IMAGE CAPTION GENERATOR USING CONVOLUTION NEURAL NETWORK AND LONG SHORT TERM MEMORY

Dr.T.Gobinath
Senior Assistant Professor
gobinath@chettinadtech.ac.in

M.Mahalakshmi
Final CSE
mahalakshmicse24@gmail.com

M.S.Niranjana
Final CSE
niranjanasiva2001@gmail.com

P.Pranitha
Final CSE
pranithap68@gmail.com

Abstract - In artificial intelligence, automatically captioning an image is a difficult problem that combines computer vision and natural language processing, but the difficulty increases when the caption needs to be posted on Instagram. This is difficult since an Instagram caption incorporates more abstract elements than a straightforward explanation of the image, such as jokes, sarcasm, allusions, etc. While social media post captioning has come a long way, there is still plenty to be done. In light of this, I suggest a deep learning model that combines a Convolutional Neural Network (CNN) with a Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN). The first significant step was to get a dataset, however no such widely used dataset is currently accessible. The possibility of scraping Instagram was investigated, but it turned out that this was not allowed. Fortunately, I was able to obtain a dataset from Kaggle that had 35,000 records, but owing to computational constraints, I was only able to utilize 5000 photos and captions for training. The trained model ultimately performed poorly because of the low quality dataset, the low number of epochs (20), and the tiny training set.
Despite the fact that the model did not produce improved outcomes, an approach for creating captions for Instagram pictures was developed.

Keywords - CNN, LSTM, Neural Networks

I. INTRODUCTION :

Adding captions to social media postings has emerged as a new trend, and it has been observed that posts with captions receive more likes than ones without captions. Yet, coming up with the ideal, snappy title for the post may be challenging, and occasionally people will refrain from posting on social media because they lack a suitable description. Hence, I've decided to employ a deep learning model to automatically recommend captions to users as a solution. Convolutional neural networks (CNN) and long short-term memory (LSTM) recurrent neural networks are combined in my model to do this (RNN). A subset of deep neural networks known as convolutional networks, which analyze picture data primarily, consists of three layers: convolution, pooling, and fully connected. The majority

of the computation happens in the convolutional layer, which is the first layer in a CNN. Input data, a filter, and a feature map are the other three layers that make up the convolutional layer itself. After applying the convolutional layer, the pooling layer is used to lower the dimension of the feature map by down - sampling. It comes in two varieties: average pooling and maximum pooling. The Fully Connected Layer of CNN provides classification based on the characteristics collected from the preceding layers and their various filters. This layer acts on a flattened input where each node in the output layer links to a node in the preceding layer. A kind of deep neural network called recurrent neural networks allows output from earlier steps to be used as input in more recent phases. When dealing with issues in the area of Natural Language Processing (NLP), where we need to anticipate the previous words in a phrase in order to predict the subsequent words, this property of RNN makes it beneficial. Nevertheless, a significant drawback of RNN is that it performs better in short-term dependencies and worse in long-term dependencies. As a result, LSTM was proposed in order to store information for a long time.

My suggested model additionally adds some sarcasm, puns to the description and does so with less words than typical caption generating models, which just describe the items featured in the image and communicate their relation to one another. Several articles employ the MSCOCO, Flickr30k, or Pascal datasets, however the main issue with these datasets is that they include a basic description of the photographs, which will not enable us to achieve our aim. Prior to implementing the deep learning model, it is required to have the appropriate dataset. The only option, then, is to compile your own dataset of user postings and captions. As it is presently against the law to scrape user posts from Instagram, we were fortunate to find one on Kaggle. The dataset was 35,000 bytes, but the quality of it was quite bad, especially the captions, since the majority of them were stuffed with hashtags and other user tags. Even though I was able to remove a lot of pointless data after executing all the cleaning procedures, the issue of tagged people and hashtags continued to exist, which negatively impacted the trained model's quality. Nonetheless, the total procedure demonstrated that the model may perform better with a higher-quality dataset and more epochs. In order for the Keras model to operate on mobile devices, I also attempted to convert it to the Tensor Flow Lite model. However, Tensor Flow Lite does not presently support the conversion model that includes both CNN LSTM neural networks.

