**A report on**

Deep Learning for Image Caption Generation

**[CSE- 3281]**

Submitted By

**Chetan Krishna M 210962126**

**Avinash Raju Konduru 210962166**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**MANIPAL INSTITUTE OF TECHNOLOGY,**

**MANIPAL ACADEMY OF HIGHER EDUCATION**

**APR 2023**

Deep Learning for Image Caption Generation with CNNs and LSTMs

CHETAN KRISHNA M AVINASH RAJU KONDURU

*Department of Computer Science Department of Computer Science*

*Manipal Institute of Technology Manipal Institute of Technology*

Udupi, INDIA Udupi, INDIA

chetan.maddieni@learner.manipal.edu konduru.raju@learner.manipal.edu

***Abstract*— This deep learning project delves into the realm of Image Caption Generation, integrating Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), specifically the Long Short-Term Memory (LSTM) variant. We extract intricate image features by leveraging the Flickr8k dataset and ResNet50 pre-trained model. The neural network architecture comprises an image feature extraction sub-model, a captions generation sub-model, and interconnected LSTM layers. Trained on a subset of the dataset, the model, along with the essential vocabulary dictionary, is preserved for future use.**

I. INTRODUCTION

The research aims to implement an Image Caption Generation system using deep learning techniques, explicitly combining Convolutional Neural Networks (CNNs) for image feature extraction and Recurrent Neural Networks (RNNs), specifically Long Short-Term Memory (LSTM), for generating human-readable captions. The dataset used is Flickr8k, containing images and corresponding captions. TheResNet50 pre-trained model is employed to extract image features. The captions are pre-processed, tokenised, and converted into numerical sequences. A custom generator function is designed to feed the image features, previously generated text, and the next word to predict into the model.

The neural network model consists of an image feature extraction sub-model, a captions generation sub-model, and connected LSTM layers. The model is trained on this architecture using a subset of the dataset. The trained model, along with the vocabulary dictionary, is saved for future use. Finally, sample predictions are generated by randomly selecting images and predicting captions using the trained model.

II. LITERATURE REVIEW

[1]."Show and Tell: A Neural Image Caption Generator" proposed a sequence-to-sequence model where a convolutional neural network (CNN) extracts image features, and a recurrent neural network (RNN) generates captions. The model is trained using a combination of maximum likelihood estimation and reinforcement learning to improve the quality of generated captions.

[2]. "Deep Visual-Semantic Alignments for Generating Image Descriptions" introduced an image captioning model that leverages deep convolutional networks and recurrent neural networks. The model aligns image and text representations in a shared semantic space, enabling it to generate more accurate and contextually relevant captions for images.

[3]. "Attend and Tell: Neural Image Caption Generation with Visual Attention" utilised an attention mechanism to dynamically focus on different parts of the image during caption generation. This approach enhances the model's ability to capture fine-grained details and improves the overall quality of generated image captions.

[4]. "Image Captioning with Semantic Attention" incorporated semantic attention into the image captioning process. By considering semantic information during caption generation, the model achieved improved alignment between image content and generated captions, resulting in more contextually rich and meaningful descriptions

[5]. "Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning" introduced an adaptive attention mechanism with a visual sentinel to decide when to focus on relevant image regions. This approach improved the model's capability to selectively attend to critical visual features, leading to more informative and coherent image captions.

[6]. "Stacked Cross Attention for Image-Text Matching" proposed a stacked cross-attention mechanism that enables effective image-text matching. The model leverages multiple layers of cross-attention to capture intricate relationships between image and text features, enhancing the overall performance of image captioning and text-based image retrieval tasks.

[7]. "Image Captioning with Transformer" applied the transformer architecture, initially designed for sequence-to-sequence tasks, to image captioning. This approach demonstrated the effectiveness of self-attention mechanisms in capturing long-range dependencies and relationships between image regions, leading to improved caption generation.

[8]. "Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering" introduced a model with bottom-up and top-down attention mechanisms. The bottom-up attention focuses on salient image regions, while the top-down attention captures contextual information. This dual attention approach significantly improved the model's ability to generate accurate and contextually relevant image captions.

[9]. "Generating Natural Language Descriptions for Semantic Representations of Human Brain Activity" focused on generating natural language descriptions for semantic representations of human brain activity. The study employed a combination of neural networks and semantic representations to bridge the gap between brain activity patterns and human-understandable language.

[10]. "Neural Image Caption Generation with Visual and Semantic Alignments" integrated visual and semantic alignments into the image captioning process. By jointly considering visual and semantic information, the model achieved improved alignment between image content and generated captions, resulting in more contextually relevant and coherent image descriptions.

