Journal of
Data Acquisition and Processing

ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.777626

IMAGE CAPTION GENERATION USING DEEP LEARNING - CNN AND LSTM APPROACH

Narayanamoorthy M[1]* Haadhim Mubarak Ali[2], Himanshu Kurrey[3], Sudhanshu Amarendra Singh[4]

[1] Department of IoT, School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology (VIT), Vellore, TamilNadu, India

[2] School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology (VIT),Vellore, TamilNadu, India.

[3] School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology (VIT), Vellore, Tamilnadu, India.

[4] School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology (VIT), Vellore, Tamilnadu, India.

Email:mnarayanamoorthy@vit.ac.in ; haadhimmubarak.ali2022@vitstudent.ac.in ; himanshu.kurrey2022@vitstudent.ac.in ; sudhanshu.amarendra2022@vitstudent.ac.in

- Corresponding author

***Abstract*:** Image Caption Generation entails the creation of natural language descriptions for images. Due to the complexity of the visual content as well as the semantic details of the corresponding natural language, a problem arises. In this project, we propose a system for generating image captions by integrating the architecture of Convolutional-Neural-Networks along with Long-Short-Term Memory networks. The methodology involves the CNN gathering of visual features followed by the LSTM generation of captions. We also incorporate attention mechanisms to enhance the performance of the model by enabling it to concentrate on pertinent visual features while generating captions. Using CIDEr, METEOR and BLEU scores, for evaluation of the performance of our modified and restructured model using standard benchmark datasets (Flickr8k) and BLEU, CIDEr, and METEOR scores. Our proposed system has the potential for use in the image retrieval, the image description, and other multimedia applications requiring image analysis and natural language processing.

Keywords: *Convolutional-Neural-Network, BLEU, CIDEr and METEOR score, Long-Short-Term-Memory, Image Captioning.*

INTRODUCTION

Image Caption Generation is an interesting and difficult problem involving the generation of description of images. Recently, the task has gained enormous focus in numerous disciplines, including image retrieval, image description, and multimedia applications. This endeavor is difficult due to the complexity of the visual content and the semantic nuances of NL (natural language).

In recent years, deep learning techniques, especially CNNs for the extraction of image features and LSTM networks for extraction of caption features, have been employed to address this issue. Due to their capacity to acquire hierarchical representations of visual content, CNNs are extensively considered for image feature-extraction. Similarly, LSTMs are effective at generating sequences of natural language, preparing them for the caption generation. Incorporating attention mechanisms is also a popular method for enhanced quality for caption generation, as it enables the model to concentrate on pertinent visual features while generating the captions.

In this paper, we present an modified and enhanced architecture to produce captions for images that incorporates CNN and LSTM networks for visual feature extraction and caption generation, respectively. In addition, we incorporate attention mechanisms to enhance the model's performance. We demonstrate the state-of-the-art efficacy of our suggested system by evaluating it against standard benchmark datasets.

We compare our model’s performance to that of various cutting-edge methodologies and demonstrate its superior performance among others in CIDEr, METEOR and BLEU. In addition, we conduct the procedure experiments to determine the impact of our system's numerous components.

