iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** IV **Month of publication:** April 2022

DOI: https://doi.org/10.22214/ijraset.2022.41309

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue IV Apr 2022- Available at www.ijraset.com

Generate Detailed Captions of an Image using Deep Learning

Khan Shayaan Shakeel[1], Masalawala Murtaza Shabbir[2], Qazi Faizan Ahmed[3], Nafisa Mapari[4]
[1, 2, 3, 4] *Computer Engineering, M. H. Saboo. Siddik COE, Mumbai, India*

Abstract: This paper shows the implementation of image caption generation using deep learning algorithm. The project is one of the primary example of computer vision. The main aim of computer vision is scene undertanding. The algorithms used in this model are CNN and LSTM. This model is an extension of the model based on CNN - RNN Model which suffers from the drawback of vanishing gradient. Xception model is used for image feature extraction and is a CNN model that is trained using ImageNet dataset. Extracted features from the Xception model is fed as the input to the LSTM model which in turn generates the caption for the image. The dataset used for training and testing is Flickr_8k dataset.

I. INTRODUCTION

For humans the process of describing an image is a very simple process but for computers we achieve this with the help of computer vision. The main goal of computer is to understand the scenario in an image. Not only should it understand the image but also it should be able to express it in the human language. Image captioning is a process where the system must be capable enough to distinguish between the different objects and then later express it in the terms of language which is understood by humans. We create a system that links the objects in the image and creates a logical sequence. This logical sequence of description comes with the help of learning the data and the dataset consists of images along with descriptions that help us to train our model and predict the results.

"man in black shirt is playing guitar."

Figure 1: Example of man playing guitar

In the above diagram we can see that the caption generated is very accurate. The general idea is to divide the system into logically 2 modules where the first module is an Image based model and the other is a Language based model.

Image based model is built with the help of Convolutional Neural Network (CNN). This model is used to extract the features from the image and identifies the different segments of an image and assigns weight to it, which helps in the classification of the image. CNN is found to be very useful in image classification however our main goal is to extract the features. CNN is generally used in layers where the output of the first layer is fed as the input to the second layer and so on. After a series of layers we get the vectorial representation of image which is fed as an input to the language based model.

Language based model is built with the help of a Long Short Term Memory Network (LSTM). LSTM is a type of Recurrent Neural Network and is generally is used in sequence prediction problems. RNN can also be used for sequence prediction but the limitation with RNN is short term memory, as a result LSTM is found to be more efficient for predicting sequence.

II. RELATED WORK

This section gives detailed information about the research work that has been done on Image Caption Generation. Recently the quality of image caption generation has improved considerably by using combinations of CNN to obtain vectorial representation of images and RNN to decode those representations into natural language sentences.

Yao et al have published a research paper that explains the process of Image and Video to Text conversion. The entire process is based on Image Comprehension.

The process is divided into three steps. In the first step visual features are extracted. In the second step the output of the first stage is given as input to second stage which converts it into textual description. In the final stage the description is transformed into semantically meaningful, human understandable captions. Users can not only obtain captions for images but for videos as well.

Li et al have published a paper that incorporates storytelling for videos. The main aim is to produce coherent and concise stories for long videos.

With the help of the Multimodal Embedding Research, they have designed a Residual Bidirectional RNN to use past and future contextual knowledge. Multimodal embedding is also used for video clip phrases.

O. Vinyals et al have developed a model known as NIC which is a end-to-end neural network model that automatically generates caption for the input image [4]. The entire model is dependent on CNN which is used for features extraction and then later it is trained by a RNN to generate sentences.

This system has proved to be producing accurate results for larger datasets. The model quantitative evaluations is done either by using BLEU or ranking metrics to assess the generated descriptions.

S. Shabir, S. Arafat et al have published that since there are many research is going on to find new ways for generating captions, they have given detailed overview over technical aspects and techniques of image captioning. The research paper is all about the most common process for image captioning to new ways that have been discovered. The research paper also talks about the all related points in detail. The paper has even proposed the fields where the potential efforts should be made in order to improve the results.

Hao Fang et al have published a system that divides the process of image caption generation into three major steps. First the system reasons with the image sub-regions rather than the entire image. Next with the help of the CNN the features from the sub-regions are extracted and then fine-tuned on the training data.