II. RELATED WORK :

There has been a lot of work done utilizing computer vision and natural language processing to describe the contents of pictures, such as the NIC model developed by Oriol Vinyals and colleagues, which featured CNN coupled to the RNN and could produce whole phrases from the input image. The Picture Caption Generator was created by the author using CNN and RNN-LSTM models that were trained on the Flickr 8k dataset. In an effort to reduce the computational cost of the model and enable it to operate on low-end hardware devices, the author used Deep Reinforcement Learning, however the model's accuracy suffered as a result. [3] The author compares numerous picture captioning systems, discusses their many flaws, and discusses them in detail. [4] The author discusses the applications of visual question answering (VQA) for more in-depth picture analysis. The study suggests combining top-down and bottom-up attention techniques such that

significant areas of at the object level, the image may be computed. The picture region associated with its feature is shown via a bottom-up technique. Vector whereas feature weights employ a top-down approach. [5] The author evaluates several methods of producing attention regions and comes to the conclusion that spatial transformers produce the greatest outcomes when combined with the capacity to simultaneously train with other network components and the image's unique attention areas. [7] The author discusses the role of RNNs in picture caption generation; in his opinion, RNNs perform the function of generators rather than encoders. [8] The Context Sequence Memory Network captioning model that the authors developed is really intriguing (CSMN). Their model updates itself from a prior memory network in the following ways: I it uses memory as storage for obtaining various types of context information; ii) it avoids the vanishing gradient problem by adding previously generated words to memory in order to capture long-term information; and iii) it uses CNN for having better factor understanding. Their algorithm made use of a dataset with 1.1 million posts from 6,300 Instagram users. To caption a movie, the authors [9] used LSTM and Generative Adversarial Network (GAN). The purpose of using GAN in this case was to make up for the mistakes made by LSTM. GAN provides text descriptions of the video's contents and also regulates the precision of the generated phrases. The use of GAN for text description is novel and merits further investigation. more. [10] Sherstinsky Alex addresses the principles of RNN and LSTM networks in his articles. He also analyzes the numerous challenges associated with training recurrent neural networks (RNN), and as a result, he recommends that RNN be converted into Vanilla LSTM. [11] For picture captioning, the authors of this research mix top-down and bottom-up techniques. Their model incorporates semantic ideas into hidden states and RNN outputs. Feedback created by the combination and selection connects top-down and bottom-up. [12] By implementing reinforcement learning and using the MS COCO dataset, the authors of this work attempted to increase the effectiveness of picture captioning. The authors of this research [13] present an approach that combines an LSTM with a conditional network, and they utilize the MS COCO dataset to explain how it works. [14] Instead of using manually labeled picture-sentence pairs, the authors of this article attempted to develop an unsupervised learning model that makes use of an image collection, a sentence corpus, and a visual concept detector. As a consequence, the suggested model was able to produce respectable outcomes. [15] The authors put up assessment criteria for picture captions that can distinguish between those created by people and those by computers. In terms of caption level and system-level human correlation assessment measures, they perform better than Flickr 8k and COCO, respectively. [16] The Attention on Attention module, which the authors presented in this research to quantify the significance between attention results and queries, was also applied to the encoder and decoder of the image captioning model. [17] To investigate the relationships between objects for picture captioning, the authors used Graph Convolutional Networks (GCN) in conjunction with LSTM. [18] For more believable captions, the authors suggested a Scene Graph Auto-Encoder (SGAE), which incorporates linguistic inductive bias into the encoder-decoder picture captioning system. When tested on the MS COCO dataset, the SGAE demonstrated a respectable performance. By adding the usage of previous and future context

information at high-level semantic space, the authors developed a CNN Bidirectional LSTM model to enhance the model's capacity to learn long-term visual-language interactions. [20] To aid in guessing the next word based on the present situation, the authors suggested a decision-making framework for picture captioning. Each of the players whose papers were addressed above focuses on creating a straightforward description of the items seen inside the image. My model, which is a little more complex than the straightforward explanation, is focused on creating captions for a social networking platform called Instagram.