III. RESEARCH GAP

"The existing methodologies for image caption generation face significant challenges that limit their effectiveness and applicability. These challenges include difficulties in bridging the semantic gap between visual content and language, inadequate comprehension of contextual nuances within images, leading to imprecise descriptions, and the presence of inaccuracies and ambiguities due to reliance on non-deep learning methods. Furthermore, scalability issues arise when traditional models encounter diverse datasets or evolving visual recognition tasks, hampering their adaptability and generalization capabilities. Addressing these gaps requires the development of a novel image caption generation system leveraging deep learning techniques, specifically integrating Convolutional Neural Networks (CNNs) for robust image feature extraction and Recurrent Neural Networks (RNNs) for context-aware caption generation."

The problem at hand revolves around the shortcomings of existing image caption generation methods, prompting the need for an advanced and more effective solution. Key issues include:

• Semantic Gap: Traditional methods struggle to bridge the semantic gap between visual content and language, resulting in captions that lack detail and precision.

•Limited Context Understanding: Conventional approaches often fail to comprehend the broader context of images, leading to inadequate descriptions, especially in complex scenes or scenarios involving multiple objects.

• Inaccuracies and Ambiguities: The inherent limitations of non-deep learning methods contribute to inaccuracies and ambiguities in generated captions, impacting the overall quality and informativeness.

•Scalability Challenges: Traditional models may face scalability challenges, mainly when dealing with diverse datasets or evolving visual recognition tasks, restricting their adaptability and generalisation.

Addressing these issues requires the development of a novel image caption generation system that leverages the power of deep learning, explicitly combining Convolutional Neural Networks (CNNs) for robust image feature extraction and Recurrent Neural Networks (RNNs) for context-aware caption generation.

V. LIMITATIONS:

* Existing methods need help to capture nuanced relationships between visual content and descriptive language.
* Traditional approaches produce generic and sometimes inaccurate captions due to the limited capacity to discern intricate details within images.
* Handling complex scenes, dynamic contexts, or images with multiple objects is difficult.
* Scalability and adaptability issues hinder performance on diverse datasets.
* There is a need for more sophisticated models that integrate deep learning techniques to capture intricate visual features and context.

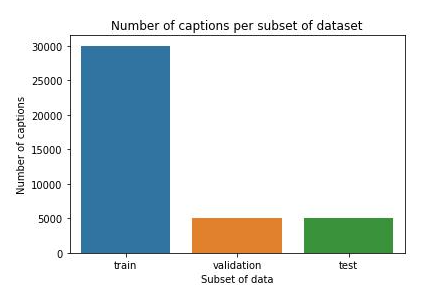
IV. OBJECTIVES

* Develop an advanced Image Caption Generation system using cutting-edge deep learning techniques to bridge the gap between visual content and language description.
* Integrate Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) with Long Short-Term Memory (LSTM) cells to enhance the model's ability to capture fine-grained visual features and generate precise captions.
* Leverage the robust ResNet50 pre-trained architecture for feature extraction to improve the model's ability to accurately understand and describe complex visual scenes.
* Utilise the Flickr8k dataset for training, aiming to develop a sophisticated system capable of generating contextually relevant captions for a wide range of images, thus improving scalability and adaptability.
* Contribute to the advancement of computer vision and artificial intelligence by developing a nuanced understanding of the relationship between images and language, thus driving significant advancements in the field.

V. METHODOLOGY

*A. Dataset loading:*

We used the Flickr 8k dataset for the image caption generator application. This dataset includes various images with diverse settings and circumstances. There are 8000 images in the Flickr 8k dataset, each with five captions. We divided the 8000-image dataset into three groups: 6000, 1000, and 1000. Each image has separate training, validation, and testing sets according to dimensions.

.

*B. Data Transformation:*

Objective of Dataset Transformation: Start by stating the purpose of dataset transformation: to preprocess the images and captions in a standardized format suitable for training the model.

Description of Transformations:

Resize: Images are resized to a consistent size of 224x224 pixels using bilinear interpolation. This step ensures that all images have the exact dimensions, which is required to feed them into the neural network.

To Tensor: Converts the image data from PIL Image to PyTorch tensor format. This transformation is necessary as PyTorch models expect input data in tensor format.

Normalisation: Normalizes the pixel values of the images using mean and standard deviation values of (0.485, 0.456, 0.406) for mean and (0.229, 0.224, 0.225) for standard deviation. Normalisation helps in stabilising the trainingprocess and improving convergence.