The training is done at Maximum Entropy (ME) from training data set descriptions. This training results in capturing of commonsense knowledge about the image through language statistics. The final stage is re-ranking of a set of high-likelihood sentences by a linear weighting.

These weights are assigned on the basis of Minimum Error Rate Training (MERT). In addition to this they have used Deep Multimodal Similarity Model (DMSM) that maps the similarity between text and image. This in turn improves the selection of quality captions.

Kelvin Xu el at have proposed a system with two approaches. The first one is soft deterministic attention mechanism that is trained on the basis of standard back-propagation methods and the second one is hard deterministic attention mechanism which is trained by maximizing an approximate variational lower bound or by REINFORCE. The paper showcase the how we can gain insights and interprets the results. It visualize where and what the attention is focused on in an image. The paper also show the usefulness of the caption generated by evaluating it against state of art performance.

III. METHODOLOGY

A. System Design

The entire module can be logically divided into two modules:

1) Image Based Model
2) Language Based model

In the Image Based Model the input image is converted into vectorial representations. For image based model convolutional neural networks are used in combinations and is also known as the Encoder. In the Language Based model the vectorial representations are converted into natural language. The vectorial representations are decoded with the help of the LSTM network and this model is also known as the Decoder.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue IV Apr 2022- Available at www.ijraset.com

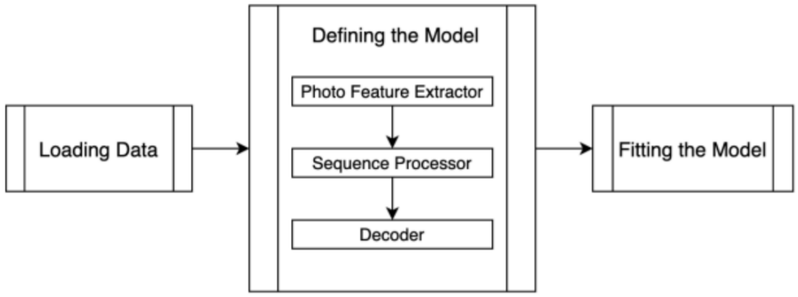


Figure 2: Model Structure

Both the models are integrated together to complete the entire process of image to text generation and later with the help of pyttsx3 or gtts we can convert the generated caption to speech, which will help the visually impaired people.

The process is divided into following logical steps:
Step1: Firstly after pre-processing the data, the data is loaded into the model for training
Step2: This step can be further bifurcated into three more steps:

- Photo Feature Extractor: In this module, it will extract the features from the image by using different combinations of convolutional neural networks.
- Sequence Processor: The output from the Photo Feature Extractor is then fed into the sequence processor and this uses Long Short Term Memory Network (LSTM) for managing texts.
- Decoder: It produces the most logically correct sequence by combining sequence processor and photo feature extractor.

B. Data Collection

The dataset used for training and testing model is flickr_8k dataset and is divided into 6000 images for training, 1000 images for validation and 1000 images for testing. The dataset consist of two directories:
1) Flickr8k_dataset: It consists of 8092 photographs in JPEG format.
2) Flickr8k_text: It consists of number of files having descriptions for the image.

C. Convolutional Neural Network (CNN)

A Convolutional layer is also known as CNN or CoonvNet and consist of three layers viz. convolutional layer, pooling layer and fully connected layer.
1) *Convolutional Layer:* All the load of computational work is handled by the convolutional layer. This layer performs dot product between two matrices, where one matrix represents the kernel i.e. learnable parameters and the other matrix represents restricted portion of the receptive field. The kernel is smaller than an image but it is more in depth. The Kernel slides across the height and width of the image, producing a two dimensional representation of the image.
2) *Pooling Layer:* The pooling layer is used to derive summary statistics of nearby outputs at certain locations. This results in reducing the spatial size of representation which in turn reduces the computation and weights. Every slice of the representation has its own pooling operation. Several pooling opeartions are there such as the average of the rectangular neighbourhood, L2 norm of the neighbourhood, weighted average based on the distance from the central pixel, but the most commonly used is the Max pooling, where the maximum of the neighbourhood is taken into consideration.
3) *Fully Connected Layer:* This layer helps to map the input and output of all the layers. The Neurons in this layer are fully connected with all the neurons in the preceding and succeeding layers. CNN is a Deep Learning Algorithm and uses the concept of weights for image classification. CNN assign weights to different objects present in the image, which helps in the classification of image. For vectorial representation of image, layers of CNN are used together. The output of first layer is fed as an input to the second layer and this process continues for all the subsequent layers. After a series of convolutional network, it is necessary to connect a fully connected layer and which result in a N dimensional vector which is in the encoded form.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue IV Apr 2022- Available at www.ijraset.com

