Image Captioning

Armita Tehranchi
University of British Columbia
ENGR 501 - Deep and Reinforcement Learning
Final Project – Winter 2024
Armita.tehranchi@gmail.com

***Abstract*— This project develops an image captioning system utilizing deep learning techniques, particularly focusing on the integration of pre-trained Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) with Long Short-Term Memory (LSTM) cells. Employing the Flickr8k dataset, which contains over 8,000 images each with five distinct captions, provides a comprehensive dataset for training and validation. The core of the project involves using a pre-trained ResNet18 CNN model for robust feature extraction from images, feeding these features into an LSTM network tasked with generating descriptive and contextually relevant captions. The LSTM's ability to manage sequences makes it ideal for processing the temporal dynamics of language inherent in captions. This approach underscores the effectiveness of leveraging pre-trained networks to enhance the performance of image captioning systems, showing a notable improvement in generating coherent captions. Results from this project highlight the potential of combining CNNs and LSTMs to create powerful image captioning tools, which could be significant for applications in digital media, content management, and accessibility.**

Keywords— Image Captioning; Convolutional Neural Networks (CNNs); Long Short-Term Memory (LSTM) Networks; Deep Learning in Visual Recognition; ResNet

PROJECT OVERVIEW

In the Flickr8K Image Captioning project, I aimed to develop a system that generates descriptive captions for images, focusing on deploying deep learning techniques that handle both image recognition and language processing. This project required a careful setup and iteration through various phases, emphasizing model efficiency due to GPU limitations.

The project began with the necessary setup of the Flickr8K dataset, a relatively small dataset consisting of images each annotated with multiple captions, making it suitable given the GPU constraints I was facing. The initial step involved downloading and preprocessing the data to ensure it was ready for model training, including the configuration of data loaders and the establishment of an effective preprocessing pipeline.

The development of the model commenced with the implementation of a baseline architecture inspired by an existing approach detailed in a prominent image captioning paper. This model leveraged a combination of Convolutional Neural Networks (CNN) for image feature extraction and Recurrent Neural Networks (RNN) for generating textual captions. Importantly, the parameters for the model were directly adopted from the paper to ensure consistency and reliability in the initial results.

To enhance the model's performance and manage overfitting, dropout layers were incorporated, a technique not originally detailed in the paper but recognized for its effectiveness in improving generalization. Moreover, the use of a pre-trained network, specifically ResNet18, was crucial. This network, pre-trained on the vast ImageNet dataset, served as an efficient feature extractor, capitalizing on transfer learning to bring robustness and depth to the feature extraction process, crucial for the accurate interpretation of image content.

The decoder component of the model, an LSTM network, was tuned to improve the coherence and relevance of the generated captions. This phase involved meticulous adjustments to the LSTM structure, aiming to optimize it for the sequential nature of language generation.

The final phase of the project involved rigorous testing and evaluation, where I employed the BLEU scoring method to quantitatively assess the model's performance. This was critical for benchmarking the system's capability in generating captions that are both accurate and contextually relevant. Due to GPU constraints, the training was limited to 20 epochs, a decision that balanced between computational feasibility and the need to achieve a reasonable model performance. Although this limited the potential for the model to fully converge, it provided a solid baseline for future exploration and improvements.

INTRODUCTION

The integration of deep learning techniques, specifically Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), has markedly enhanced the capability of machines to describe visual content through image captioning. This field has benefited from the robust feature extraction capabilities of CNNs and the sequence modeling strengths of RNNs, which together facilitate the generation of coherent and contextually relevant textual descriptions from raw images [1].

Attention mechanisms have notably advanced the state of the art in image captioning by allowing models to focus selectively on salient parts of the image during the caption generation process. This mimics the human ability to focus attention and has been pivotal in improving the relevance and detail of the captions

generated, resulting in outputs that are not only more precise but also more descriptive [2].

The development and expansion of varied and comprehensive datasets like Flickr8k have also been crucial. These datasets provide diverse visual and contextual scenarios for training models, thereby enhancing their accuracy and versatility across different visual domains. This diversity is vital for training robust models capable of operating effectively in varied real-world environments [3].