LITERATURE REVIEW

Tan and Chan's caption generation model uses a CNN that extracts relevant features from the images input to it, which then creates captions using hierarchical LSTM. Initially, their model finds phrases instead of individual words, and captions based on those phrases. [1]. Fang et al.'s method for image captioning combines a CNN with an attention-based LSTM network. The CNN extracts visual information in the form of features from the input picture, while the attention-based LSTM network produces captions by working on different features in the image one at a time.[2]. Maroju et al.'s image caption generating model employs a combination of a CNN and an LSTM network. LSTM generates captions from the features extracted through CNN.[4]. Phukan and Panda's efficient technique for captioning of an image to extract the features using CNN, and generates captions corresponding to each individual features using LSTM. Their model also incorporates a beam search algorithm to enhance the quality of the captions generated.[3]. Anderson et al.'s model employs a bottom-up attention mechanism to identify salient image regions, which are then combined with a top-down attention mechanism to generate captions.[5]. Long et al.'s DeepCaption framework uses a two-stage attention mechanism to attend to prominent sections of the image and generated words during caption generation. Their model also incorporates a consistency loss term to encourage coherence between the image and its corresponding caption.[6]. Lu et al.'s adaptable attention mechanism employs a visual sentinel to tell the model when to stop attending to the image and attend to the previous word instead. This leads to improved performance by helping the model emphasize the most important aspects of the input.[7]. Faghri et al.'s model attends to both image and text representations at multiple levels of abstraction for appropriate image-caption matching. By capturing association among different objects with varied shape, attributes and other features; the performance of their system is enhanced.[8]. Hendricks et al.'s model for generating visual explanations generates captions that highlight the important regions and features in the input image. Their model incorporates a visual attention mechanism and a language model to produce natural language descriptions of the image.[9]. Gan et al.'s compositional network for picture captioning models associating between objects, attributes, and relationships in an image using compositional operations. The model attends to different sections of the image and generates captions based on the composition of these parts. [10]

METHODOLOGY

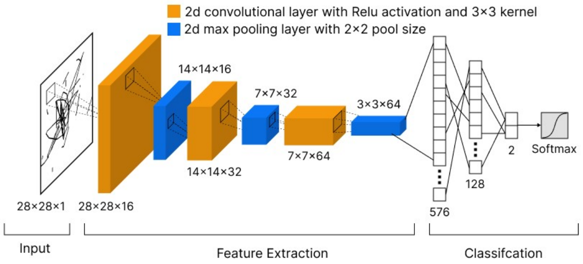

Fig 1:Convolution-Neural-Network Model

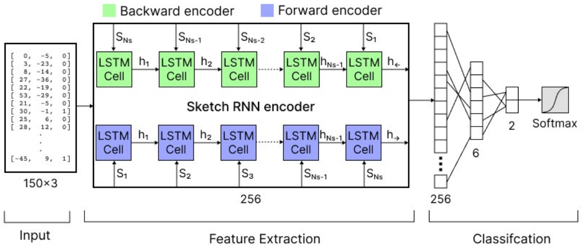

Fig 2: Long-Short-Term-Memory Model

[Fig-1] Convolutional Neural Networks (CNN): are utilized predominantly for processing grid-like data, especially images. They are designed to acquire hierarchical structure and features automatically from input data. The layers of CNN architecture include convolutional, pooling and fully connected layers.

[Fig-2] Long Short-Term Memory (LSTM): a form of RNN-architecture to process data in sequential format.

The methodology for building an image caption generator typically involves the following steps:

1. *Dataset Preparation:* Obtain an appropriate dataset consisting of associated images and captions. Normalize the pixel values and resize the images to a fixed dimension prior to processing. Create a vocabulary mapping for the words that result from tokenizing the captions into words or sub-words.

2. *Image Feature Extraction*: Use CNN that has been previously trained to extract meaningful image features. Remove the final layer of classification from CNN and record the output from the layer preceding it. The output of this layer represents the image characteristics that can be input into the captioning model.

3. *Text Data Preprocessing*: Process the captions by tokenizing them into individual words or sub words. Create the vocabulary by mapping each unique word to a numerical index. Pad or truncate the captions to a fixed length to ensure consistent input size for the captioning model.
4. *Model Architecture:* Design the architecture of the model for captioning. Combining CNN along with LSTM is a common technique. CNN is responsible for extracting image features, while LSTM processes the sequential character of the captions and generates the output captions.
5. *Model Training*: Use the prepared dataset to train the captioning model. The extracted image features from step 2 are supplied into the CNN portion of the model, while the captions are processed by the LSTM portion. Minimizing a loss function: cross-entropy loss, enables the model to generate captions. Utilizing backpropagation and an optimizer, such as Adam or RMSprop, update the model's parameters.
6. *Caption Generation*: Following training, the model is able to utilize to create new image captions. Using the trained CNN, extract the features of an image input. Initiate the LSTM with a special start token and generate the following word by repeatedly feeding the prior word and image features into the LSTM. Repeat this procedure until a token denoting the sentence end is generated or the max length has been exceeded.
7. *Evaluation*: Evaluate the generated captions using suitable metrics: CIDEr, METEOR and BLEU, compares the captions generated with reference captions from the dataset to assess their quality and similarity.
8. *Fine-tuning and Improvements:* Experiment with various architectural variants, hyperparameter tuning, and regularization strategies to enhance the performance of the model.