D. Long Short Term Memory (LSTM)

LSTM is a type of a Recurrent Neural network. RNN and LSTM are generally used for predicting orders. The idea behind using LSTM is that when we go into deep neural networks, if the gradients are very low or zero then the training cannot take place which eventually leads to poor prediction performance.

Long Short Term Memory is an advanced RNN algorithm which overcomes the limitations of traditional RNN. RNN remembers the past information and uses it for it's current operation, but due to the short term memory (also known as vanishing gradient) it cannot remember long term dependencies. LSTM overcomes the limitations of the traditional RNN and proves to be more efficient in long term sequence prediction.

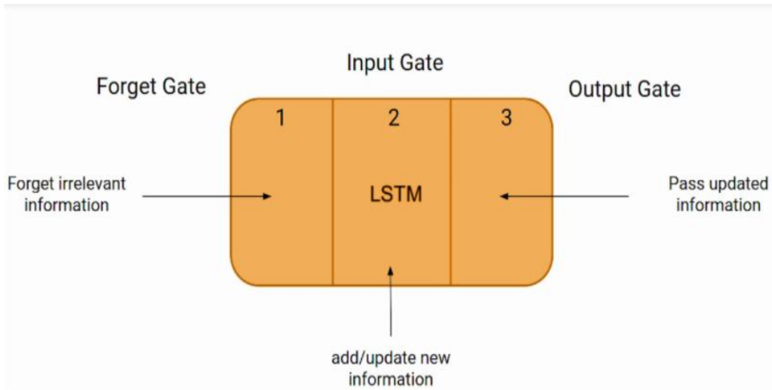


Figure 3: LSTM Gates

LSTM consists of three parts. The first part tells whether the information coming from the previous timestamp is relevant or not and if it is not then the information is discarded. The second part tries to learn new information from the input provided to it and finally the third part that helps in updating the information from current timestamp to the next timestamp. Depending upon the functionality of all the three parts they are known as the Forget gate, the Input gate and the Output gate respectively.

IV. IMPLEMENTATION

1) *Data Preprocessing and Cleaning:* This process starts by loading the text file for data cleaning. The motive behind data cleaning is to ensure that our text doesn't contains any punctuation marks, converting the entire text into lowercase and removing words that contains numbers or taking care of stop words.

Figure 4: Data Cleaning

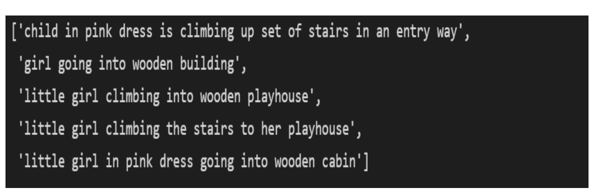


Figure 5: Example of clean descriptions

2) *Creating Vocabulary for the Image:* Machines cannot handle raw text. First the cleaning of the text is important which is done by splitting it into words, handling punctuations and removing words with the numbers. Each unique word is mapped to a unique index value which could be understood by the machines.

```
#  All words in description dictionary
all_vocab =  []

for key in descriptions.keys():
    [all_vocab.append(i) for des in descriptions[key] for i in des.split()]

print('Vocabulary Size: %d' % len(all_vocab))
print(all_vocab[:15])

Vocabulary Size: 373837
['child', 'in', 'pink', 'dress', 'is', 'climbing', 'up', 'set', 'of', 'stairs', 'in', 'an', 'entry', 'way', 'girl']
```