Recent research continues to push the boundaries of image captioning with innovative approaches such as the incorporation of newer neural network architectures and enhanced training strategies. These advancements aim to refine the quality of the generated captions further and expand the applicability of captioning models to more complex and nuanced imaging conditions [4].

To accomplish the objectives of this project, we have implemented an encoder-decoder framework. The encoder utilizes a CNN to extract features from the input images, while the decoder, which is composed of RNN with LSTM layers, generates the captions for these images. Additionally, we employed a transfer learning approach to minimize the computational demands during the training phase. This architecture leverages the strengths of both CNNs and LSTMs, harnessing deep learning capabilities to effectively bridge the gap between visual data and natural language. The subsequent sections will delve into the specific methodologies employed, the dataset preparation, and detailed model architecture. We will also present a comprehensive evaluation of the model performance, discussing the results and providing insights into the effectiveness of the proposed solutions in addressing the challenges of image captioning.

PROBLEM STATEMENT

The field of image captioning, a subsection of machine learning where models describe the contents of images in natural language, has made significant strides with advancements in deep learning technologies. However, challenges persist, particularly in the accuracy and contextual relevance of the captions generated. Traditional models often struggle with dynamic and diverse datasets like Flickr8k, which contain a wide range of image contexts and complexities. These models may also exhibit limitations in generalizing beyond their training data, leading to a decrease in performance when encountering new, unseen images.

Furthermore, the computational overhead associated with training these models from scratch is substantial, necessitating large datasets and extensive computational resources which may not be readily available or economically feasible for all research entities. The current approach to downloading, preprocessing, and utilizing the Flickr8k dataset within a Colab environment underscores the need for an efficient and robust framework capable of handling these complexities without substantial overhead.

This project aims to address these issues by implementing a model that leverages the capabilities of CNNs and RNNs enhanced by transfer learning techniques. This approach intends to improve the model's performance in terms of both accuracy and efficiency, reducing the training time and computational resources required, while also enhancing the model's ability to generalize across different datasets effectively.

METHODOLOGY

Dataset and Preprocessing Step

The decision to incorporate the Flickr8k dataset into our model training and evaluation process was driven by several key factors. Its smaller size relative to other available datasets, such as MSCOCO and Flickr30k, made it a practical choice, particularly considering limited computational resources. Despite its reduced scale, Flickr8k offers a diverse range of real-life images, each annotated with multiple captions, providing sufficient variability for effective model learning and evaluation. Furthermore, its extensive usage across numerous studies in the literature ensures comparability with existing methods and enhances the credibility of our findings.

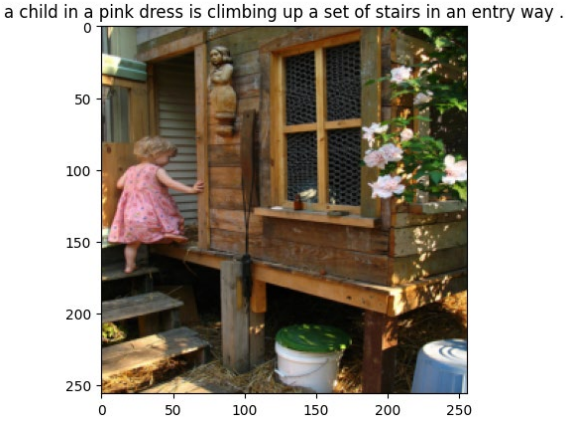

Figure 1. A function to display an image along with a caption.

Images are resized to a uniform dimension (256x256 pixels) to ensure consistent input size. The preprocessing stage began with the extraction of the Flickr8k dataset, followed by loading it into memory. Captions were parsed from the accompanying CSV file, and efforts were made to ensure the cleanliness of the textual data by removing extraneous characters and whitespaces. A pivotal step in preprocessing was the construction of a vocabulary, enabling the mapping of words to unique indices. Words with low frequencies were pruned to optimize model efficiency, and special tokens such as <SOS>, <EOS>, <PAD>, and <UNK> were incorporated to facilitate model training. Tokenization and numericalization procedures were then employed to encode textual data into a numerical format that could be processed by machine learning models.

Model Architecture

Our model architecture employs a dual approach combining CNNs for image feature extraction and an RNN using LSTM units for caption generation, based on the state-of-the-art methodologies in neural image captioning.