By adhering to these steps, you can create an image caption generator capable of generating descriptive and meaningful captions for an input image.

PROPOSED ARCHITECTURE

The neural network architecture includes two main components: an image feature processing network and a text processing network, followed by a encoding/merging and decoding section. The model’s networks that processes images takes 1D embeddings as input with size 4096 and To avoid overfitting, a regularization layer is performed with a 50% dropout. Then, a dense layer brings down the size of the input from 4096 to 512. The output of this layer is then forwarded to the subsequent layers.

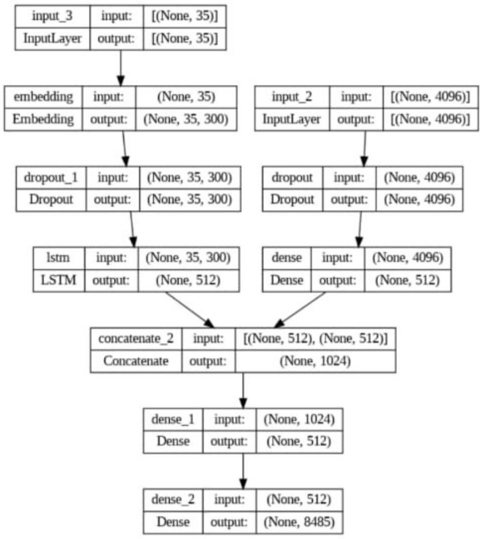
input_3 | input: | [(None, 35)]
InputLayer | output: | [(None, 35)]
embedding | input: | (None, 35)
Embedding | output: | (None, 35, 300)
input_2 | input: | [(None, 4096)]
InputLayer | output: | [(None, 4096)]
dropout_1 | input: | (None, 35, 300)
Dropout | output: | (None, 35, 300)
dropout | input: | (None, 4096)
Dropout | output: | (None, 4096)
lstm | input: | (None, 35, 300)
LSTM | output: | (None, 512)
dense | input: | (None, 4096)
Dense | output: | (None, 512)
concatenate_2 | input: | [(None, 512), (None, 512)]
Concatenate | output: | (None, 1024)
dense_1 | input: | (None, 1024)
Dense | output: | (None, 512)
dense_2 | input: | (None, 512)
Dense | output: | (None, 8485)

Fig 3:X-Architecture

The text preparing section of the model operates on a 1D text input vector of size 35. It starts with an embedding layer using which word embeddings are obtained from the words. Similarly, a regularization layer provided with a dropout rate of 50% is employed to prevent overfitting. After dropout, the input is forwarded to a LSTM network with similar size, enabling the generation of richly semantic and descriptive sentences for an image.

Once the results from the image and text networks are obtained, the merging and decoding section comes into play. It incorporates a supplementary layer that combines the outputs of the image and text networks. Subsequently, a dense 1D layer with a dimension of 512 and 'relu' activation function is applied after the merging. Finally, with the same dimension a fourth dense layer and softmax activation is added to finally produce probabilities used for generating descriptive captions.

In summary, the architecture employs an image feature capturing and processing network, a text processing network, and a merging and decoding section that produces image captions collectively in a descriptive manner.

RESULT AND DISCUSSION

Performance Metrics

1. The Bilingual Evaluation Understudy-N [BLEU-N]: metric evaluates machine translations by comparing them to reference translations. BLEU-N compares the translated n-grams to the reference(s). BLEU-N scores range from 0 to 1 and reflect how well an n-gram in the translation matches the reference(s). BLEU-1/2/3/4 is usually computed to evaluate translation quality.
2. Metric for Evaluation of Translation with Explicit Ordering [METEOR]: compares produced captions to reference captions to assess quality. Aligning n-grams (contiguous sequences of n words) between produced and reference captions emphasizes precision as well as recall.

3. Consensus-based Image Description Evaluation [CIDEr]: evaluates caption quality. It captures picture reference caption consensus. Generated captions that match consensus descriptions from human annotators score better.

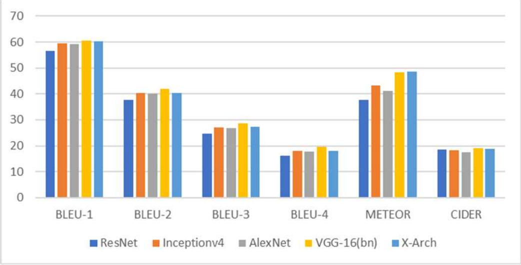

Fig 4: Performance Graph

CNN name	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr
ResNet	56.5	37.8	24.6	16.2	37.7	18.5
Inceptionv4	59.49	40.47	27	18.03	43.17	18.22
AlexNet	59.24	40.17	26.82	17.87	41.09	17.51
VGG-16(bn)	60.56	41.98	28.66	19.51	48.41	19.04
X-Arch	60.16	40.36	27.21	18.1	48.56	18.7

Fig 5: Performance evaluation of different architecture

CONCLUSION

Image caption generators are a powerful tool for automatically generating textual descriptions of images, bridging the gap between visual and textual information, and enabling machines to understand and communicate about visual content. This research gives an insight to achieve higher accuracy and performance with revised model. Further, the model's performance is assessed through BLEU, CIDEr and METEOR scores as evaluation metrics.

ACKNOWLEDGEMENT

We wish to express our sincerest gratitude to Dr. Narayanamoorthi M, whose invaluable guidance has been instrumental in our project's success. We extend our deepest appreciation to the instructors of Vellore Institute of Technology's Department of IoT for sharing their expertise and teachings that have significantly contributed to our progress. We are equally thankful for the unwavering support and provision of essential resources from the university administration. Lastly, we would like to acknowledge and express our gratitude to everyone who aided and motivated us throughout this project.

REFERENCES

[1] Tan, Y. H., & Chan, C. S. (2019). Phrase-based image caption generator with hierarchical LSTM network. Neurocomputing, 333, 86-100.

[2] Fang, F., Wang, H., Chen, Y., & Tang, P. (2018). Looking deeper and transferring attention for image captioning. Multimedia Tools and Applications, 77, 31159-31175.

[3] Phukan, B. B., & Panda, A. R. (2021). An efficient technique for image captioning using deep neural network. In Cognitive Informatics and Soft Computing: Proceeding of CISC 2020 (pp. 481-491). Springer Singapore.

[4] Maroju, A., Doma, S. S., & Chandarlapati, L. (2021). Image Caption Generating Deep Learning Model. INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH TECHNOLOGY (IJERT), 10(09).

[5] Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., & Zhang, L. (2018). Bottom-up and top-down attention for image captioning and visual question answering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 6077-6086).

[6] Long, Q., Wang, T., Huang, G., & Shen, W. (2018). DeepCaption: A Deep Learning-based Image Captioning Framework. IEEE Transactions on Multimedia, 20(8), 2037-2050.

[7] Lu, J., Xiong, C., Parikh, D., & Socher, R. (2017). Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(12), 2517-2529.

[8] Faghri, F., Fleet, D. J., Kiros, J. R., & Fidler, S. (2018). Stacked Cross Attention for Image-Text Matching. IEEE Transactions on Pattern Analysis and Machine Intelligence, 41(6), 1409-1422.

[9] Hendricks, L. A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., & Darrell, T. (2018). Generating Visual Explanations. International Journal of Computer Vision, 126(2-4), 221-235.

[10] Gan, Z., Gan, C., He, X., Pu, Y., Tran, K., Gao, J., & Carin, L. (2017). Semantic Compositional Networks for Visual Captioning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(11), 2075-2089.