Figure 6: Vocabulary Creation

3) *Data Generator:* For this supervised model 6000 input images are provided and each image has 4096 length feature vector. This amount of large data cannot be stored in the memory, so we use a generator that yields batches.

```
def data_generator(train_descriptions,encoding_train,word_to_idx,max_len,batch_size):
    X1,X2, y = [],[],[]

    n =0
    while True:
        for key,desc_list in train_descriptions.items():
            n += 1

            photo = encoding_train[key+".jpg"]
            for desc in desc_list:

                seq = [word_to_idx[word] for word in desc.split() if word in word_to_idx]
                for i in range(1,len(seq)):
                    xi = seq[0:i]
                    yi = seq[i]

                    #0 denote padding word
                    xi = pad_sequences([xi],maxlen=max_len,value=0,padding='post')[0]
                    yi = to_categorcial([yi],num_classes=vocab_size)[0]

                    X1.append(photo)
                    X2.append(xi)
                    y.append(yi)

                if n==batch_size:
                    yield [[np.array(X1),np.array(X2)],np.array(y)]
                    X1,X2,y = [],[],[]
                    n = 0
```

Figure 7: Data Generator

4) *CNN-LSTM Model:* With the help of CNN-LSTM we generate the captions. The Structure of the model is built with the help of keras library that consist of three vital components. These are as follows:

a) Photo Feature Extractor: This module will extract the features from the image by using different combinations of convolutional neural networks.
b) Sequence Processor: The output from the Photo Feature Extractor is fed to the sequence processor. It uses Long Short Term Memory Network (LSTM) for managing text.
c) Decoder: It produces the most logically correct sequence by combining sequence processor and photo feature extractor.

```
model.summary()

Output exceeds the size limit. Open the full output data in a text editor
Model: "model_1"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_3 (InputLayer)           [(None, 35)]         0           []

 input_2 (InputLayer)           [(None, 2048)]       0           []

 embedding (Embedding)          (None, 35, 50)       92400       ['input_3[0][0]']

 dropout (Dropout)              (None, 2048)         0           ['input_2[0][0]']

 dropout_1 (Dropout)            (None, 35, 50)       0           ['embedding[0][0]']

 dense (Dense)                  (None, 256)          524544      ['dropout[0][0]']

 lstm (LSTM)                    (None, 256)          314368      ['dropout_1[0][0]']

 add (Add)                      (None, 256)          0           ['dense[0][0]',
                                                                  'lstm[0][0]']

 dense_1 (Dense)                (None, 256)          65792       ['add[0][0]']

 dense_2 (Dense)                (None, 1848)         474936      ['dense_1[0][0]']

...
Total params: 1,472,040
Trainable params: 1,472,040
Non-trainable params: 0
```

Figure 8: Model Summary

5) *Training the Model:* The flickr_8k dataset consists of 6000 images in jpeg format for training. Each image has 5 descriptive captions.

Train Our Model

```
epochs = 20
batch_size = 3
steps = len(train_descriptions)//64
```

```
def train():

    for i in range(epochs):
        generator = data_generator(train_descriptions,encoding_train,word_to_idx,max_len,batch_size)
        model.fit_generator(generator,epochs=1,steps_per_epoch=steps,verbose=1)
        model.save('model_weights/model_'+str(i)+'.h5')
```

```
model = load_model("./model_weights/model_9.h5")
```

Figure 9: Training the Model

6) *Testing the Model:* After the model is trained, we test the model against random images and evaluate the generated captions.

Predictions

```
def predict_caption(photo):
    in_text = "startseq"

    for i in range(max_len):
        sequence = [word_to_idx[w] for w in in_text.split() if w in word_to_idx]
        sequence = pad_sequences([sequence], maxlen=max_len, padding='post')

        ypred =  model.predict([photo,sequence])
        ypred = ypred.argmax()
        word = idx_to_word[ypred]
        in_text+= ' ' +word

        if word =='endseq':
            break


    final_caption =  in_text.split()
    final_caption = final_caption[1:-1]
    final_caption = ' '.join(final_caption)

    return final_caption
```