Justification for Each Transformation:

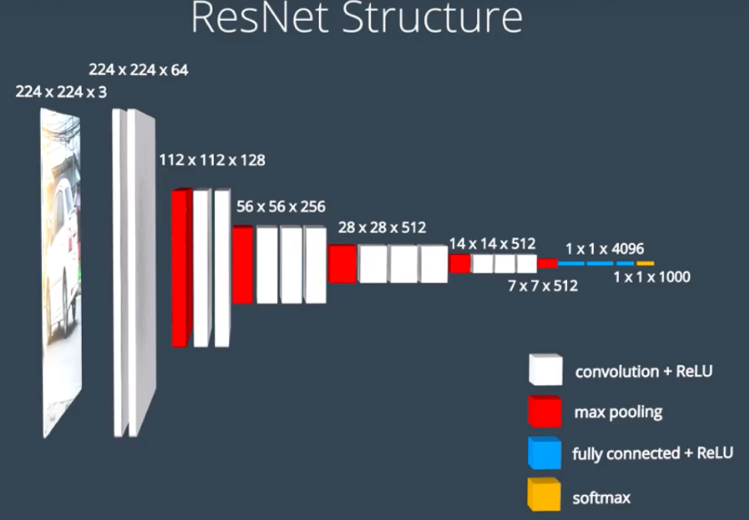
Resize: Ensures uniformity in image dimensions, which is essential for efficient batch processing and model training.

ToTensor: Converts the image data to tensors, enabling compatibility with PyTorch's computational graph and tensor-based operations.

Normalisation: Centres the data around zero mean and scales it to unit variance, stabilising the training process by preventing gradients from vanishing or exploding.

Implementation Details:

*C. Model Definition:*



* Model Architecture: Our image captioning system employs the famous encoder-decoder architecture. The encoder processes input images to extract meaningful features, while the decoder generates corresponding captions based on these features. The encoder component utilises a pre-trained ResNet-50 convolutional neural network. ResNet-50 is chosen for its depth and effectiveness in extracting high-level image features. We employ a Long Short-Term Memory (LSTM) recurrent neural network in the decoder. LSTM is well-suited for sequential data processing tasks and helps generate coherent captions.
* Encoder Details: We leverage ResNet-50 as our encoder due to its remarkable performance in image classification and feature extraction tasks. The network is pre-trained on ImageNet, providing robust feature representations for our image captioning task. Features extracted by ResNet-50 are flattened and passed through a linear layer to obtain a fixed-size feature vector representing each image.
* Decoder Details: Our decoder consists of LSTM cells, which sequentially process the encoded image features and generate captions word by word. An embedding layer maps input word tokens to continuous vector representations, facilitating meaningful representations for each word in the vocabulary. The LSTM cells produce hidden states at each time step, capturing the context and generating predictions for the next token. Finally, a fully connected layer computes the probability distribution over the vocabulary, predicting the following word in the caption.
* Model Hyperparameters: We set the embedding size to 400 and the hidden size of the LSTM decoder to 512, chosen based on empirical studies and experimentation. The decoder consists of two LSTM layers to capture intricate dependencies in the sequential data. We incorporate a dropout probability of 0.3 in the decoder to mitigate overfitting during training.
* Implementation Details: We implement our model architecture using PyTorch, a widely used deep learning framework. The model is instantiated, and parameters are initialized according to standard practices. During the forward pass, input images are processed by the encoder, and the resulting features are fed into the decoder to generate captions sequentially.

*D. Training loop*

* Training Loop Overview: The training loop is a critical component of our image captioning system and is responsible for optimizing the model parameters to minimize captioning errors. This loop iterates over the entire training dataset multiple times (epochs), updating the model parameters using backpropagation and gradient descent.
* Steps in the Training Loop: Data Loading: At the beginning of each epoch, the training dataset is loaded using a Data Loader, which handles batching and parallel data loading for efficiency. Forward Pass: The input images and their corresponding captions are fed into the model for each batch of data. Loss Calculation: The model's predictions are compared with the ground truth captions using a suitable loss function, such as cross-entropy loss. Backpropagation: The gradients of the loss function concerning the model parameters are computed using backpropagation. Parameter Update: The optimizer updates the model parameters (weights and biases) based on the calculated gradients and the chosen optimization algorithm (e.g., Adam). Logging: Optionally, relevant metrics such as loss values are logged to monitor the training process and performance evaluation. Repeat: These steps are repeated for the entire training dataset for a predefined number of epochs.
* Implementation Details: We implement the training loop using Py Torch, a popular deep-learning library which provides efficient tensor computations and automatic differentiation. The training loop is encapsulated within a Python function or a class method, making it modular and reusable across different experiments. Hyperparameters such as learning rate, batch size, and the number of epochs are carefully chosen based on empirical studies and best practices in deep learning.
* Performance Monitoring: Throughout the training process, we monitor vital metrics such as training loss and, optionally, validation loss on a separate validation dataset. Early stopping may be employed based on the validation loss to prevent overfitting and determine the optimal stopping point.
* Resource Utilization: The training loop utilises computational resources such as CPU or GPU for model training. We leverage GPU acceleration to expedite training and efficiently handle large-scale datasets.

