International Journal of Development Mathematics Vol 1 Issue 3, Page No. 162 - 173

INTERNATIONAL JOURNAL OF DEVELOPMENT MATHEMATICS

ISSN: 3026-8656 (Print) | 3026-8699 (Online)

journal homepage: https://ijdm.org.ng/index.php/Journals

An Intelligent Image Caption Generator Model using Deep Learning

Kayode O. Oluborode[a*], Adaiti A. Kadams[a], Mohammed Usman[a]

[a]Department of Computer Science, Modibbo Adama University, Yola, Nigeria.

ARTICLE INFO

Article history:
Received 11 May, 2024
Received in revised form 18 July, 2024
Accepted 29 July, 2024

Keywords:
CNN, Image Caption Generator, LSTM.

MSC 2020 Subject classification:
68M07, 68M10

ABSTRACT

Replicating the innate human capability to understand image content and create descriptive text is a formidable challenge for machines. The application of image captioning is highly versatile and holds great importance. It involves the creation of concise and descriptive captions for images using a variety of advanced techniques such as Natural Language Processing (NLP), Computer Vision (CV), and Deep Learning (DL). These techniques enable the automated generation of accurate and meaningful captions, which have numerous practical applications across industries such as healthcare, entertainment, marketing to mention but few. This study employed the machine learning algorithms of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) to train computer vision using the Hibs dataset. The model leveraged input images with associated captions to identify and analyze predicted captions within the computer system. Model evaluation is a step in assessing the model's performance. This involves metrics such as BLEU score and METEOR score to measure the quality of the model's output in comparison to reference translation. These metrics help in quantifying the accuracy and fluency of the generated text, and the results are then presented to provide insights into the model's performance.

1. Introduction

The focus is on developing a tool to automatically generate descriptive captions for images. This paper explores converting visual representations into NLP and analysing image scenes and extracting features (Chaithra *et al.,* 2022). Every day, various images flood the internet, news articles, and ads, presenting diverse ideas and information. People usually understand them, but now there are systems that add words to pictures, showing the need for help understanding images. an intelligent image caption generator model using deep learning highlight the need for interpretive help. Image captioning is crucial for improving internet searches and indexing. As computer vision technology advances, the value of detailed descriptions becomes clear. Image captioning has diverse uses in biomedicine, commerce, information retrieval, defence, and popular social media platforms (Rohan, 2020).

Image captioning is a valuable tool in Artificial Intelligence (AI) applications. It involves creating written descriptions for images and requires combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) architectures. Despite progress in computer vision using Hibs dataset, enabling computers to describe visual stimuli is still relatively new. Image captioning frameworks make tasks such as object recognition and scene understanding feasible using CNNs for feature extraction. LSTM models use image features to create descriptions. Image captioning, driven by AI, is crucial in today's digital world. These systems aim to automatically describe images, enhancing user experiences on social media and other platforms (Dhirendra and Minu, 2022). In the age of artificial intelligence, automatic image descriptions will enhance human interaction with digital content.

* Corresponding author. Tel.: +2348034409227
E-mail addresses: kayodeoo@mau.edu.ng (Kayode O. O.)
https://doi.org/10.62054/ijdm/0103.12

2. Related Works

The landscape of image caption generation within deep learning frameworks is rich and diverse, encompassing a plethora of studies employing various datasets and methodologies. Chaithra, *et al.*, (2022) present a model leveraging deep learning to generate descriptive captions from images, utilizing the Flickr8K dataset and implementing a neural network architecture capable of generating English text captions. Their model classifies generated captions into four categories based on error presence and relevance to the image content. Akash, *et al.*, (2022) contribute to the field with a model employing VGG16 Hybrid places as an encoder and LSTM as a decoder, trained on Flickr8k and MSCOCO captions datasets. Their evaluation encompasses standard metrics such as BLEU, METEOR, GLEU, and ROUGEL.

Aishwarya *et al.*, (2021) introduce a deep-learning model harnessing Convolutional Neural Network (ResNet) as an encoder and Recurrent Neural Network (LSTM) as a decoder, facilitating the generation of image captions through the amalgamation of deep learning and Natural Language Processing (NLP) techniques. Sreejith, *et al.*, (2021) contribute a model utilizing pre-trained deep machine learning in a Python Jupiter environment, with Keras 2.0 backend on the Tensorflow library. Their evaluation reveals a high degree of similarity between human and model-generated captions, indicating an accuracy of approximately 75%. Preksha *et al.*, (2021) devise an image caption generator employing CNN-LSTM architecture trained on the Flickr_8k database, achieving grammatically structured captions through the combined encoding-decoding process.

Vinayak *et al.*, (2020) amalgamate concepts from computer vision and natural language processing domains, employing CNN and LSTM models to automatically generate descriptive sentences for imported images. Their evaluation utilizes BLEU Scores to assess model efficiency. Rohan, (2020) employs CNN and LSTM models alongside natural language processing and computer vision concepts to develop an image caption generator. The work demonstrates the utilization of libraries such as Keras and numpy, and flickr_dataset for image classification. Ali, (2020) presents an image captioning model utilizing CNN and LSTM, incorporating image feature extraction and sequence processing phases, with evaluation based on BLEU score algorithm. Subrata, *et al.*, (2018) contribute to military image captioning using deep learning, employing hybrid RNN/CNN deep network models and implementing an image captioning model with the deep fusion concept. Evaluation encompasses both subjective and objective assessment through BLEU score calculation and NLTK package for evaluation.

3. Methodology

In this section, we outline the methodology utilized in this study. The main goal was to develop a model for generating image captions using a Convolutional Neural Network (CNN) as an encoder and a Long Short-Term Memory (LSTM) as a decoder. The model was thoroughly assessed to validate its ability to learn from the Hibs dataset. Image caption generators typically combine computer vision techniques for image comprehension with Natural Language Processing (NLP) methods for text generation.

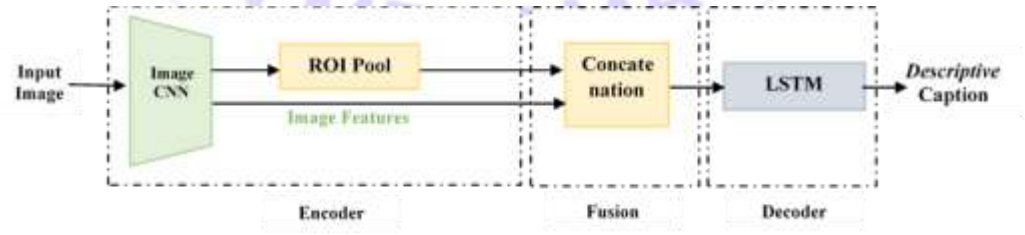

Figure 1: Model Architecture

In the model architecture described above, detailed Image Captioning entails generating a textual description of an image using deep learning models, typically by combining Convolutional Neural Networks (CNNs) and Long Short-

Term Memory networks (LSTMs). In this model, CNNs are utilized to extract features from the input image dataset. The networks are pre-trained using Visual Geometry Group (VGG16), which is a standard deep Convolutional Neural Network (CNN) architecture with multiple layers. The model uses 13 convolutional layers to extract image features and takes advantage of the network's ability to generate rich feature representations. It also incorporates 5 max-pooling layers and eliminates the 3 fully connected layers typically used for classification. The Rectified Linear Unit (ReLU) is employed as the activation function for the convolutional layers in neural networks. The output from the last convolutional layer or a pooling layer just before the fully connected layers is utilized as the feature map. For instance, a 7x7x512 feature map was produced as the output from the final pooling layer.

The model also incorporates the concept of Region of Interest (ROI) to attend to different regions of the image at different times. This allows the model to generate contextually relevant words while describing a character in an image dataset. The mechanism focuses on actual parts of an image when generating each word in the caption, thereby improving the description by focusing on relevant parts of the image. This spatial awareness ensures that the generated captions are closely aligned with the content of the image. The context vector from the attention mechanism is combined with word embeddings and fed into an LSTM, which sequentially generates the caption. This integrated approach enables the model to produce contextually relevant and descriptive captions for images.

3.1 Feature extraction

To extract rich, high-level features from the input image that capture important spatial information necessary for generating descriptive captions. A pre-trained VGG16 network, which is known for its depth and ability to capture intricate details of images, is used for feature extraction.

Truncated VGG16: The network is typically truncated before the fully connected layers, keeping the convolutional layers which output a spatial feature map.

Output Feature Map: The resulting feature map from the last convolutional block is used for further processing. For a 224x224 input image, the feature map might be a 7x7x512 tensor.

Input Image: A 224x224 RGB image.

VGG16 Output: A 7x7x512 feature map which encapsulates high-level visual information.

3.2 Word embeddings

To convert words in the vocabulary into dense vectors that can be processed by neural networks. This is crucial for feeding text data into the LSTM (Long Short-Term Memory) network.

Embedding Layer: An embedding layer is used to transform each word in the caption into a fixed-dimensional vector. These embeddings capture semantic relationships between words.

Training: The word embeddings pre-trained on large text corpora using Word2Vec, learned jointly with the image captioning model.

Vocabulary: Due to small volume of the dataset used, vocabulary size is 1000 words.

Embedding Dimension: Each word is converted into a 256-dimensional vector.

3.3 Decoder network

To generate the caption for the image one word at a time by combining the image features with the word embeddings of the previously generated words.

LSTM (Long Short-Term Memory): The core of the decoder network. LSTM is well-suited for sequential data and helps in maintaining the context over long sequences.

Attention Mechanism: Similar to ROI pooling, it dynamically focuses on different parts of the feature map at each step of the caption generation process, creating a context vector.

Fusion: The context vector from the attention mechanism is combined with the word embeddings and fed into the LSTM to predict the next word in the sequence.

Initialize with Start Token: The caption generation starts with a special <start> token.

Generate Context Vector: The attention mechanism calculates attention weights over the feature map, producing a context vector.

Predict Next Word: The context vector and the embedding of the current word are fed into the LSTM to predict the next word.

Repeat: This process is repeated until the <end> token is generated, signaling the end of the caption.

Input: The feature map (7x7x512) and the embedding of the <start> token.

Output: The next word in the caption and updated LSTM states.

3.4 Performance Evaluation

To assess the accuracy and quality of the generated captions against ground truth captions.

The metrics like BLEU (Bilingual Evaluation Understudy), which measures the precision of n-grams in the generated captions compared to the reference captions. It considers precision for different lengths of n-grams (typically up to 4-grams) and METEOR (Metric for Evaluation of Translation with Explicit Ordering) which considers precision, recall, and alignment of chunks between the generated and reference captions, also factoring in synonyms and stemming.

Finally, the both metrics were compared with each other to know the actual perfect one among.

4. Results and Discussion

The view of the work is for machine learning to utilize the techniques of deep learning algorithms of CNN and LSTM to generate predicted captions of images trained from a dataset.

1. Importing the needed Modules

   - Os: – It handles files using system commands.
   - Pickle: - used to store numpy features extracted.
   - Numpy: - used to perform a wide variety of mathematical operations on arrays.
   - Tqdm: - progress bar decorator for iterators. Includes a default range iterator printing to stderr.
   - VGG16, preprocess_input: - imported modules for feature extraction from the image data.
   - Load_img, img_to_array: - used for loading the image and converting the image to a NumPy array.
   - Tokenizer: - used for loading the text and converting it into a token.
   - Pad_sequences: - used for equal distribution of words in sentences filling the remaining spaces with zeros.
   - Plot_model: - used to visualize the architecture of the model through different images.

2. Extract Image Features by loading the CNN model of VGG16 and also restructure the VGG16 model. extracted features of the image to fit the model of VGG16 to 224, and also to load the images from the dataset and then store them.

   - Dictionary 'features' is created and will loaded with the extracted features of image data.
   - Load_img(img_path, target_size=(224, 224)) - custom dimension to resize the image when loaded to the array.
   - Image.reshape((1, image.shape[0], image.shape[1], image.shape[2])) - reshaping the image data to preprocess in a RGB type image.
   - Model.predict(image, verbose=0) - extraction of features from the image.
   - Img_name.split('.')[0] - split of the image name from the extension to load only the image name.

3. At this point, separate and attach the image's caption data.
   - This loads the total number of captions in the text file of the dataset.
   - The dictionary' mapping' is constructed with image_id as the key and the accompanying caption text as the values.
   - If image_id is not in mapping, the same image may have more than one caption: mapping[image_id] = [] generates a list that may be used to add captions to the associated images.
   - This will load the total number of images in the dataset.

4. Preprocessing the text data.
   - By cleaning and removing letters, characters or symbols in the text file that are not needed.
   - Defined to clean and convert the text for quicker process and better results.

- Visualization the text before preprocessing it.
- Preprocess the text to fit well.
- The result after preprocessing the text.
- Words with one letter were deleted.
- All special characters were deleted.
- 'startseq' and 'endseq' tags were added to indicate the start and end of a caption for easier processing.
- Load the total number of captions in the dataset.
- The first 15 captions of the text file.
- Start processing the text.
- Number of unique words.
- Determining the longest caption possible, which will be used as a guide for the padding sequence.

5. Test the split
   - Following data preprocessing, to train, test, and divide the data.
   - 90% used which is denoted as 0.90.
   - Then will now include the padding sequence and define a batch by creating a data generator.
   - Padding sequence normalizes the size of all captions to the max size filling them with zeros for better results.
6. Model creation of the CNN(encoder) and LSTM(decoder) model image caption generator
   - Shape=(4096,) - the VGG model's feature output length
   - Dense: a linear layer array with only one dimension
   - Dropout() is a tool for adding regularization to data, preventing overfitting, and removing a portion of the layers' data.
   - Model.compile() – the model's compilation
   - Loss="sparse_categorical_crossentropy." - loss function for outputs in a category
   - Optimizer='adam' - automatically modifies the model's learning rate throughout the number of epochs
   - Concatenation of the inputs and outputs into a single layer is displayed in the model plot.
   - VGG was already used to extract the image's features.
7. Generate the captions
   - Caption generator that adds all the text to a picture.
   - The model predicts the caption until the 'endseq' appears, starting with the 'startseq' in the caption.
8. The results of the images and captions with their description.
9. Test with Real Images
10. Validate the data using BLEU and METEOR score in order to evaluate the performance of the model.
    - In a collection of tokens, the predicted text is assessed using the BLEU and METEOR Score in comparison to a reference text.
    - Every word appended from the captions data (actual_captions) is present in the reference text.
    - An outcome deemed satisfactory is a BLEU and METEOR Score more than 0.4, 0.2 respectively; to achieve a higher score, increase the number of epochs correspondingly.

4.1 Image caption generator

This cutting-edge model seamlessly combines visual and textual data to generate vivid and contextually appropriate image captions, effectively translating visual content into human language.

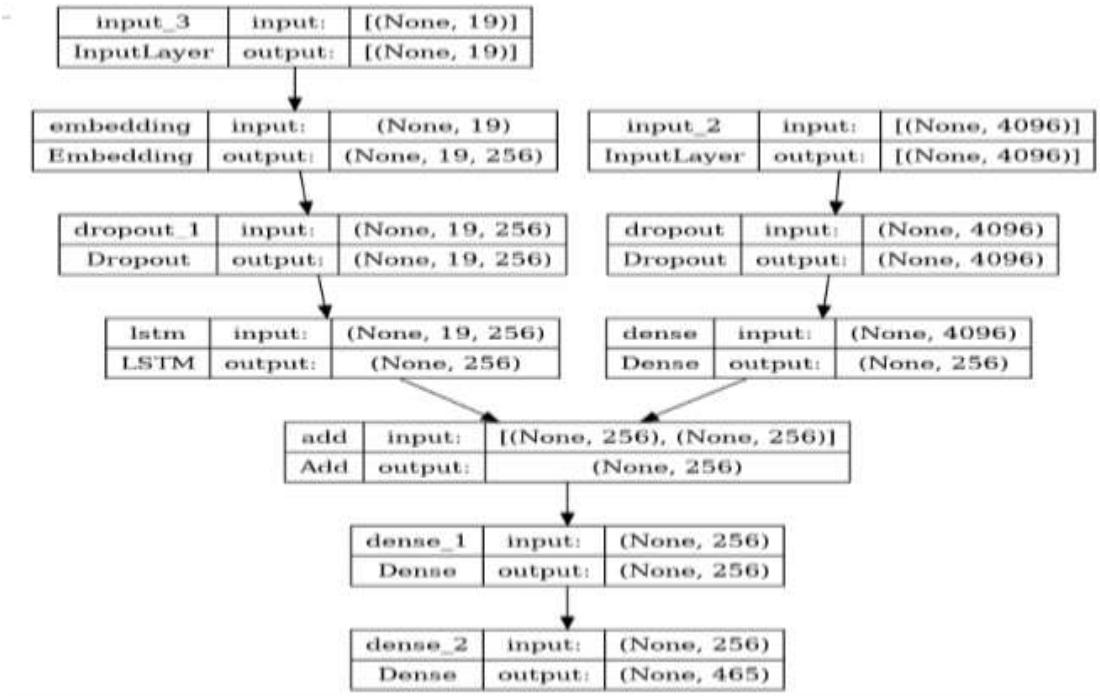

Figure 2: Image Caption Generator model

4.2 VGG16 training

This process guarantees that the model is trained to produce precise and vivid captions for images. It harnesses the robust feature extraction abilities of VGG16 and the sequential modeling strengths of LSTM networks with attention mechanisms.

Train the model with epochs and batch size.

- Steps = len(train) // batch_size - back propagation and fetch the next data.
- Loss decreases gradually over the iterations.
- Increase the number of epochs for better results.
- Assign the number of epochs and batch size accordingly for quicker results.

The Table 1 shows the block of VGG16 utilized as the backbone for this model, the trainable parameters are primarily resided in the decoder network, VGG16 is exclusively employed for feature extraction without any fine-tuning. However, it's layers are fine-tuned, these contribute to the trainable parameters. The decoder typically encompasses an embedding layer, an LSTM layer, and dense layers, alongside an attention mechanism, all of which collectively contribute to the trainable parameters.

Table 1: VGG16 Trainable Parameter

553467096/553467096 [==============================] - 5s 0us/step
Model: "model"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928

block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312

Total params: 134260544 (512.16 MB)
Trainable params: 134260544 (512.16 MB)
Non-trainable params: 0 (0.00 Byte)

Table 2: VGG16 Training Process

1/1 [==============================] - 0s 229ms/step - loss: 0.7301
1/1 [==============================] - 0s 220ms/step - loss: 0.7181
1/1 [==============================] - 0s 204ms/step - loss: 0.7096
1/1 [==============================] - 0s 198ms/step - loss: 0.7051
1/1 [==============================] - 0s 196ms/step - loss: 0.6963
1/1 [==============================] - 0s 191ms/step - loss: 0.6870
1/1 [==============================] - 0s 186ms/step - loss: 0.6761
1/1 [==============================] - 0s 186ms/step - loss: 0.6675
1/1 [==============================] - 0s 186ms/step - loss: 0.6647
1/1 [==============================] - 0s 187ms/step - loss: 0.6568
1/1 [==============================] - 0s 192ms/step - loss: 0.6406
1/1 [==============================] - 0s 188ms/step - loss: 0.6356
1/1 [==============================] - 0s 195ms/step - loss: 0.6250
1/1 [==============================] - 0s 188ms/step - loss: 0.6251
1/1 [==============================] - 0s 198ms/step - loss: 0.6112
1/1 [==============================] - 0s 193ms/step - loss: 0.6113
1/1 [==============================] - 0s 191ms/step - loss: 0.6083
1/1 [==============================] - 0s 193ms/step - loss: 0.5928
1/1 [==============================] - 0s 184ms/step - loss: 0.5955
1/1 [==============================] - 0s 207ms/step - loss: 0.5799
1/1 [==============================] - 0s 192ms/step - loss: 0.5771
1/1 [==============================] - 0s 189ms/step - loss: 0.5719
1/1 [==============================] - 0s 185ms/step - loss: 0.5636
1/1 [==============================] - 0s 193ms/step - loss: 0.5530
1/1 [==============================] - 0s 184ms/step - loss: 0.5500
1/1 [==============================] - 0s 188ms/step - loss: 0.5435

```
1/1 [==============================] - 0s 218ms/step - loss: 0.5389
1/1 [==============================] - 0s 214ms/step - loss: 0.5355
1/1 [==============================] - 0s 210ms/step - loss: 0.5328
1/1 [==============================] - 0s 241ms/step - loss: 0.5185
```

4.2 Sample of image datasets

The Hibs Dataset is a curated collection of images sourced from local sources, meticulously compiled to serve as training data for an advanced image caption generator model based on deep learning algorithms. This dataset currently encompasses a set of 100 diverse images, carefully selected to represent various scenarios and objects. The primary objective of this initiative is to assess the efficacy of training a captioning model with a relatively small dataset, thereby shedding light on the potential impact of limited data on the model's performance.

Figure 3: Samples of Hibs Dataset used

4.3 Image output and their descriptions

Figure 4: Output of First and Second Sample of Image Captioning and their Description

Figure 5: Output of Third and Fourth Sample of Image Captioning and their Description

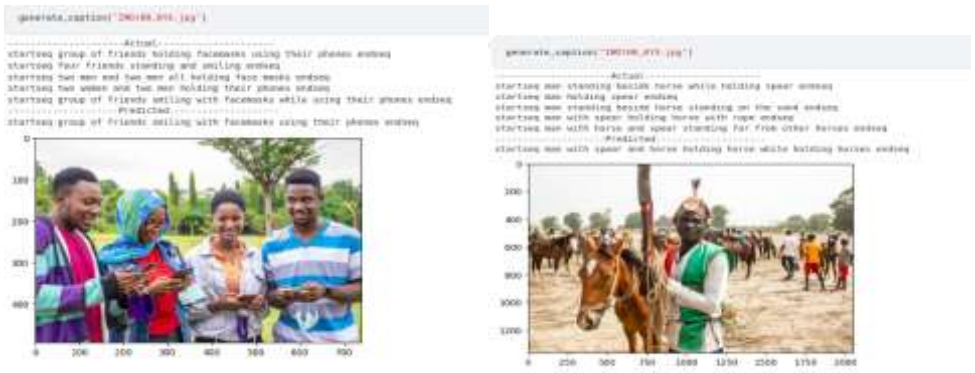

Figure 6: Output of Fifth and Sixth Sample of Image Captioning and their Description

Figure 7: Output of Seventh and Eighth Sample of Image Captioning and their Description

4.4 Performance metric

The evaluation of image caption generators depends heavily on three key metrics: BLEU-1, BLEU-2, and METEOR scores. BLEU-1 specifically assesses the accuracy of individual words in the generated captions, while BLEU-2 focuses on the precision of short sequences of words. METEOR, on the other hand, provides insights into the semantic and syntactic quality of the generated captions. By delving into the nuances of these metrics and actively working to improve them, we can refine and advance the development of image captioning models, ultimately leading to the creation of more precise and contextually coherent captions for images.

Table 3: BLEU-1, 2 and METEOR Scores on Proposed Model

10%| | 1/10 [00:09<01:21, 9.01s/it] Metric | Score --- BLEU-1 | 0.600000 BLEU-2 | 0.434956 METEOR | 0.272921	40%| | 4/10 [00:11<00:11, 1.88s/it] Metric | Score --- BLEU-1 | 0.413043 BLEU-2 | 0.140245 METEOR | 0.220294	70%| | 7/10 [00:12<00:02, 1.00it/s] Metric | Score --- BLEU-1 | 0.402597 BLEU-2 | 0.131355 METEOR | 0.218764	100%| | 10/10 [00:14<00:00, 1.47s/it] Metric | Score --- BLEU-1 | 0.413462 BLEU-2 | 0.148299 METEOR | 0.222792
20%| | 2/10 [00:09<00:32, 4.02s/it] Metric | Score --- BLEU-1 | 0.526316 BLEU-2 | 0.400846 METEOR | 0.211416	50%| | 5/10 [00:11<00:06, 1.39s/it] Metric | Score --- BLEU-1 | 0.436364 BLEU-2 | 0.132116 METEOR | 0.221280	80%| | 8/10 [00:13<00:01, 1.18it/s] Metric | Score --- BLEU-1 | 0.423529 BLEU-2 | 0.148329 METEOR | 0.223195	
30%| | 3/10 [00:10<00:17, 2.51s/it] Metric | Score --- BLEU-1 | 0.451613 BLEU-2 | 0.127000 METEOR | 0.219334	60%| | 6/10 [00:12<00:04, 1.08s/it] Metric | Score --- BLEU-1 | 0.444444 BLEU-2 | 0.152944 METEOR | 0.225224	90%| | 9/10 [00:14<00:00, 1.25it/s] Metric | Score --- BLEU-1 | 0.416667 BLEU-2 | 0.154746 METEOR | 0.230885	

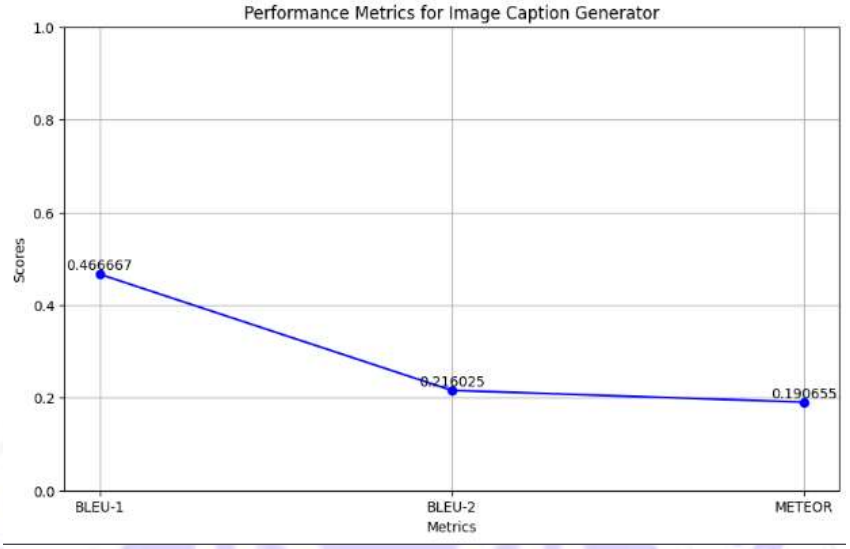

Figure 8: Comparison Between BLEU-1, BLEU-2 and METEOR_Score Performance Metrics

5. Conclusion

The field of image caption generation presents a rich domain for research within the realms of computer vision and artificial intelligence, offering pathways for progress in image detection and recognition through rigorous simulation and modelling endeavours. The diverse applications of image captioning span various sectors including biomedicine, commerce, web search, military operations, and the realm of autonomous vehicles for efficient navigation. Furthermore, the integration of image captioning capabilities into social networking platforms such as Facebook and Instagram signifies a paradigm shift, enabling automated caption generation for images across diverse contexts. The versatility of image caption generators extends to images of any pixel configuration, whether in colour or monochrome, allowing for seamless integration into diverse digital environments. At its core, the overarching objective of image caption generators remains the elucidation of contextual nuances within images, thereby providing natural language descriptions, predominantly in English. This endeavour draws upon the amalgamation of principles from computer vision and natural language processing, underscoring the interdisciplinary nature of this evolving field. As research in image captioning continues to progress, it holds the promise of revolutionizing how we interact with and interpret visual content, bridging the semantic gap between images and textual descriptions. Looking ahead, concerted efforts in this domain are poised to unlock novel avenues for innovation, fostering enhanced understanding and communication in an increasingly visual-centric digital landscape.

References

Aishwarya, M., Sneha, S.D. & Lahari, C. (2021) Image Caption Generating Deep Learning Model, *International Journal of Engineering Research and Technology (IJERT), Vol. 10 Issue 09,* September-2021, pp. 616-618.

Ali A. M. (2020) Image Caption Using CNN and LSTM.

Akash, V., Arun, K.Y., Mohit, K. & Divakar, Y. (2022) Automatic Image Caption Generation Using Deep Learning, *National Institute of Technology Hamirpur, Department of CSE, NIT Hamirpur, June 21st, 2022,* pp. 1-6.

Chaithra, V., Charitra R., Deeksha, K., Shreya, & Jagadisha, N. (2022). Image Caption Generator Using Deep Learning. International Research Journal of Engineering and Technology (IRJET) Volume: 09 Issue: 01 | Jan 2022. p-ISSN: 2395-0072. www.irjet.net.

Dhirendra P. & Minu C. (2022). Image Caption Generator using Deep Learning with Flickr Dataset. *International Journal for Research Trends and Innovation (IJRTI)* 7(8) | ISSN: 2456-3315

Preksha, K., Vishal, D., Aishwarya, K., Prachi, K., (2021). Image Caption Generator using CNN-LSTM *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

Rohan, P., (2020) Image Caption Generator project, A Project Report of Capstone Project – 2, *Galgotias University*, *School of Computing and Science and Engineering*, 8- 12.

Sreejith, S. P. & Vijayakumar, A. (2021). Image Captioning Generator using Deep Machine Learning. *International Journal of Trend in Scientific Research and Development (ijtsrd)*, 5(4), 832-834

Subrata, D., Lalit, J. & Arup, D. (2018). Deep Learning for Military Image Captioning. doi.10.23919/ICIF.2018.8455321, 2165-2171.

Vinayak, D.S., Mahiman, P. D., Anuj M. S., Amit, C. D., (2020), Image Caption Generator Using Big Data and Machine Learning, *International Research Journal of Engineering and Technology (IRJET),* 7(4). 6197-6199.