Figure 10: Testing the Model

V. RESULTS

A. *Perfectly Generated Captions*

Figure 11: Perfectly Generated Caption Example 1

Figure 12: Perfectly Generated Caption Example 2

B. Closely Related Captions

Figure 13: Closely Related Caption Example 1

Figure 14: Closely Related Caption Example 2

C. Unrelated Captions

Figure 15: Unrelated Captions Example 1

Figure 16: Unrelated Caption Example 2

VI. FUTURE SCOPE

The model is currently trained with flickr_8k dataset. In future the CNN-LSTM model can be trained against the dataset containing much larger volume of images like 1000000 images which will improve the overall accuracy of the model. Instead of LSTM we can use another RNN algorithm known as Long Term Recurrent Convolutional Neural Network. LRCN combines a deep hierarchical visual feature extractor (such as NN) with a model that can learn to recognize temporal dynamics for task involving sequential data, linguistics etc.

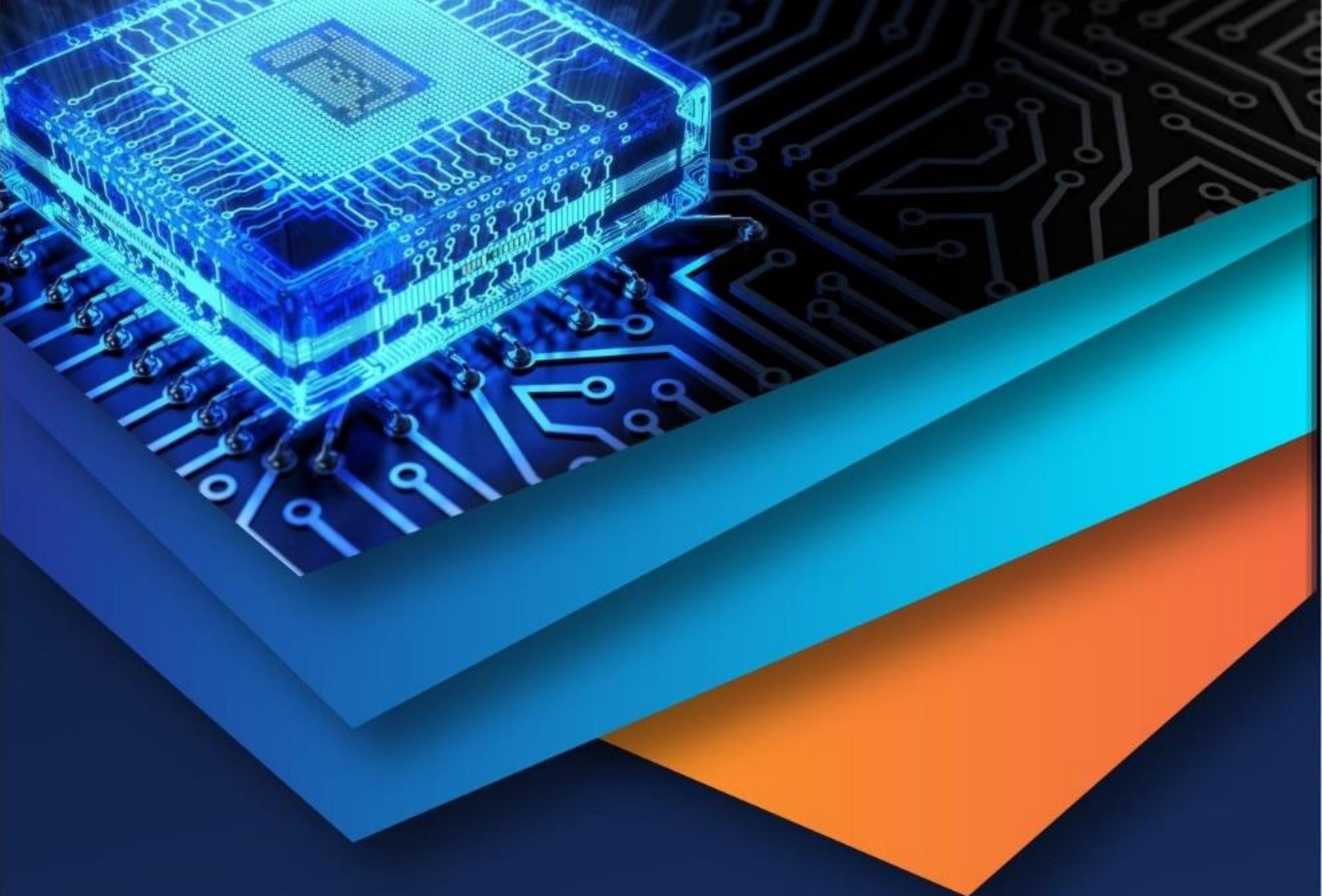
VII. CONCLUSION

We have proposed a system that will generate logical captions for an image. The model can also be tested for its evaluation against BLEU and METEOR metric system. The system is so designed that it will be able to mimic human like behaviour for describing an image. In addition to that the model that we have proposed, uses very few hard coded assumptions.
We hope that our research will encourage and help students in their future researches and areas of work.

REFERENCES

[1] V. Julakanti, "Image Caption Generator using CNN-LSTM Deep Neural Network", International Journal for Research in Applied Science and Engineering Technology, vol. 9, no., pp. 2968-2974, 2021.
[2] S.-H. Han and H.-J. Choi, "Domain-Specific Image Caption Generator with Semantic Ontology", IEEE International Conference on Big Data and Smart Computing (BigComp), 2020.
[3] M. Wang, L. Song, X. Yang, and C. Luo, " A parallel-fusion RNN-LSTM architecture for image caption generation ", IEEE International Conference on Image Processing (ICIP), 2016.
[4] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[5] S. Shukla, S. Dubey, A. Pandey, V. Mishra, M. Awasthi and V. Bhardwaj, "Image Caption Generator Using Neural Networks", International Journal of Scientific Research in Computer Science, Engineering and Information Technology, pp. 01-07, 2021.