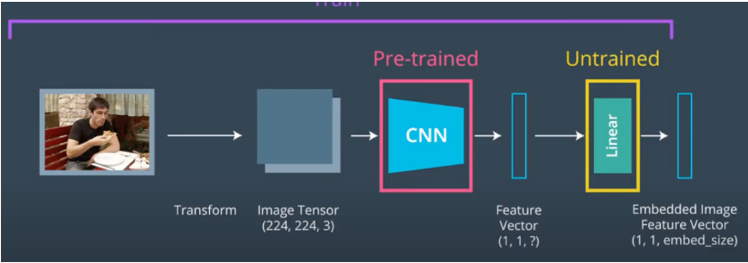
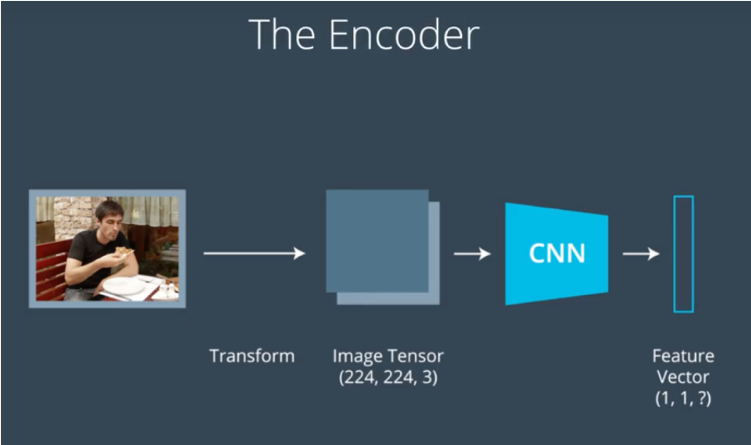
*E. Validation loop*

Several annotated picture datasets are available for the task of captioning photos. The two most popular ones are Flickr and Pascal VOC dataset, MSCOCO Dataset and 8K. The dataset for captioning Flickr 8K images is employed in the suggested model. 8,092 images make up the dataset known as Flickr 8K. Photos from the website Flickr.com. This dataset includes various daily activities and the subtitles that go with them. Each item in the first image after the image is identified is a description based on the things in a picture.[4] We divided this corpus of 8,000 photos into three separate sets. 6000 pictures make up the training data (DTrain), while each development and test dataset has 1000 photos. The potential of the image-caption pairs to connect previously undiscovered images and captions must be assessed to evaluate them. The model that creates natural language sentences can be evaluated using the BLEU (Bilingual Evaluation Understudy) Score. It explains how human-generated sentences differ from natural sentences. It is frequently used to assess how well machine translation performs.

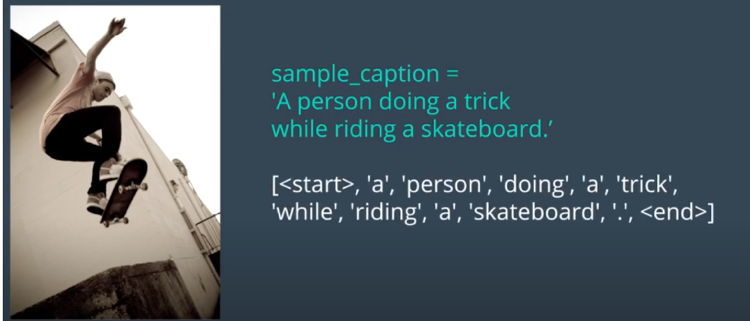
Summary:

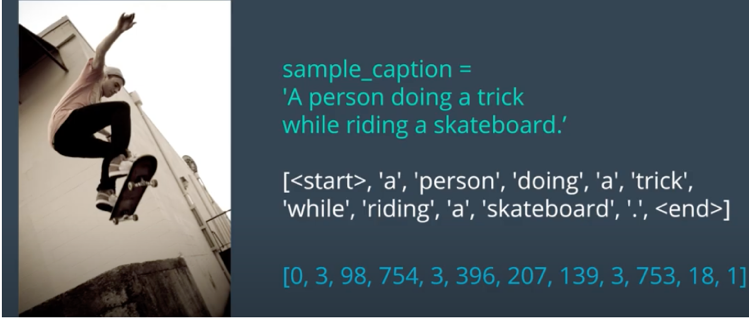
Work is proposed in three phases.