III. ARCHITECTURE AND WORKING:

A. Convolution neural network :

CNN is a type of deep neural network that can categorize and identify pictures and objects by processing data as a 2D matrix. By reading or scanning the image from top to bottom or left to right, the details of the image may be retrieved. The information in the image may be obtained by analyzing the characteristics.

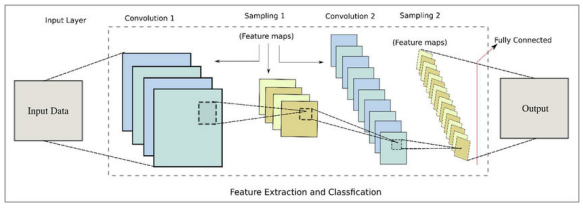

Figure 1 Architecture of CNN

Moreover, LSTM, an RNN that can anticipate based on past input or text and what should be a new word, can be utilized for sequence prediction purposes.

B. Working:

Python was used to make this work come to life. Several Python libraries were utilized for the implementation, such as Keras, which included the VCG net responsible for object detection, and TensorFlow, created by Google and used to build deep learning neural networks by performing a number of basic algorithms.

1. After feature extraction with CNN, we applied a number of layers to our model on the training set for which we had a text that corresponded to the activity seen in the image.

We were able to successfully extract the characteristics of each picture by employing multiple layers, including conv2d, max pooling, dropout, and activation function.

2. We then utilized Google's word2vec model to extract characteristics from further testing photos. Word2vec information It is a Google model that enables us to convert our words into numerals for use in processing that involves words. It is more frequently used in research relating to NLP, RNN, and LSTM.

3. When a word is provided, word2vec converts it into a vector with set dimensions.

4. Using the LSTM layer, we continue to keep the context of the words that make up sentences since the meaning of a phrase might change depending on the word and its placement.

As an illustration:

1. He is good.
2. He is useless.

As a result, the term "for nothing" altered the context of the entire phrase. To tackle this task, LSTM, which takes into account the word weights, is employed. Also, RNN is utilized as a recurrent model using the tan function, where the output of one cell is used as the input of the next cell. The model performed nicely for the training set and also on testing dataset samples likewise it gave about 85% good meaningful and accurate captioning.

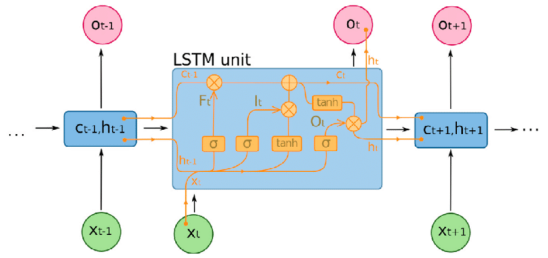

Figure 2 Architecture of LSTM by François Deloche

C. Proposed model:

The model consists of two parts: a convolutional neural network for classifying images and a recurrent neural network for modeling sequences. Hence, a 16-layer Oxford Visual Geometry Group (VGG) model was used to analyze the contents of the images, but the final layer was deleted in order to collect the extracted characteristics predicted by the model and then feed them into an RNN decoder that creates captions. ConvNet configurations are shown in Fig. 3 with the softmax layer removed for preprocessing, and a description of the model used for post processing training is shown in Fig. 4.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 3 ConvNet Configurations

Tokenization at the character level is used to keep the emojis in the caption intact. Breaking a statement into smaller pieces, such as individual words, is the process of tokenization. Each of these more compact parts is referred to as a token. The next step was encoding each caption by converting it to an array of integers, where each character was represented by an index. I created input-output vectors for each caption, with the input being the picture's feature vector plus the first n letters and the output being the n+1th character. There was a separate input-output pair made for each character in the caption vector. The picture and the first character of the caption were used as input in the first pairing, and the image and the complete caption were used as input and output in the final pairing. I only utilized 5,000 photographs for training, and a caption size of no more than 80 characters, thus this was done to prevent a lot of input-output pairs from becoming explosive and to reduce the bias of any one specific caption. Next, as a language model, I trained a Long Short-Term Memory (LSTM) decoder that accepted the feature vector and encoded character arrays as inputs and outputted an encoded character.

```
Model: "model_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_3 (InputLayer)            [(None, 82)]         0           []
input_2 (InputLayer)            [(None, 1000)]       0           []
embedding (Embedding)           (None, 82, 256)      187904      ['input_3[0][0]']
dropout (Dropout)               (None, 1000)         0           ['input_2[0][0]']
dropout_1 (Dropout)             (None, 82, 256)      0           ['embedding[0][0]']
dense (Dense)                   (None, 256)          256256      ['dropout[0][0]']
lstm (LSTM)                     (None, 256)          525312      ['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, 734)          188638      ['dense_1[0][0]']
==================================================================================================
Total params: 1,223,902
Trainable params: 1,223,902
Non-trainable params: 0
__________________________________________________________________________________________________
None
```

Figure 4 Model summary

IV RESULTS

The "flickr 8k" dataset that we used for our research is accessible online. The dataset underwent preprocessing to make it suitable for future analysis and use. It had 8000 photos, out of which we used 70:30 each for training and testing. We had a total of 25,636,712 parameters at the time of feature extraction, out of which 25,583,592 were successfully trained and 53,120 were

untrainable parameters. The performance of the system was examined using the general confusion matrix. This matrix includes the output from each model together with its forecasts. Over a regular batch size of 50 iterations, a total of 150 iterations were performed.

C. Accuracy :

Using the common equation shown below, the system's accuracy was measured

$$\text{ACCURACY} = \frac{\text{NUMBER OF CORRECT PREDICTIONS}}{\text{TOTAL NUMBER OF PREDICTIONS MADE}}$$

.

```
Epoch 1/10
1875/1875 [==============================] - 6s 3ms/step - loss: 0.6490 - accuracy: 0.8325
Epoch 2/10
1875/1875 [==============================] - 6s 3ms/step - loss: 0.3463 - accuracy: 0.9035
Epoch 3/10
1875/1875 [==============================] - 6s 3ms/step - loss: 0.2928 - accuracy: 0.9170
Epoch 4/10
1875/1875 [==============================] - 6s 3ms/step - loss: 0.2591 - accuracy: 0.9269
Epoch 5/10
1875/1875 [==============================] - 6s 3ms/step - loss: 0.2343 - accuracy: 0.9339
Epoch 6/10
1875/1875 [==============================] - 6s 3ms/step - loss: 0.2146 - accuracy: 0.9398
Epoch 7/10
1875/1875 [==============================] - 6s 3ms/step - loss: 0.1980 - accuracy: 0.9452
Epoch 8/10
1875/1875 [==============================] - 6s 3ms/step - loss: 0.1834 - accuracy: 0.9486
Epoch 9/10
1875/1875 [==============================] - 6s 3ms/step - loss: 0.1712 - accuracy: 0.9515
Epoch 10/10
1875/1875 [==============================] - 6s 3ms/step - loss: 0.1602 - accuracy: 0.9553
```

Figure 5 Number of epochs executed with loss values

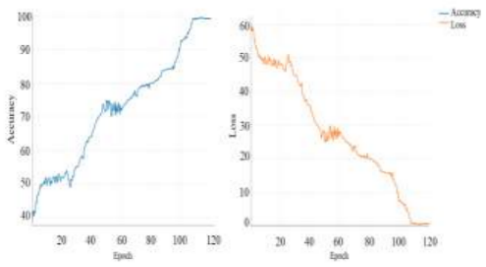

Figure 6 Epoch vs. lossand Epoch vs. accuracy

We can tell from the graph that our model was implemented successfully, and we are eager to train it on a larger dataset.

V CONCLUSION :

By creating a model based on LSTM based RNN capable of scanning and extracting information from any supplied image and changing it into a single line phrase based on a natural language English, we have overcome past limitations that were experienced in the field of image captioning. Most of the time, it is mentioned that it can be difficult to prevent overfitting of data, but we are happy that we have been able to do so. The main emphasis was on the core algorithms of various attention mechanisms and a summary of how the attention mechanism is used.

VI FUTURE SCOPE :

In the future, we would aim to train our model on a larger dataset made up of more photos, which would produce a model with more accuracy, efficiency, and horizon. Also, we wish to introduce our idea to a bigger group of individuals, mostly blind people. A product that will assist blind people in crossing highways and ensure that they may go securely anywhere without depending on the kindness of others can be produced by utilizing IoT equipment like Arduino kits, Electrical Equipment, Cameras, and a few other items like Bluetooth.

In order to record real-time surroundings video and gain a mechanism to wirelessly connect it to the blind person's Bluetooth in-ear, we were able to implant a camera in the front face of the shoe, as seen in the image. The captions will be created in a dynamic environment and designed to be played in the blind person's Bluetooth device so that he can cross with greater caution now that this Arduino equipment is being employed. By doing this, accidents and incidents notably affecting blind persons would surely decrease.

VII REFERENCES :

[1] O. Vinyals, A. Toshev, S. Bengio and D. Erhan, "Show and tell: A neural image caption generator," 2015 IEEE Conference on Computer Vision and Pattern

Recognition (CVPR), 2015, pp. 3156-3164, doi: 10.1109/CVPR.2015.7298935.

[2] P. Mathur, A. Gill, A. Yadav, A. Mishra and N. K. Bansode, "Camera2Caption: A real-time image caption generator," 2017 International Conference on Computational Intelligence in Data Science(ICCIDS), 2017, pp. 1-6, doi:10.1109/ICCIDS.2017.8272660.

[3] Wang, Haoran, Yue Zhang, and Xiaosheng Yu. "An overview of image caption generation methods." Computational intelligence and neuroscience 2020 (2020).

[4] P. Anderson et al., "Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 6077-6086, doi: 10.1109/CVPR.2018.00636.

[5] Pedersoli M, Lucas T, Schmid C, Verbeek J (2017) Areas of attention for image captioning. In: 2017 IEEE international conference on computer vision (ICCV), pp 1251–1259

[6] Preksha Khant, Vishal Deshmukh, Aishwarya Kude, Prachi Kiraula, “Image Caption Generator using CNN-LSTM” International Research Journal of Engineering and Technology (IRJET), 2021

[7] Tanti M, Gatt A, Camilleri KP. What is the role of recurrent neural networks (rnns) in an image caption generator?. arXiv preprint arXiv:1708.02043. 2017 Aug 7.

[8] Chunseong Park C, Kim B, Kim G. Attend to you: Personalized image captioning with context sequence memory networks. InProceedings of the IEEE conference on computer vision and pattern recognition 2017 (pp. 895-903).

[9] Yang, Yang & Zhou, Jie & Ai, Jiangbo & Bin, Yi & Hanjalic, Alan & Shen, Heng & Ji, Yanli. (2018). Video Captioning by Adversarial LSTM. IEEE Transactions on Image Processing. 27. 1-1. 10.1109/TIP.2018.2855422.

[10] Sherstinsky A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena. 2020 Mar 1;404:132306.

[11] You, Quanzeng, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. "Image captioning with semantic attention." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4651-4659. 2016.

[12] Rennie, Steven J., Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. "Self-critical sequence training for image captioning." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7008-7024. 2017.

[13] Aneja, Jyoti, Aditya Deshpande, and Alexander G. Schwing. "Convolutional image captioning." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5561-5570. 2018.

[14] Feng, Yang, Lin Ma, Wei Liu, and Jiebo Luo. "Unsupervised image captioning." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4125-4134. 2019.

[15] Cui, Yin, Guandao Yang, Andreas Veit, Xun Huang, and Serge Belongie. "Learning to evaluate image captioning." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5804-5812. 2018.

[16] Huang, L., Wang, W., Chen, J. and Wei, X.Y., 2019. Attention on attention for image captioning. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 4634-4643).

[17] Yao, Ting, Yingwei Pan, Yehao Li, and Tao Mei. "Exploring visual relationship for image captioning." In Proceedings of the European conference on computer vision (ECCV), pp. 684-699. 2018.

[18] Xu Yang, Kaihua Tang, Hanwang Zhang, Jianfei Cai. Auto-Encoding Scene Graphs for Image Captioning. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pages 10685-10694, Computer Vision Foundation IEEE, 2019. [doi]

[19] Wang, C., Yang, H. and Meinel, C., 2018. Image Captioning with Deep Bidirectional LSTMs and Multi-Task Learning. ACM Transactions on Multimedia Computing, Communications, and Applications, 14(2s), pp.1-20.

[20] Ren Z, Wang X, Zhang N, Lv X, Li LJ. Deep reinforcement learning-based image captioning with embedding reward. InProceedings of the IEEE conference on computer vision and pattern recognition 2017 (pp.290-298).

[21] M. P. R, M. Anu and D. S, "Building A Voice Based Image Caption Generator with Deep Learning," 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2021, pp. 943-948, doi: 10.1109/ICICCS51141.2021.9432091.

[22] V. Agrawal, S. Dhekane, N. Tuniya and V. Vyas, "Image Caption Generator Using Attention Mechanism," 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2021, pp. 1-6, doi: 10.1109/ICCCNT51525.2021.9579967.

[23] P. Mathur, A. Gill, A. Yadav, A. Mishra and N. K. Bansode, "Camera2Caption: A real-time image caption generator," 2017 International Conference on Computational Intelligence in Data Science(ICCIDS), Chennai, India, 2017, pp. 1-6, doi: 10.1109/ICCIDS.2017.8272660.

[24] O. Vinyals, A. Toshev, S. Bengio and D. Erhan, "Show and tell: A neural image caption generator," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 3156-3164, doi: 10.1109/CVPR.2015.7298935.

[25] A. Verma, H. Saxena, M. Jaiswal and P. Tanwar, "Intelligence Embedded Image Caption Generator using LSTM based RNN Model," 2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2021, pp. 963-967, doi: 10.1109/ICCES51350.2021.9489253.

[26] M. M. A. Baig, M. I. Shah, M. A. Wajahat, N. Zafar and O. Arif, "Image Caption Generator with Novel Object Injection," 2018 Digital Image Computing: Techniques and Applications (DICTA), Canberra, ACT, Australia, 2018, pp. 1-8, doi: 10.1109/DICTA.2018.8615810.

[27] N. K. Kumar, D. Vigneswari, A. Mohan, K. Laxman and J. Yuvaraj, "Detection and Recognition of Objects in Image Caption Generator System: A Deep

Learning Approach," 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 2019, pp. 107-109, doi: 10.1109/ICACCS.2019.8728516.

[28] S. -H. Han and H. -J. Choi, "Explainable Image Caption Generator Using Attention and Bayesian Inference," 2018 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2018, pp. 478-481, doi: 10.1109/CSCI46756.2018.00098.

[29] A. Singh and D. Vij, "CNN-LSTM based Social Media Post Caption Generator," 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM), Gautam Buddha Nagar, India, 2022, pp. 205-209, doi: 10.1109/ICIPTM54933.2022.9754189.