# Image Captioning With RNN

Aakash Divakar
Virginia Tech
aakash4869@vt.edu

## 1. Abstract

*Image captioning, a subfield of computer vision and natural language processing, aims to generate textual descriptions for images automatically. This report investigates the application of Long Short-Term Memory (LSTM) networks, a type of recurrent neural network (RNN), in image captioning tasks. LSTM networks have shown promising results in sequence modeling due to their ability to capture long-term dependencies and handle variable-length input sequences. This report provides an overview of the image captioning process, with InceptionV3 and ResNet50 used for feature extraction and results of both will be compared. To evaluate the model's performance Bleu score was chosen and was implemented on test dataset.*

## 2. Introduction

Images surround us in our daily lives, from online sources to printed materials. While humans intuitively understand most images, automating this understanding for machines would unlock powerful applications. To bridge this gap, automatic image captioning systems rely on descriptive information. Image captioning provides elegant solutions to complex problems. Automatic indexing powers content-based image retrieval (CBIR), tackling challenges faced by diverse industries. Doctors can efficiently search medical image databases, retailers can improve product discovery, and historians can unlock visual archives.

Image captioning, a vibrant area of AI research, focuses on teaching machines to both interpret images and generate descriptive text. This process requires understanding objects, their relationships, and scene context. Deep learning techniques are powerful here: convolutional neural networks (CNNs) excel at feature extraction, while recurrent neural networks (RNNs) are commonly used for language generation.

The main aim of this report is to build a image captioning project first using CNN such as InceptionV3 and ResNet50 to extract features and then use LSTM which is a advanced version of RNN for captioning.

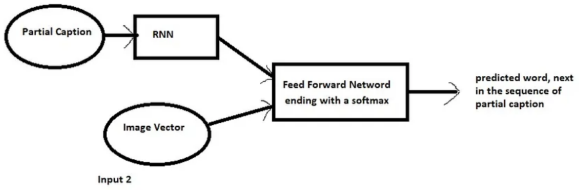


Figure 1. Basic Architecture

## 3. Approach

### 3.1. 1. Dataset Preparation:

The very first step was to start by obtaining a dataset suitable for image captioning, such as the Flickr dataset, which contains images paired with human-generated captions. It contained 6000 training images and 1000 test images with each containing 5 captions. Data preprocessing for image captioning involves cleaning and formatting both image and text data. Firstly, text data is cleaned by removing noise like special characters and punctuation. Spelling errors also need to be corrected. Duplicate captions are removed to prevent bias. Next, captions are tokenized into individual words or subwords, and a vocabulary mapping each token to a numerical index is created. Caption lengths are standardized through padding or truncation. For images, resizing to a consistent dimension ensures uniformity for input to the neural network. Common sizes include 224x224 or 299x299 pixels. Pixel values are normalized to a fixed range, typically between 0 and 1. The dataset is then split into training, validation, and test sets for model training, hyperparameter tuning, and evaluation, respectively. These preprocessing steps ensure that the data is properly formatted and ready for feature extraction and training with LSTM networks.

### 3.2. Model Architecture:

This step involves designing a neural network architecture that combines convolutional neural networks (CNNs) for image feature extraction and LSTM networks for sequential caption generation. I have used a pre-trained CNNs

like ResNet50 and InceptionV3 individually to extract image features and compare their result and feed them into the LSTM network. The model architecture is shown in Fig 2.

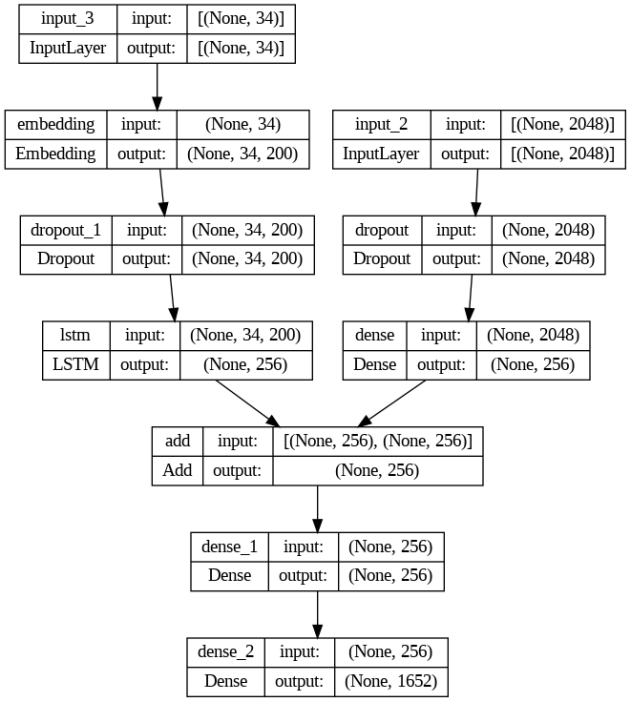


Figure 2. Model Architecture

### 3.3. Image Feature Extraction:

Next step is to implement the CNN model to extract features from input images. The Classification layer from the CNN needs to be removed and then use the output of the last convolutional layer as image features. These features capture the visual information necessary for generating captions. I have used ResNet50 and InceptionV3 and compared their results to see which performs better. The model had an input layer with shape 2048 followed by a dropout of 0.5 then a dense layer of 256 units and finally ReLu activation function was added.

### 3.4. Text Sequence Generation:

After feature extraction an LSTM network is built to generate captions based on the extracted image features. Initialize the LSTM network with the image features and feed it with the tokenized captions. Train the LSTM to predict the next word in the caption sequence given the previous words. This branch had an input layer of shape of maximum length which is the maximum length of input sequence. Then an embedding layer of vocab size and embedding dimension was added which was followed by a 0.5 dropout rate. Then the LSTM layer with 256 units was added and the last layer was softmax layer.

### 3.5. Model Training:

Train the combined CNN-LSTM model using the image features and corresponding tokenized captions. Using techniques like teacher forcing, where the ground truth words are used as input during training, to stabilize the training process. Techniques such as dropout regularization were used to prevent overfitting.

### 3.6. Caption Generation:

After training, the model can generate captions for new images. For inference, input the image to the CNN to extract features, then feed the features to the LSTM network to generate a sequence of words one at a time. Use greedy search or beam search to select the most likely word at each step until an end token is generated or a maximum sequence length is reached.

### 3.7. Evaluation:

Evaluate the performance of the model using metrics such as BLEU (Bilingual Evaluation Understudy) to measure the similarity between generated and reference captions.

## 4. Experimental Results

The model was trained first using InceptionV3 CNN and then the LSTM. Training was done for 40 epochs and the loss curve is shown in Fig 2. The loss was reported as 2.225. The Learning rate was kept at 0.0001 while a number of picture per patch was kept at 6 which were fed to the data generator. A similar procedure was followed for ResNet50 also. No significant difference was reported. Random images were fed and tested to check was good the model was performing and the captions were analysed. To evaluate the model BLEU score was used which was implemented on all images in the test dataset. The reference captions were taken from the dataset and predicted captions were generated using the trained model which were used for evaluation using BLEU score. BLEU score off all the images were stored and highest was 0.91. To check how model performed overall an average BLEU score was calculated which was 0.545.

## 5. Discussion

After evaluating the model on test set of 1000 images it was observed that the average BLEU score was 0.545. An average BLEU score of 0.54 suggests that the machine-generated translation has moderate similarity with the human reference translations. Some translations were highly accuracy and had a BLEU score of 0.91 while some were almost 0. Some were partially correct in describing the image but failed to mention or add a specific detail such as 'boy in

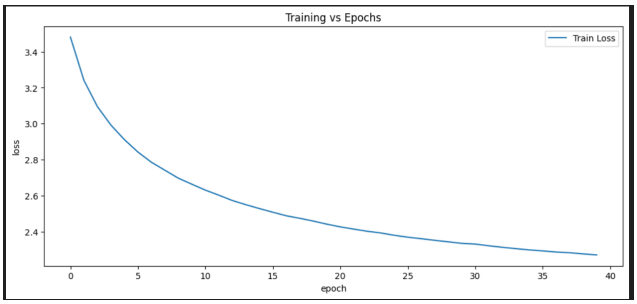


Figure 3. Plot of Loss Vs Epochs

blue shorts jumps into pool' while the boy was wearing red shorts as shown in the Fig 4, here the entire action is correctly predicted moreover the objects are also correct but only the color was wrong.

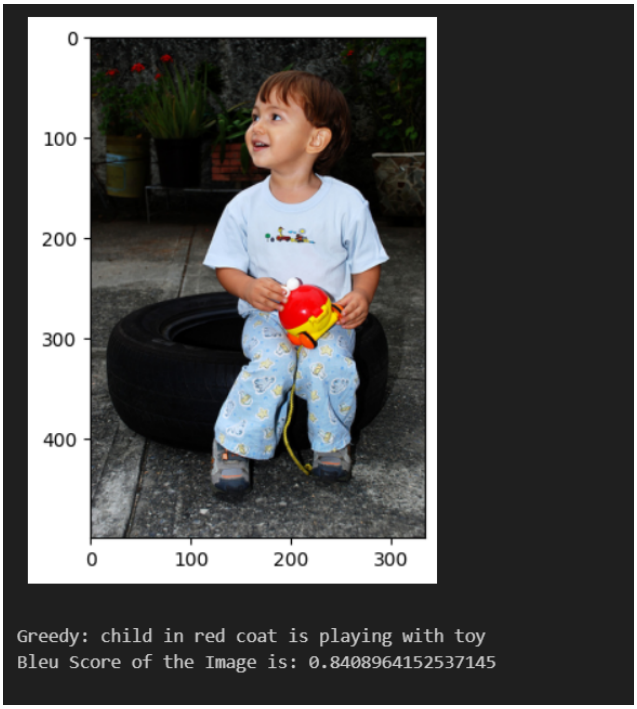


Figure 4. Plot of Loss Vs Epochs

For one more similar example as seen in the Fig 5, the boy with blue shirt running with the ball. Here too the color was mistaken but all other actions were rightly predicted.

In Fig 6, the image is captioned as two boys play soccer on the grass. Here the model fails to identify number of objects but correctly predicts the action and the place.

In Fig 7, the captioned image is man in red jacket is climbing the mountain. Here the action, object and even the color is correctly predicted. This image is captioned in the right manner successfully.

Even though many good results were obtained few results were completely wrong and were not predicted correctly at all. For example the figure 8 is captioned 'the furry furry is wearing red shirt', this does not relate to the picture at all and is completely wrong.

Similar case is in the Fig 9 where the caption is 'the duck waves the rain' and two motorcycles are seen racing. Model completely failed to predict the caption for such images.


Figure 5. Result


Figure 6. Plot of Loss Vs Epochs


Figure 7. Plot of Loss Vs Epochs

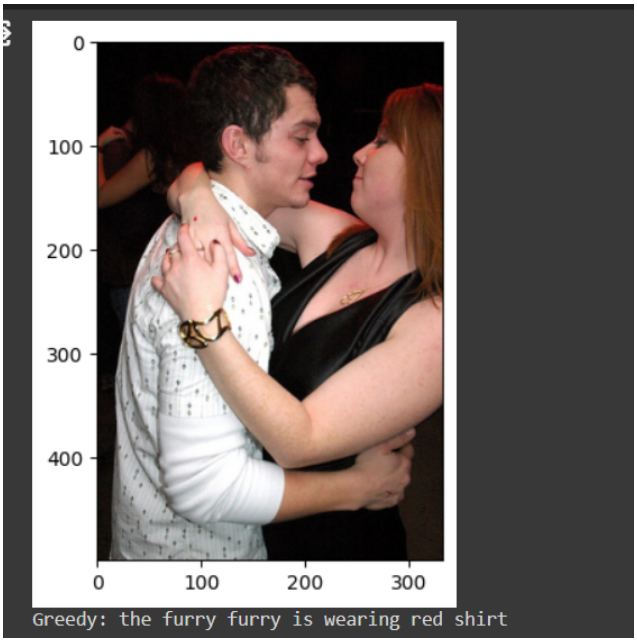


Figure 8. Result


Figure 9. Result

## 6. Conclusion and Future scope

This report contributes to the image captioning field by offering a comprehensive review of deep learning approaches. The analysis highlights the strengths of using CNNs for feature extraction and RNNs for language generation, providing a valuable foundation for further development and optimization of these powerful techniques. More advanced version of image captioning can be built by improvising certain aspects of this report such as trying out different models for feature extraction and also trying to improve the captioning model i.e LSTM could help in achieving better results. Training the model for more epochs can also be worth experimenting to check if loss decays further significantly which will ultimately result in better captioning. Beam search can be implemented instead of using greedy search where at each step, beam search keeps track of the 'k' most likely sequences (where 'k' is the beam width). This allows it to explore alternative word choices and potentially discover better sequences that greedy search might miss. Advance Transformers are also powerful tools for Natural language processing tasks which can prove useful for high efficiency. Implementing these changes can further help in research of image captioning.

## 7. References

[1] Herdade, S., Kappeler, A., Boakye, K., Soares, J. (Year unavailable). Image Captioning: Transforming Objects into Words. Yahoo Research, San Francisco, CA, 94103.

[2] Inuwa, M. (2023, June 27). Vision Transformers (ViT) in Image Captioning Using Pretrained ViT Models. Analytics Vidhya. Retrieved from https://www.analyticsvidhya.com/blog/2023/06/vision-transformers/

[3] Lamba, H. (2018, November 4). Image Captioning with Keras. Towards Data Science. Retrieved from https://towardsdatascience.com/image-captioning-with-keras-teaching-computers-to-describe-pictures-c88a46a311b8