A. Extraction: Images are being extracted for their various features. Vector features, also referred to as embeddings, are produced. The CNN model removes the original photos' characteristics before being reduced to smaller and feature vectors compatible with RNNs. It also goes by the name Encoder.



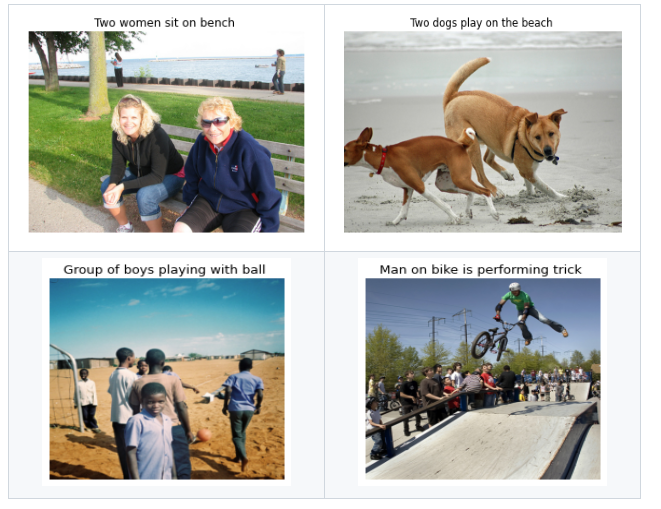
B. Tokenization Tokenization: The application's next stage is RNN, which decodes the feature vectors from CNN that were fed to it. The order of the words is as follows: is assumed and regardless of how the captions are produced.





C. Prediction: The final stage after tokenization is prediction. The vectors in this are decoded, and the get prediction command generates the final output () function.

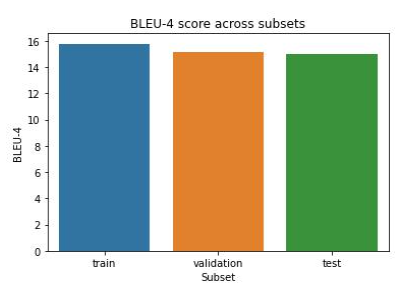
VI. DISCUSSION AND ANALYSIS OF RESULTS



Model performance was evaluated using the BLEU score. These results (quantitative and qualitative) were acquired by leveraging greedy decoding. Results of higher quality can be obtained by using beam search.

Model Performance

Below, we can see the model performance of all of the subsets. We can see that the model has a high generalisation performance.



Based on quantitative results, we can see that the BLEU-4 score is relatively high. This model was trained on a local GPU, GTX 1650Ti, so there were limitations in the hardware computing power. We can achieve better results if we train this model for a long time. Besides that, using beam search would also improve results, but beam search can add a significant computational overhead.

.

VII. CONCLUSION AND FUTURE WORK

The model has been trained and tested to generate the correct captions for the imported photos effectively.

The suggested model is based on multi-label categorization and generates captions using a CNNRNN technique. CNN serves as an encoder, and RNN as a decoder.

we examined deep-learning approaches to image captioning. It highlighted the benefits and drawbacks, showed a general block diagram of the critical groupings, and how to categorize image annotation systems. Each metric and dataset's advantages and disadvantages have been listed separately.

While comprehensive image labelling methods that can produce high-quality labels for virtually every image have yet to be developed, deep learning-based image labelling systems have made substantial advancements in recent years. Automated captioning has continued to be a hot area of research for some time, especially with the advent of new deep-learning network architectures. It uses the captions stored in a text file and the 8000 photos from the Flickr 8k collection.

. VIII. REFERENCES

1. O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and Tell: A Neural Image Caption Generator."

2. A. Karpathy and L. Fei-Fei, "Deep Visual-Semantic Alignments for Generating Image Descriptions."

3. K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, "Attend and Tell: Neural Image Caption Generation with Visual Attention."

4. Q. Chen, X. Zhu, Z. Ling, S. Wei, and H. Jiang, "Image Captioning with Semantic Attention."

5. J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning."

6. X. Huang, Z. Zhao, Y. Zhang, Y. Chen, M. Wu, and Y. Zhuang, "Stacked Cross Attention for Image-Text Matching."

7. L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, "Image Captioning with Transformer."

8. P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering."

9. R. Hu, M. Rohrbach, J. Andreas, T. Darrell, and K. Saenko, "Generating Natural Language Descriptions for Semantic Representations of Human Brain Activity."

10. J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, "Neural Image Caption Generation with Visual and Semantic Alignments