[6] M. Panicker, V. Upadhayay, G. Sethi and V. Mathur, "Image Caption Generator", International Journal of Innovative Technology and Exploring Engineering, vol. 10, no. 3, pp. 87-92, 2021.
[7] J. Karan Garg and Kavita Saxena, "Image to Caption Generator", International Journal for Modern Trends in Science and Technology, vol. 6, no. 12, pp. 181-185, 2020.
[8] F. Fang, H. Wang, and P. Tang, "Image Captioning with Word Level Attention”, 25th IEEE International Conference on Image Processing (ICIP), 2018.
[9] Y. Huang, J. Chen, W. Ouyang, W. Wan and Y. Xue, "Image Captioning With End-to-End Attribute Detection and Subsequent Attributes Prediction", IEEE Transactions on Image Processing, vol. 29, pp. 4013-4026, 2020.
[10] P. Mathur, A. Gill, A. Yadav, A. Mishra, and N. K. Bansode, " Camera2Caption: A real-time image caption generator ", International Conference on Computational Intelligence in Data Science (ICCIDS), 2017.
[11] S. Amirian, K. Rasheed, T. Taha and H. Arabnia, "Automatic Image and Video Caption Generation With Deep Learning: A Concise Review and Algorithmic Overlap", IEEE Access, vol. 8, pp. 218386-218400, 2020.
[12] Y. Zhou, Y. Sun, and V. Honavar, " Improving Image Captioning by Leveraging Knowledge Graphs", IEEE Winter Conference on Applications of Computer Vision (WACV), 2019.
[13] Y. Zhenyu and Z. Jiao, “Research on Image Caption Method Based on Mixed Image Features", IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2019.
[14] M. Tanti, A. Gatt, and K. Camilleri, “What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator?", Proceedings of the 10th International Conference on Natural Language Generation, 2017.
[15] Simao Herdade, Armin Kappeler, Kofi Boakye, Joao Soares, “Image Captioning: Transforming Objects into Words.", Proceedings of the 10th International Conference on Natural Language Generation, 2017.
[16] A. Deshpande, J. Aneja, L. Wang, A.G. Schwing, D. Forsyth, “Fast, diverse and accurate image captioning guided by part-of-speech”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
[17] R. Subash, “Automatic Image Captioning Using Convolution Neural Networks and LSTM”, Journal of Physics Conference, 2019.
[18] B.Krishnakumar,K.Kousalya, S.Gokul,R.Karthikeyan, D.Kaviyarasu, “Image Caption Generation Using Deep Learning”, International Journal of Advanced Science and Technology, 2020.



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY