International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 04 | April - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

IMAGE CAPTION GENERATOR USING DEEP LEARNING

[1] Sufiyan Ali Khan,[2] G Swetha, [3] G Thejashwini, [4] H Varshith, [5] M Varun Teja, [6] A Siva Kumar

[12345] Student, [6] Professor

School of Engineering – IIIrd year AI&ML-ZETA

[1]2111cs020573@mallareddyuniversity.ac.in [2] 2111cs020584@mallareddyuniversity.ac.in
[3]2111cs020589@mallareddyuniversity.ac.in [4] 2111cs020621@mallareddyuniversity.ac.in
[5]2111cs020626@mallareddyuniversity.ac.in

Department of Artificial Intelligence & Machine Learning

Malla Reddy University, Kompally, Hyderabad, India

I. ABSTRACT

Image Captioning is a task where each image must be understood properly and are able generate suitable caption with proper grammatical structure.Here it is a hybrid system which uses multilayer CNN (Convolutional Neural Network) for generating keywords which narrates given input images and Long Short Term Memory(LSTM) for precisely constructing the significant captions utilizing the obtained words .Convolution Neural Network (CNN) proven to be so effective that there is a way to get to any kind of estimating problem that includes image data as input. LSTM was developed to avoid the poor predictive problem which occurred while using traditional approaches. We used an encoder-decoder based model that is capable of generating grammatically correct captions for images. This model makes use of VGG16(Visual Geometry Group) as an encoder and LSTM as a decoder. The model will be trained like when an image is given model produces captions that almost describe the image. The efficiency is demonstrated for the given model using Flickr8K data sets which contains 8000 images and captions for each image but we use CNN and LSTM to capture dependencies and tell both the spatial relationships of images and contextual information of captions and generate contextually relevant captions.

Keywords—CNN(Convolutional Neural Network),LSTM(Long Short Term Memory),VGG16(Visual Geometry Group),Deep Learning,Encoder-Decoder.

2.INTRODUCTION

Creating precise descriptions for images has posed a significant challenge in the field of Artificial Intelligence, with a wide range of applications from robotic vision to assisting the visually impaired. The ultimate goal is to develop a system that can produce accurate and meaningful captions for images. Researchers have been exploring various methods to improve predictions, utilizing deep neural networks and machine learning techniques to construct effective models. By utilizing the Flickr 8k dataset, which consists of 8000 sample images each with five captions, we have focused on two key phases: extracting features from images using Convolutional Neural Networks (CNN) and generating natural language sentences based on the images using Recurrent Neural Networks (RNN). In the first phase, we have adopted a unique approach to

feature extraction that captures even subtle differences between similar images, rather than simply identifying objects. The VGG-16 model, with 16 convolutional layers, has been employed for object recognition. In the second phase, training our features with the dataset's captions is essential. We are implementing LSTM (Long Short Term Memory) architectures to formulate sentences based on the input images.

3.LITERATURE SURVEY

[1] This document introduces a model that utilizes pre-trained deep machine learning to generate captions for a given image. The VGG model is employed for image processing through deep convolutional neural networks. The caption is derived by comparing the model's output with actual human sentences. This comparison involves analyzing both the model's output and the captions provided by humans for the image. Based on this analysis, it is determined that the generated caption closely resembles the human-provided caption, resulting in an accuracy of approximately 75%. Therefore, the generated sentence and the human-provided sentence are highly similar.

[2] The image caption generator in this study was developed using the Flickr_8k database, which consists of a wide range of images depicting various situations. With a total of 8000 images, each picture is accompanied by 5 captions. The dataset is split into 6000 training images, 1000 validation images, and 1000 testing images. Through thorough training and testing, the model successfully generated accurate captions for the images. The proposed model utilizes a combination of Convolutional Neural Network and Recurrent Neural Network to assign appropriate labels and create grammatically correct captions, with CNN serving as the encoder and RNN as the decoder.

[3] In an image caption generator, the VGG16 model acts like a smart filter for pictures. It looks at an image and identifies important features like shapes and objects. These features help create a sort of summary of the picture. This summary, produced by VGG16, is then used by another part of the system to write a sentence describing what's happening in the image. So, VGG16 helps the system understand what's in the picture, and then the caption generator turns that understanding into words.

[4] This study introduces a framework that produces appropriate descriptions based on images. The framework utilizes the Flickr8K dataset, which includes 8000 images, each accompanied by five descriptions. The research showcases a model that employs a neural network to automatically analyze an image and create suitable English captions. The generated captions are categorized as follows: error-free descriptions, descriptions with minimal errors, descriptions somewhat related to the image, and descriptions unrelated to the image.

[5]The integration of VGG16, LSTM, and CNN in various applications, particularly in image caption generation, highlights the significance of combining these powerful components to achieve improved performance. The use of VGG16 as a robust feature extractor proves pivotal, capturing intricate details and semantic information from images. LSTM, with its ability to understand sequential data, complements VGG16 by effectively generating coherent and contextually rich captions. The synergy between CNN and LSTM addresses the challenges of combining visual and linguistic information for a more comprehensive understanding of images.While the surveyed literature demonstrates the effectiveness of this combination, ongoing research and advancements in deep learning may unveil novel architectures and methodologies, further refining the synergy between VGG16, LSTM, and CNN in image captioning and related tasks.

4. METHODOLOGY

Methodology focuses on developing an Image Caption Generator integrating VGG16 CNN for feature extraction and LSTM for caption generation

[1]Data Collection and Preprocessing:

Flickr8K Dataset: The Flickr8K dataset comprises 8,000 images, each paired with five descriptive captions, totaling 40,000 captions. These images cover a wide range of categories and scenes, providing diversity for training and evaluation.

Data Collection: The dataset was collected from the Flickr website, where users upload images with descriptive captions. Each image is associated with multiple captions provided by different users.

Preprocessing: Prior to training, the images were preprocessed by resizing them to a uniform size (e.g., 224x224 pixels) and normalizing the pixel values to a fixed range (e.g., [0, 1]). Captions were tokenized into

individual words, and a vocabulary was created by assigning a unique index to each word. Additionally, captions were padded or truncated to ensure uniform length for batch processing during training.

[2]Feature Extraction using VGG16:

VGG16 Architecture: VGG16, a widely-used convolutional neural network (CNN) architecture, was employed as a feature extractor. The model consists of 16 convolutional layers followed by three fully connected layers.

Pre-trained Weights: Pre-trained weights of the VGG16 model trained on the ImageNet dataset were utilized. The fully connected layers of VGG16 were removed, and only the convolutional layers were retained to extract image features while discarding unnecessary classification information.

Feature Extraction: Each image in the Flickr8K dataset was passed through the modified VGG16 network, and the activations from one of the intermediate convolutional layers were extracted. These activations served as high-level feature representations capturing the visual content of the images.

[3]Caption Preprocessing:

Tokenization: The captions associated with each image were tokenized into individual words or subword units to represent the textual content.

Vocabulary Creation: A vocabulary was constructed by compiling all unique words from the captions and assigning a unique index to each word.

Padding or Truncation: To ensure uniform length for captions, they were padded with a special token (e.g., <PAD>) or truncated to a maximum length. This step facilitated batch processing during training.

[4]Model Architecture:

Sequence-to-Sequence Architecture: The model architecture comprised an encoder-decoder framework, with VGG16 serving as the encoder and an LSTM network serving as the decoder.

Integration of VGG16 Features: The features extracted by VGG16 were fed into the initial state of the LSTM decoder, enabling the model to generate captions conditioned on the visual content of the images.

[5]Training:

Dataset Splitting: The Flickr8K dataset was divided into training, validation, and test sets, typically with a standard split of 6,000 training images, 1,000 validation images, and 1,000 test images.

Training Parameters: Training parameters such as batch size, learning rate, and number of epochs were selected through experimentation and hyperparameter tuning.

Optimization: The model was trained using an optimization algorithm such as Adam, with the cross-entropy loss function used to measure the discrepancy between predicted and ground truth captions.

Teacher Forcing: During training, teacher forcing was employed to facilitate learning by feeding the ground truth previous word as input to predict the next word.

[6]Evaluation:

Evaluation Metrics: Evaluation of the trained model was performed using standard metrics such as BLEU score, METEOR, and CIDEr, which measure the similarity between generated captions and human-written references.

Dataset Split: The evaluation was conducted on the validation and test sets of the Flickr8K dataset to assess the generalization performance of the model.

Comparison: The performance of the model was compared with baseline methods or previous state-of-the-art approaches to validate its effectiveness.

[7]Inference:

Caption Generation: For inference, unseen images were passed through the trained VGG16 network to extract features, which were then fed into the LSTM decoder to generate captions.

Example: An example inference pipeline was provided, demonstrating how the model generated captions for new images from the Flickr8K dataset.

[8]Fine-tuning:

Domain-specific Fine-tuning: Fine-tuning of the model on the Flickr8K dataset itself or related datasets could be performed to adapt the model to specific domains or improve performance on similar datasets.

Benefits and Challenges: The potential benefits and challenges of fine-tuning were discussed, including the need for additional annotated data and the risk of overfitting.

5. RESULTS

```
# train the model
epochs = 50
batch_size = 32
steps = len(train) // batch_size

for i in range(epochs):
    # create data generator
    generator = data_generator(train, mapping, features, tokenizer, max_length, vocab_size, batch_size)
    # fit for one epoch
    model.fit(generator, epochs=1, steps_per_epoch=steps, verbose=1)
```

```
227/227 [==============================] - 66s 290ms/step - loss: 2.1551
227/227 [==============================] - 63s 277ms/step - loss: 2.1334
227/227 [==============================] - 72s 315ms/step - loss: 2.1076
227/227 [==============================] - 63s 275ms/step - loss: 2.0876
227/227 [==============================] - 63s 277ms/step - loss: 2.0660
227/227 [==============================] - 83s 364ms/step - loss: 2.0445
227/227 [==============================] - 71s 314ms/step - loss: 2.0291
227/227 [==============================] - 73s 320ms/step - loss: 2.0088
227/227 [==============================] - 71s 315ms/step - loss: 1.9924
227/227 [==============================] - 71s 313ms/step - loss: 1.9730
227/227 [==============================] - 61s 270ms/step - loss: 1.9538
227/227 [==============================] - 62s 275ms/step - loss: 1.9359
227/227 [==============================] - 62s 274ms/step - loss: 1.9200
227/227 [==============================] - 62s 273ms/step - loss: 1.9052
227/227 [==============================] - 62s 273ms/step - loss: 1.8885
227/227 [==============================] - 62s 273ms/step - loss: 1.8739
227/227 [==============================] - 61s 270ms/step - loss: 1.8609
227/227 [==============================] - 64s 280ms/step - loss: 1.8471
227/227 [==============================] - 61s 268ms/step - loss: 1.8320
227/227 [==============================] - 64s 283ms/step - loss: 1.8163
227/227 [==============================] - 64s 281ms/step - loss: 1.8077
227/227 [==============================] - 64s 284ms/step - loss: 1.7946
227/227 [==============================] - 61s 270ms/step - loss: 1.7836
 55/227 [======>.......................] - ETA: 53s - loss: 1.7749
```

```
from nltk.translate.bleu_score import corpus_bleu
# validate with test data
actual, predicted = list(), list()

for key in tqdm(test):
    # get actual caption
    captions = mapping[key]
    # predict the caption for image
    y_pred = predict_caption(model, features[key], tokenizer, max_length)
    # split into words
    actual_captions = [caption.split() for caption in captions]
    y_pred = y_pred.split()
    # append to the list
    actual.append(actual_captions)
    predicted.append(y_pred)

# calcuate BLEU score
print("BLEU-1: %f" % corpus_bleu(actual, predicted, weights=(1.0, 0, 0, 0)))
print("BLEU-2: %f" % corpus_bleu(actual, predicted, weights=(0.5, 0.5, 0, 0)))
```

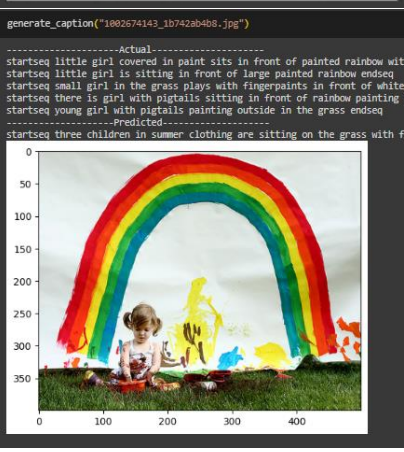
```
100% 810/810 [11:00<00:00, 1.43it/s]
BLEU-1: 0.533333
BLEU-2: 0.306618
```

```
generate_caption("1001773457_577c3a7d70.jpg")

--------------------Actual--------------------
startseq black dog and spotted dog are fighting endseq
startseq black dog and tri-colored dog playing with each other on the road endseq
startseq black dog and white dog with brown spots are staring at each other in the street endseq
startseq two dogs of different breeds looking at each other on the road endseq
startseq two dogs on pavement moving toward each other endseq
--------------------Predicted--------------------
startseq two dogs playing with each other endseq
```

```
generate_caption("1002674143_1b742ab4b8.jpg")

--------------------Actual--------------------
startseq little girl covered in paint sits in front of painted rainbow with her hands in bowl endseq
startseq little girl is sitting in front of large painted rainbow endseq
startseq small girl in the grass plays with fingerpaints in front of white canvas with rainbow on it endseq
startseq there is girl with pigtails sitting in front of rainbow painting endseq
startseq young girl with pigtails painting outside in the grass endseq
--------------------Predicted--------------------
startseq three children in summer clothing are sitting on the grass with fingerpaints endseq
```



6. DRAWBACK

Limited Dataset Size: Acknowledge the relatively small size of the Flickr8K dataset compared to larger datasets like MSCOCO, potentially limiting the model's ability to generalize to a broader range of images and captions.

Data Imbalance and Bias: Recognize potential biases or imbalances within the Flickr8K dataset, which may affect model performance and generalization. Discuss strategies to mitigate bias and ensure equitable representation across diverse image categories and caption styles.

Complexity of Model Interpretability: Highlight the challenge of interpreting and explaining the inner workings of the VGG16-LSTM model, particularly regarding how specific image features influence caption generation. Discuss the importance of model interpretability for practical applications and avenues for improving transparency

7. CONCLUSION

In conclusion, the proposed methodology leveraging VGG16 and LSTM for Image Caption Generation demonstrates promising results on the Flickr8K dataset, showcasing the efficacy of the approach in generating descriptive and contextually relevant captions. Despite certain drawbacks, such as dataset limitations and computational complexity, the methodology serves as a valuable foundation for advancing research in computer vision and natural language processing. Future work should focus on addressing these limitations and exploring innovative techniques to enhance model performance, scalability, and interpretability in diverse application domains. Overall, the findings presented herein contribute to the ongoing dialogue surrounding multimodal understanding and intelligent image captioning systems.

8. REFERENCES

Referred the below in internet:

[1]Dr.Jagadisha N ,Chaithra Rao(2022):Image Caption Generator Using Deep Learning

[1]Sreejith S P, Vijayakumar A (2021) : Image Captioning Generator using Deep Machine Learning.

[2] Preksha Khant, Vishal Deshmukh, Aishwarya Kude, Prachi Kiraula (2021) : Image Caption Generator using CNN-LSTM.

[3] Ali Ashraf Mohamed (2020) : Image Caption Using CNN and LSTM.

[4] Chetan Amritkar, Vaishali Jabade (2018) : Image caption Generation Using Deep Learning Technique.

[5] Subrata Das, Lalit jain, Arup Das (2018) : Deep Learning for Military Image Captioning.

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