to the subject of the image. This is achieved using Deep Learning in particular Computer vision and Natural Language Processing. Computer Vision helps in understanding the context of the image and NLP helps in generating the caption with respect to the context of the image.The caption can describe the objects, attributes and relationships of the entities in the image. To make the total system unique we have added a fact generation module that generates facts describing the primary feature in the image. For Computer vision, we have used InceptionV3 and also the VGG model to extract the image features and LSTM for NLP to generate texts for the image. We have also used custom-created intents files to generate facts based on the subject of the image. Image Captioning is also incorporated into various technologies as an accessibility feature for the blind so that the image or scenery can be explained via Text-to-Speech.

In our day-to-day life, Data becomes more and more on the internet and other social media sites. We need to analyze the data and make use of it .these data contains a lot of images where most of the image does not contain any description or an explanation of the image .our deep learning model can able to recognize and analyze most of the features in the image so that we can find the meaning of the image.The image can be described using natural sentences and the features in the image. But the challenge begins in the relationship between the object of the image. This is where image captioning comes into place. To do this we must have a large amount of dataset where we have used the Flickr8k dataset.

LITERATURE SURVEY

- In the paper [5] entitled Image captioning and visual question answering based on attributes and external knowledge. They have developed a CNN and RNN-based architecture where it retrieves the caption from the image and uses the main objects to ask the question about the image For attribute retrieval they have used computer vision like object recognition, image annotation and image retrieval For image captioning they have used computer vision and natural language processing for building the framework as CNN and RNN. They have also implemented a baseline of the VGGNET and LSTM model for evaluation purposes by calculating the BLUE, METEOR and CIDEr values.

DOI: 10.17148/IJARCCE.2022.115208

- In the paper [7] entitled CNN+CNN Convolutional decoders for image captioning, they have used CNN+CNN architecture as the name suggests. Instead of using CNN+RNN, they have used CNN+CNN which they have compared with lstm-based models. They compared two models with the values of BLUE, METEOR and CIDEr values. They have found out that using CNN+CNN is giving comparable values in BLUE and METEOR, whereas it gives higher values in CIDEr values. They have also compared the values of the different datasets whether it affects them or not by using the MS COCO dataset and Flickr 30k.
- Jyoti Aneja and team in their paper [6] entitled “Convolutional image captioning” have used MS COCO as their data set and have used BLEU, METEOR, ROUGE and CIDEr for Evaluation Metrics. They have used VGGNet as their Image Encoder and CNN for the model building.
- In the paper[3] “Captioning images with diverse objects” have used MS COCO and ImageNet for the dataset and have used VGGNet for image encoding. Here they have used LSTM for the model and BLEU, METEOR, ROUGE and CIDEr for evaluation metrics.
- Li Zhang in their paper [4] entitled “Actor-critic sequence training for image captioning” have developed a model on actor-critic reinforcement learning in which they have to use MS COCO as dataset and BLEU, METEOR, ROUGE and CIDEr for evaluating the model which is build using LSTM and inception-V3 as image encoder.
- In the paper [2]“Self-critical sequence training for image captioning” by Steven, he developed a model using LSTM and used ResNet for encoding. In this model, they have made use of MS COCO as a dataset and several evaluation metrics such as BLEU, METEOR, CIDEr and ROUGE.
- In the paper [1]“Incorporating copying mechanism in image captioning for learning novel objects” have used both MS COCO and ImageNet as datasets for their LSTM model which uses VGGNet as their image encoder and uses METEOR as evaluation metrics.

All the studies which were performed or done have their limitations and drawbacks. Those studies are limited to only performing image captioning alone with no other additional options of facts generation and other. From the previous papers, we have seen most of them mostly compared with the LSTM baseline model. Their model can also be compared with the other models also.

SYSTEM DESCRIPTION

Data Organization

Flickr8k dataset was used for training the model. Flicker8k is one of the most widely used open-source datasets for training models based on images and their descriptions. The flicker8k contains 8000 images while its variant Flickr30k has 30000 images. Flickr 8k is preferred as the size is significantly smaller to store and use. Each image has 5 descriptions/captions respectively. 75% of the images were used for training and 25% were used for testing the model. A custom-made CSV file was used for storing the facts.

System Architecture

The proposed system architecture consists of 2 deep learning models, one to extract features from the input image and the other to generate captions. The user inputs the image, the image is then pre-processed and sent to the InceptionV3/VGG16 model for feature extraction. The extracted feature is sent to the captioning model to generate the captions. Based on the captions generated a set of facts related to the main subject of the caption is generated from a dataset of pre-defined facts. The user interface is developed using Streamlit and deployed in Streamlit Cloud. Figure 3.2 shows the system architecture.

DOI: 10.17148/IJARCCE.2022.115208

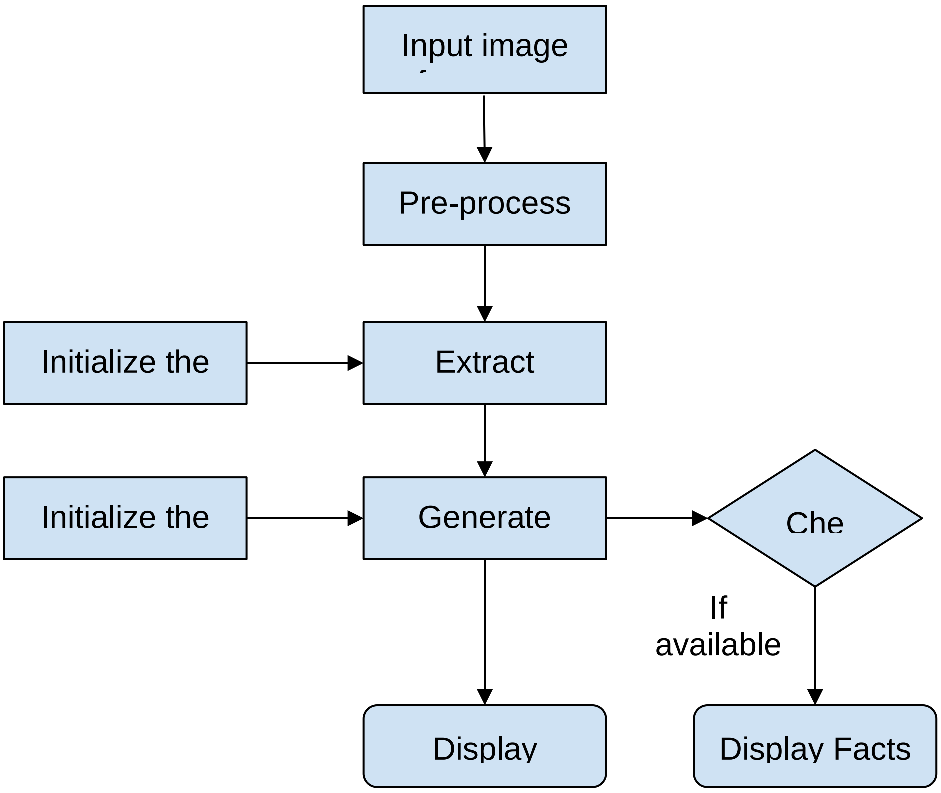


Figure 1.1: system architecture

PERFORMANCE EVALUATION

The captioning model was evaluated using the Bilingual Evaluation Understudy (BLEU) Score. Captioning Model's BLEU score:

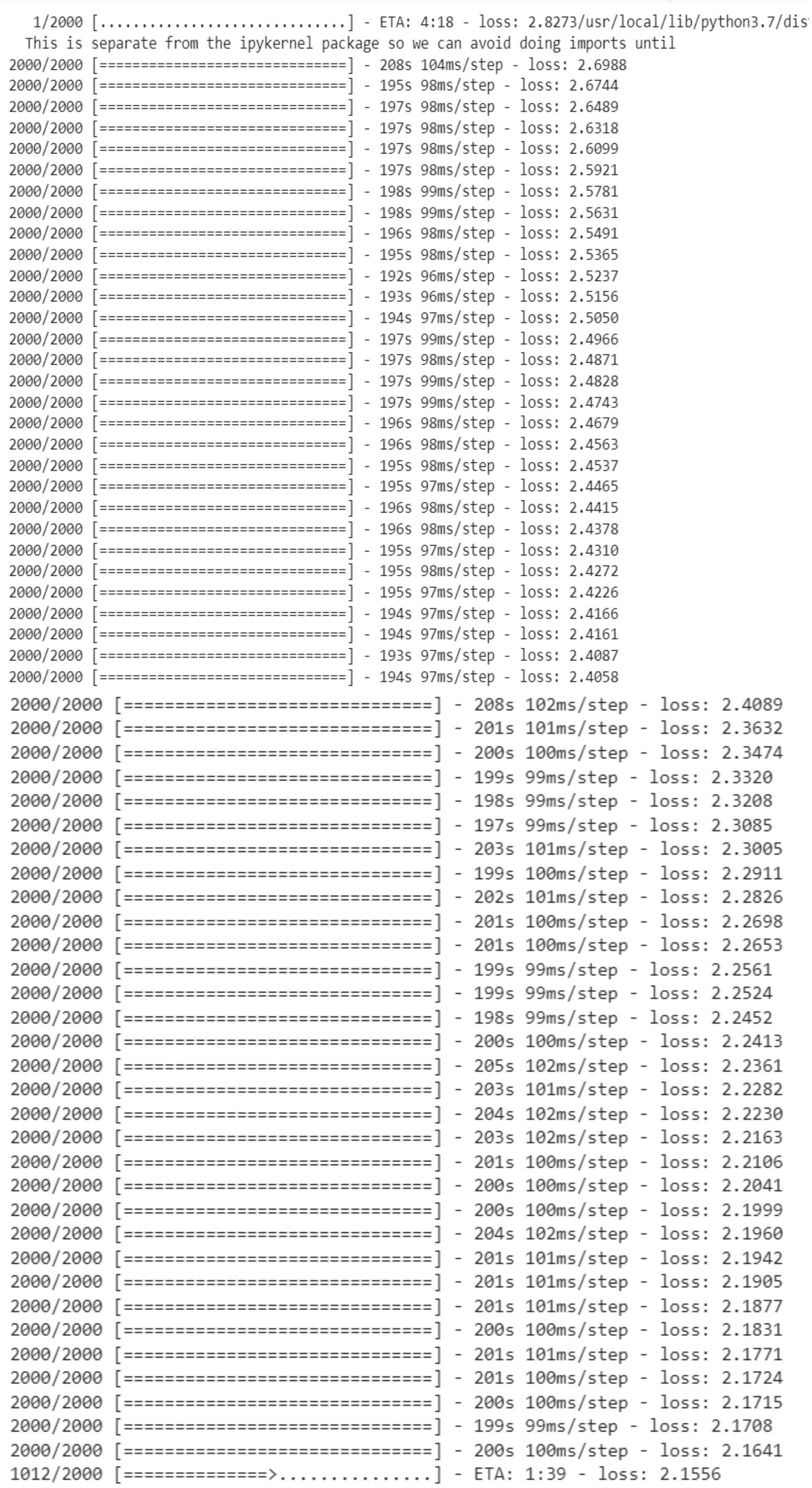
BLEU-n	**n-Gram**	**Score (out of 1)**
BLEU 1	1-gram	0.74206
BLEU 2	2-gram	0.70333
BLEU 3	3-gram	0.65291
BLEU 4	4-gram	0.61814

Table 1.1: Model's BLEU Score

Generally BLEU score, a evaluation metric which is used for evaluating the quality of the language when it's been translated by machine (i.e) Machine Translation. Here we use it to check the quality of the caption which has been generated by the model. BLEU score calculates the similarity of the actual sentence with the predicted sentence. But the BLEU score cannot capture the semantically similar sentence. Therefore, the real performance of the model can be analyzed only by using the model.

DOI: 10.17148/IJARCCE.2022.115208

RESULT AND DISCUSSION



```
   1/2000 [..............................] - ETA: 4:18 - loss: 2.8273/usr/local/lib/python3.7/dis
  This is separate from the ipykernel package so we can avoid doing imports until
2000/2000 [==============================] - 208s 104ms/step - loss: 2.6988
2000/2000 [==============================] - 195s 98ms/step - loss: 2.6744
2000/2000 [==============================] - 197s 98ms/step - loss: 2.6489
2000/2000 [==============================] - 197s 98ms/step - loss: 2.6318
2000/2000 [==============================] - 197s 98ms/step - loss: 2.6099
2000/2000 [==============================] - 197s 98ms/step - loss: 2.5921
2000/2000 [==============================] - 198s 99ms/step - loss: 2.5781
2000/2000 [==============================] - 198s 99ms/step - loss: 2.5631
2000/2000 [==============================] - 196s 98ms/step - loss: 2.5491
2000/2000 [==============================] - 195s 98ms/step - loss: 2.5365
2000/2000 [==============================] - 192s 96ms/step - loss: 2.5237
2000/2000 [==============================] - 193s 96ms/step - loss: 2.5156
2000/2000 [==============================] - 194s 97ms/step - loss: 2.5050
2000/2000 [==============================] - 197s 99ms/step - loss: 2.4966
2000/2000 [==============================] - 197s 98ms/step - loss: 2.4871
2000/2000 [==============================] - 197s 99ms/step - loss: 2.4828
2000/2000 [==============================] - 197s 99ms/step - loss: 2.4743
2000/2000 [==============================] - 196s 98ms/step - loss: 2.4679
2000/2000 [==============================] - 196s 98ms/step - loss: 2.4563
2000/2000 [==============================] - 195s 98ms/step - loss: 2.4537
2000/2000 [==============================] - 195s 97ms/step - loss: 2.4465
2000/2000 [==============================] - 196s 98ms/step - loss: 2.4415
2000/2000 [==============================] - 196s 98ms/step - loss: 2.4378
2000/2000 [==============================] - 195s 97ms/step - loss: 2.4310
2000/2000 [==============================] - 195s 98ms/step - loss: 2.4272
2000/2000 [==============================] - 195s 97ms/step - loss: 2.4226
2000/2000 [==============================] - 194s 97ms/step - loss: 2.4166
2000/2000 [==============================] - 194s 97ms/step - loss: 2.4161
2000/2000 [==============================] - 193s 97ms/step - loss: 2.4087
2000/2000 [==============================] - 194s 97ms/step - loss: 2.4058
2000/2000 [==============================] - 208s 102ms/step - loss: 2.4089
2000/2000 [==============================] - 201s 101ms/step - loss: 2.3632
2000/2000 [==============================] - 200s 100ms/step - loss: 2.3474
2000/2000 [==============================] - 199s 99ms/step - loss: 2.3320
2000/2000 [==============================] - 198s 99ms/step - loss: 2.3208
2000/2000 [==============================] - 197s 99ms/step - loss: 2.3085
2000/2000 [==============================] - 203s 101ms/step - loss: 2.3005
2000/2000 [==============================] - 199s 100ms/step - loss: 2.2911
2000/2000 [==============================] - 202s 101ms/step - loss: 2.2826
2000/2000 [==============================] - 201s 100ms/step - loss: 2.2698
2000/2000 [==============================] - 201s 100ms/step - loss: 2.2653
2000/2000 [==============================] - 199s 99ms/step - loss: 2.2561
2000/2000 [==============================] - 199s 99ms/step - loss: 2.2524
2000/2000 [==============================] - 198s 99ms/step - loss: 2.2452
2000/2000 [==============================] - 200s 100ms/step - loss: 2.2413
2000/2000 [==============================] - 205s 102ms/step - loss: 2.2361
2000/2000 [==============================] - 203s 101ms/step - loss: 2.2282
2000/2000 [==============================] - 204s 102ms/step - loss: 2.2230
2000/2000 [==============================] - 203s 102ms/step - loss: 2.2163
2000/2000 [==============================] - 201s 100ms/step - loss: 2.2106
2000/2000 [==============================] - 200s 100ms/step - loss: 2.2041
2000/2000 [==============================] - 200s 100ms/step - loss: 2.1999
2000/2000 [==============================] - 204s 102ms/step - loss: 2.1960
2000/2000 [==============================] - 201s 101ms/step - loss: 2.1942
2000/2000 [==============================] - 201s 101ms/step - loss: 2.1905
2000/2000 [==============================] - 201s 101ms/step - loss: 2.1877
2000/2000 [==============================] - 200s 100ms/step - loss: 2.1831
2000/2000 [==============================] - 201s 101ms/step - loss: 2.1771
2000/2000 [==============================] - 201s 100ms/step - loss: 2.1724
2000/2000 [==============================] - 200s 100ms/step - loss: 2.1715
2000/2000 [==============================] - 199s 99ms/step - loss: 2.1708
2000/2000 [==============================] - 200s 100ms/step - loss: 2.1641
1012/2000 [==============>...............] - ETA: 1:39 - loss: 2.1556
```

Figure 1.2: Training Result

Figure 1.2 shows the training of the model which was done for about 80 epochs, a consistent thing which was noted throughout the training process was that the loss kept on decreasing per epoch which shows that the model is consistent and testing the model through random pictures from the internet also showed good results which has been included below in Table 1.1 , but if the model could be trained for few more epochs the model would get better and better.

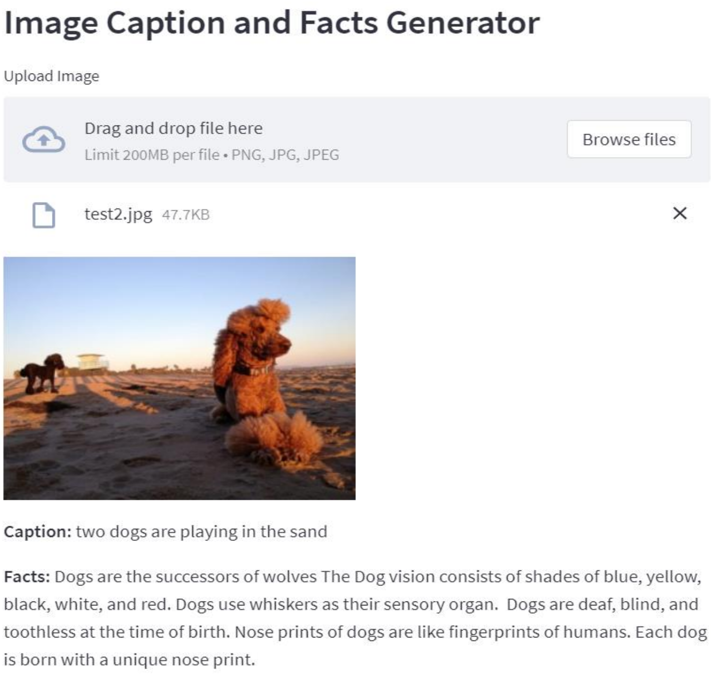


Image Caption and Facts Generator

Caption: two dogs are playing in the sand

Facts: Dogs are the successors of wolves The Dog vision consists of shades of blue, yellow, black, white, and red. Dogs use whiskers as their sensory organ. Dogs are deaf, blind, and toothless at the time of birth. Nose prints of dogs are like fingerprints of humans. Each dog is born with a unique nose print.

Figure 1.3:Caption and fact generated by the model for an image

Figure 1.3 shows the captions and facts generated by the model for an image. The features from the images are extracted by the InceptionV3 model based on which the captioning model generates the caption. The caption is generated using greedy search method. The generated captions describe the images based on the features present on the images. The captions generated also perfectly match with the primary subject of the image.

CONCLUSION

On examples of this problem, deep learning approaches have recently obtained state-of-the-art results. Deep learning models have been shown to be capable of achieving optimal outcomes in the realm of caption generating challenges. A single end-to-end model can be defined to predict a caption given a photo, rather than requiring sophisticated data preparation or a pipeline of separately designed models. To assess our model, we use the BLEU standard metric to assess its performance on the Flickr8K dataset. These findings demonstrate that our suggested model outperforms traditional models in terms of picture captioning in performance evaluation.

We present a working model of Image Captioning and Facts Generation which has been deployed as a web app using streamlit. The model identifies the subject in the image and provides captions related to it and also provides facts that have been identified as the subject by the model.

REFERENCES

[1] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. 2017. Incorporating copying mechanism in image captioning for learning novel objects. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 5263–5271.

[2] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR).1179–1195.

[3] Subhashini Venugopalan, Lisa Anne Hendricks, Marcus Rohrbach, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2017. Captioning images with diverse objects. In Proceedings of the IEEE conference on computer vision and pattern recognition. 1170–1178.

DOI: 10.17148/IJARCCE.2022.115208

[4] Li Zhang, Flood Sung, Feng Liu, Tao Xiang, Shaogang Gong, Yongxin Yang, and Timothy M Hospedales. 2017. Actor-critic sequence training for image captioning. arXiv preprint arXiv:1706.09601.

[5] Qi Wu, Chunhua Shen, Peng Wang, Anthony Dick, and Anton van den Hengel. 2018. Image captioning and visual question answering based on attributes and external knowledge. IEEE transactions on pattern analysis and machine intelligence 40, 6, 1367–1381.

[6] Jyoti Aneja, Aditya Deshpande, and Alexander G Schwing. 2018. Convolutional image captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 5561–5570.

[7] Qingzhong Wang and Antoni B Chan. 2018. CNN+ CNN: Convolutional Decoders for Image Captioning. arXiv preprint arXiv:1805.09019.