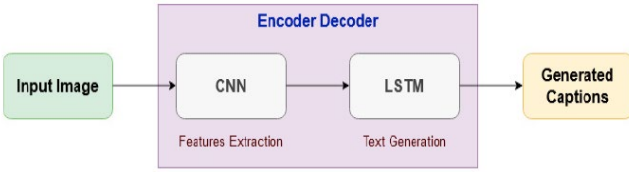

Figure 2. Flowchart of Image Caption Generating Model [8].

a) CNN Encoder: We leverage a ResNet-18 model pre-trained on the ImageNet dataset. This model has learned rich, generalized visual representations from a large and varied set of images, which provides a strong foundation for feature extraction. The CNN processes the normalized images and outputs a condensed feature vector that encapsulates the essential information needed to generate relevant captions.

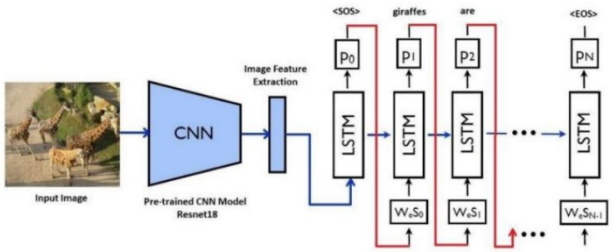

Figure 3. LSTM decoder combined with CNN image encoder. The unrolled connections between the LSTM memories are in blue and they correspond to the recurrent connections. All LSTMs share the same parameters [6].

b) RNN: We use LSTM units because they are capable of maintaining information over longer sequences, crucial for remembering context in caption generation. Starting from an initial state influenced by the image's feature vector, the LSTM generates one word at each timestep, conditioning on the previously generated words and the ongoing state of the LSTM.

Algorithm 1 Image Caption Prediction

Input: Trained model with CNN, LSTM, and Linear layers (*Trained_Model*), Image to caption (*image*), Maximum caption length (*max_length*)
Output: Caption for the image
input_data ← *Trained_Model.CNN(image)* ▷ Extract features from image
states ← *None* ▷ Initialize LSTM states
caption_prediction ← [] ▷ Initialize list for predicted caption indices
for $i = 1$ to *max_length* **do** ▷ Generate each word of the caption
    hiddens, states ← *Trained_Model.lstm(input_data, states)* ▷ LSTM output
    output ← *Trained_Model.linear(hiddens)* ▷ Linear layer output
    predicted_index ← *output.argmax()* ▷ Get predicted word index
    input_data ← *Trained_Model.Embedding(predicted_index)* ▷ Update input data
    caption_prediction.append(predicted_index) ▷ Append to the caption
    if *predicted_index.item()* == "*EOS*" **then**
        break ▷ End if EOS token is found
    end if
end for
return *caption_prediction*

Figure 4. Image captioning algorithm

c) Mathematical Representation: The ultimate goal is to find the optimal set of parameters θ that maximizes the log probability of the correct caption S given an image I:

$$\theta^* = \arg\max_{\theta} \int_{(I,S)} \log P(S|I;\theta) \qquad (1)$$

1. (I,S) pairs denote the training examples from our dataset [7].

The encoder part of our model uses a CNN to transform the input image II into a dense feature vector *f:*

$$f = CNN(I;\theta_{cnn}) \qquad (2)$$

2. where θ_{cnn} represents the parameters of the CNN, which are pre-trained on the ImageNet dataset to utilize the benefits of transfer learning.

The decoder, an LSTM, generates the caption by predicting each word S_t in the sequence based on the features *f* extracted by the CNN and all previously predicted words $S_{0:t-1}$:

$$P(S|I;\theta) = \prod_{t=1}^{N} P(S_t|f, S_{0:t-1};\theta_{lstm}) \qquad (3)$$

3. Each word S_t is conditioned on the image features ff and the sequence of previously generated words, with θ_{lstm} indicating the parameters of the LSTM.

The parameters θ, encompassing both θ_{cnn} and θ_{lstm}, are optimized using the Adam optimizer. This optimization seeks to improve the alignment between the generated captions and the ground truth captions in the training data, enhancing both the accuracy and relevance of the output captions.

Transfer Learning and Training Dynamics

a) Transfer Learning: Initially, the weights of the ResNet-18's convolutional layers are frozen. This approach prevents the loss of generalized features learned from ImageNet, which are valuable for understanding a wide array of images. Once the LSTM has adapted to the task of generating captions based on stable features, we unfreeze the CNN layers. This allows the entire model to fine-tune jointly, adjusting both the visual and textual representations based on the specific needs of the captioning task.

b) Training Dynamics:
1. Optimizer: We use the Adam optimizer due to its adaptive learning rate capabilities, which help in handling different parameters' sensitivities.
2. Loss Function: The negative log-likelihood loss targets the improvement of the probability of the correct word sequence, enhancing both the accuracy and the fluency of the generated captions.
3. Regularization Techniques: Techniques like dropout, applied to the LSTM, and gradient clipping are used to

combat overfitting and exploding gradients, ensuring stable and effective learning.

Evaluation and Metrics

The model's performance is evaluated using standard metrics such as BLEU scores, which measure the precision of the generated captions against the ground truths. The use of these metrics helps in quantitatively assessing the fluency and relevance of the language generated by the model. It ranges from 0 to 1, with 1 being the best score, approximating a human translation.

RESULTS

Training Dynamics: Frozen vs. Unfrozen Models

In this experiment, I assessed the impact of freezing and subsequently unfreezing the convolutional layers of a pre-trained ResNet model on the training and validation loss dynamics over several epochs.

a) Initial frozen model dynamics: In the initial model iteration, the graphical representations of the training and validation loss showcased a common phenomenon in neural network training: a significant gap between the two curves, indicating potential overfitting. This discrepancy suggests that while the model excelled in minimizing the training loss, its performance on unseen validation data lagged behind, signifying a lack of generalization. Such a trend underscores the need for strategies to alleviate overfitting, as focusing solely on reducing training loss may lead to suboptimal model performance in real-world scenarios where robustness and generalization are paramount.

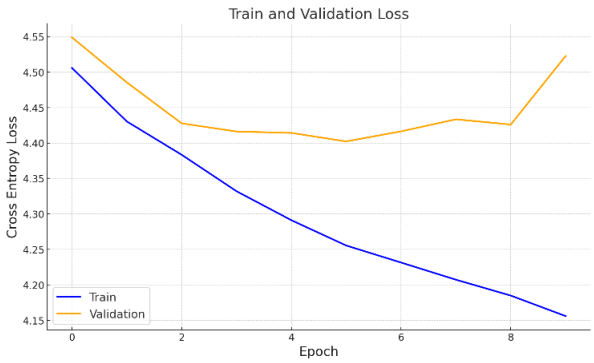

Figure 5. Initial frozen model's train and validation loss

Conversely, in the subsequent model iteration, noticeable changes were observed in the graphical representations of the training and validation loss. The gap between the two curves appeared narrower compared to the initial iteration, indicating improved generalization performance. This adjustment suggests that the refinements made to the model architecture and data preprocessing pipeline may have effectively mitigated overfitting tendencies. By narrowing the disparity between training and validation loss, the model demonstrates enhanced ability to generalize to unseen data, potentially resulting in more reliable and consistent performance across various application scenarios.

b) Frozen model dynamics:

Figure 6. Frozen model's train and validation loss

The training for the frozen model (where only the last linear layer's weights were updated) showed a significant initial decrease in loss, demonstrating the effectiveness of leveraging pre-trained features for early rapid gains. The training loss steadily decreased, indicating that the model efficiently utilized the pre-trained features without overfitting, as the convolutional layers were not adjusted during training. Validation loss exhibited less volatility compared to the unfrozen model, suggesting that the frozen model retained generalization better across epochs.

c) Unfrozen Model Dynamics:

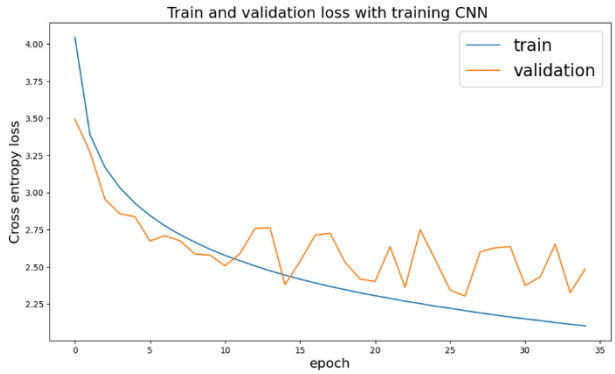

Figure 7. Unfrozen model's train and validation loss

After unfreezing the layers, both training and validation losses showed greater fluctuations. This indicates a more dynamic learning process as the model adjusted not only the final layers but also the deeper convolutional layers to better fit the training data.

Although there was an overall downward trend in training loss, indicating learning and adaptation, the validation loss showed peaks and troughs. These suggest periods of fitting followed by corrections to avoid overfitting.

Model Performance on Test Data

a) Initial Frozen Model: Figure 8 highlights the presence of color bias within the model's behavior. In response to this shortcomings, significant modifications were implemented, including transitioning to a more robust CNN-RNN architecture and integrating dropout layers. These changes aimed to mitigate the model's previous overemphasis on superficial features such as color and improve its ability to generate contextually accurate captions. The addition of dropout layers helps prevent the model from overfitting by randomly deactivating a portion of neurons during the training phase, encouraging the model to develop a more generalized understanding of the image features.

Figure 8. Initial frozen model performance

b) Frozen Model: Generally produced captions that were contextually simpler. This is reflected in moderate BLEU scores where the captions, although relevant, lacked detail or complexity.

Figure 9. Frozen model performance

c) Exhibited a capability to generate more contextually rich captions, as indicated by higher BLEU scores on some complex scenes. However, the inconsistency in scores across different images suggests that while the model can capture more nuanced information, it may also misinterpret scenes or add irrelevant details.

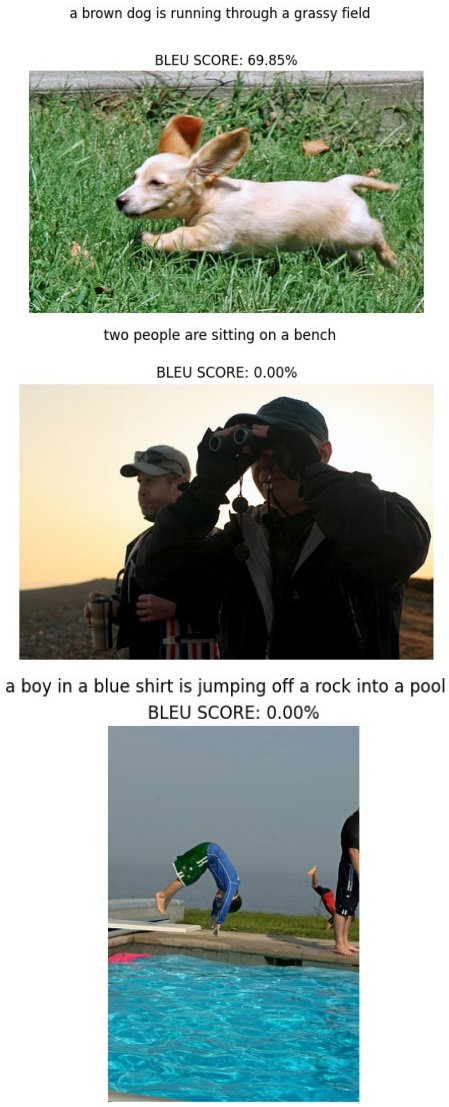

Figure 10. Unfrozen model performance

Methods Comparison

The table illustrates the performance comparison of different models trained over varying epochs. Initially, the Frozen Model showed a significant improvement in both loss and validation loss compared to the Initial Frozen Model, suggesting that fine-tuning certain layers led to better performance. However, continuing training for 20 epochs, both Frozen and Unfrozen Models demonstrated further improvements in loss metrics. Interestingly, while the Frozen Model slightly outperformed the Unfrozen Model in terms of loss after 20 epochs, the Unfrozen Model exhibited a lower validation loss, indicating better generalization ability. These results highlight the importance of model architecture and training strategy in achieving optimal performance for specific tasks.

Table 1. Comparison of model performance across epochs

Model	Epochs	Loss	Validation Loss
Initial Frozen Model	10	3.24	5.27
Frozen Model	10	2.63	2.77
Unfrozen Model	10	2.64	2.66
Frozen Model	20	2.36	2.55
Unfrozen Model	20	2.36	2.39

Comparing the frozen and unfrozen models provides valuable insights into the trade-offs between stability and adaptability in neural network training for image captioning:

1. Stability vs. Adaptability: Freezing layers provides stability and prevents overfitting by relying on general features learned from extensive pre-training. In contrast, unfreezing layers allows the model to adapt more deeply to the specific characteristics of the captioning task but at the risk of overfitting.
2. Generalization: The frozen model maintains better generalization as seen in less volatile validation losses. In contrast, the unfrozen model, while potentially more powerful, requires careful handling of epochs and regularization to harness its full potential without overfitting.a capability to generate more contextually rich captions, as indicated by higher BLEU scores on some complex scenes. However, in this case, both models yield nearly identical results, but the frozen model benefits from faster training speeds.

DISCUSSION FOR FUTURE WORKS

In this project, an advanced image captioning system using a CNN-RNN architecture is developed, and trained on the Flickr8k dataset. This dataset comprises a variety of images accompanied by human-generated captions, which helped in fine-tuning the model's ability to generate accurate and relevant descriptions. By thoroughly preprocessing this dataset, we ensured that the model learned from clean and standardized data, optimizing both the training process and the quality of the output captions.

To enhance the capabilities of our system, integrating the CLIP model from OpenAI could be a significant step forward. CLIP's dual training on both images and text allows it to bridge the gap between visual content and language effectively, potentially enabling our system to produce more contextually aligned captions. Additionally, increasing the number of training epochs could further improve the model's ability to refine its understanding of complex image features and caption nuances, leading to higher quality outputs.

Expanding the training dataset to include larger and more diverse collections, such as the MS COCO or Conceptual Captions, could also enhance the model's robustness and generalizability. This approach would expose the model to a broader range of image contexts and caption styles, crucial for improving its performance across various real-world applications. Through these advancements—leveraging CLIP, increasing training epochs, and utilizing larger datasets—the image captioning system could achieve new heights in accuracy and applicability, making it more effective in diverse scenarios.

REFERENCES

[1] Hossain, M. Z., Sohel, F., Shiratuddin, M. F., & Laga, H. (2019). "A comprehensive survey of deep learning for image captioning." ACM Computing Surveys (CSUR), 51(6), 1-36. DOI:10.1145/3295748. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[2] 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." Proceedings of the IEEE conference on computer vision and pattern recognition, 6077-6086. K. Elissa, "Title of paper if known," unpublished.

[3] Xia, F., Wang, P., Chen, L., & Yuille, A. L. (2020). "See, Think, and Act: End-to-end Learning of Task-Oriented Visual Dialog Systems with Reasoning." IEEE Transactions on Pattern Analysis and Machine Intelligence. DOI:10.1109/TPAMI.2020.3007032. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740-741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[4] Cornia, M., Baraldi, L., Serra, G., & Cucchiara, R. (2020). "Meshed-Memory Transformer for Image Captioning." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 10578-10587.

[5] Srikanth, M., Javeed, J., Babu, M. G., & Varun, N. (2024). "A Hybrid CNN-LSTM Approach for Image Caption Generation." Musik In Bayern, Vol. 89, Issue 4, April 2024, pp1-14. DOI:10.15463/gfbm-mib-2024-250.

[6] Thati, B. M. K., Voddi, S., Busa, S., Surendra, Swarup Kumar, J. N. V. R., & Raja Rao, M.V.L.N. (2023). "A CNN and LSTM-based Model for Creating Captions for Photos." International Journal on Recent and Innovation Trends in Computing and Communication, 11(6s), 543. DOI:10.17762/ijritcc.v11i6s.6965. Available online at http://www.ijritcc.org.

[7] Mullachery, V., & Motwani, V. (2018). Image Captioning. arXiv preprint arXiv:1805.09137

[8] Poddar, A. K., & Rani, R. (2023). "Hybrid Architecture using CNN and LSTM for Image Captioning in Hindi Language." Procedia Computer Science, 218, 686-696. DOI:10.1016/j.procs.2023.01.049. Available online at www.sciencedirect.com