Generating Captions for Images

Aditi Akhilesh (adakhi@iu.edu)
Amrita Prakash (amriprak@iu.edu)
Shivani Bapuji Vhatkar (svhatkar@iu.edu)
Vaishnavi Pravin Apsingkar (vapsing@iu.edu)

December 3, 2024

Abstract

Making machines more powerful by teaching them to see and describe the world is possible due to image captioning. Image captioning has become an important tool in the modern era to help make the life of humans better. Image captioning is the process of generating descriptive text for an image. This project focuses on developing a model for generating captions using **Convolutional Neural Network (CNN)** and **Long Short-Term Memory (LSTM)** models. A CNN model such as VGG16 is used for extraction of features from the image, and the LSTM is used for caption generation. For this, the **Flickr_8k dataset** has been used, which contains 8,091 images along with five text descriptions for each image. This dataset has a very diverse set of images, making it suitable for training and evaluation. The pipeline of the model includes image data pre-processing for extracting the features and text data pre-processing by cleaning and tokenizing the captions to integrate them with the model. The training process showcases gradual improvement in the quality of the captions generated, which are later evaluated using **BLEU scores** (e.g., BLEU-1 and BLEU-2) and human evaluations. The model performs well in simple contexts but faces challenges with handling complex and ambiguous situations. The overall result of the project focuses on the use of CNNs for feature extraction and LSTMs for generating textual descriptions.

1 Introduction

Image captioning is one of the advanced Artificial Intelligence tasks that combines **Computer Vision** and **Natural Language Processing (NLP)** to generate accurate and coherent captions for given input images. This involves scanning the images, analyzing the objects and contexts, and then generating text or captions for the input image. By analyzing the image content such as objects, their actions, and the contexts within an image, the model helps generate coherent text that better describes the scene.

There are various applications of image captioning in areas including education, healthcare, and technology:

- **Automatic Content Creation:** Image captioning automates the process of generating captions (descriptive text) for images, reducing manual intervention. This is particularly useful for social media platforms, such as auto-generating product descriptions for online shopping platforms.
- **Accessibility Tools for Visually Impaired Users:** Image captioning plays an important role in improving accessibility for visually impaired users, helping them understand the objects, actions, and contexts within images. Tools like **Seeing AI** provide real-time descriptions for visually impaired individuals.

- **Search Engine Optimization (SEO) for Image Search:** SEO can be enhanced by associating images with relevant textual captions automatically. This improves image discoverability in search engines like Google, as the generated captions better match user queries.

2 Proposed Model

Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a type of feed-forward neural network. They have an input layer, hidden layers, and an output layer. CNNs are specialized neural networks designed to process two-dimensional (2D) input data, such as images. They are good at identifying important features such as edges, shapes, and patterns, making them suitable for tasks like classification, object detection, and feature extraction.

VGG16 Architecture

In our project, we used **VGG16**, which is a popular CNN architecture. VGG16 has 16 layers. Out of these, 13 are convolutional layers, and 3 are fully connected layers. It is trained on the *ImageNet Dataset*. We selected VGG16 for two main reasons:

1. Training a CNN from scratch needs a lot of time and computational resources. VGG16 is already trained on ImageNet, which saves us both time and effort.
2. The second-last layer of VGG16 gives a 4096-dimensional feature vector. This feature vector contains important information about the image.

In our project, we used VGG16 only up to its second-last layer. We removed the last prediction layer because we only needed the extracted features, not the class predictions.

Applications of CNNs

- **Image Classification:** Identifying objects in images (e.g., cats or dogs).
- **Object Detection:** Detecting objects in any video footage.
- **Medical Image Analysis:** Diagnosing diseases using images like X-rays or MRIs.
- **Feature Extraction:** Getting features for other tasks, such as generating captions for images.

Long Short-Term Memory (LSTM)

LSTM (Long Short-Term Memory) is a type of Recurrent Neural Network (RNN). LSTMs are made to handle sequential data and learn long-term relationships. Unlike normal RNNs, LSTMs solve problems like the vanishing gradient. This makes them better for tasks such as speech recognition, generating text, and image captioning.

Why LSTM for Our Project?

In this project, we chose LSTM for the following reasons:

1. LSTM processes sequential data, which is important for generating captions. Each word depends on the words before it and on the features of the image.

2. LSTMs have memory cells that store important information from earlier steps and remove unnecessary information. This makes them good for generating captions that are meaningful and correct.

Applications of LSTMs

- **Speech Recognition:** Converting speech to text.
- **Text Generation:** Writing stories or completing sentences.
- **Image Captioning:** Generating captions for images.
- **Language Translation:** Translating text from one language to another (e.g., English to French).

This combination of **CNN (VGG16)** and **LSTM** is perfect for our image captioning project. CNNs are good at extracting meaningful information from images, and LSTMs can create captions step by step based on this information.

3 Implementation Details

Dataset

The **Flickr_8k dataset** is used, containing 8,091 total images, each with five textual descriptions. These captions provide diverse perspectives based on the content of the image, making it suitable for training and evaluating the caption generation model. This lightweight dataset has lower computational costs compared to larger datasets like MSCOCO, making it more effective for prototyping. Due to these advantages, the Flickr_8k dataset was chosen for this project.

Data Pre-processing

Data pre-processing is a crucial step to ensure the data is consistent and structured for deep learning models. There are two types of data: **image data** and **text data**.

Image Data Pre-processing:

Image data is formatted to be used for feature extraction with a pre-trained CNN model like **VGG16**. Steps include:

- **Resizing Images:** Each image is resized to 224x224 pixels, the required input size for the VGG16 model. This ensures uniformity and compatibility with the model architecture.
- **Feature Extraction:** Features are extracted using VGG16 and stored using the **pickle module** for efficiency. Storing pre-computed features prevents repetitive image processing and saves computational time during model training.

Text Data Pre-processing:

The text data (image captions) is cleaned and formatted for use as input to the LSTM model. Steps include:

- **Cleaning Captions:** Punctuations and digits are removed, and text is converted to lowercase for consistency.

- **Adding Start/End Tokens:** Each caption is enclosed with tokens `"startseq"` and `"endseq"` to mark the start and end of the sequence, helping the LSTM model understand where to begin and stop caption generation.
- **Tokenizing Captions:** Each unique word in the dataset is assigned a corresponding integer index. This allows the model to process captions as sequences of numbers rather than raw text.

Example of Text Preprocessing:

- **Before Cleaning:** *"A child in a pink dress climbing stairs."*
- **After Cleaning and Tokenizing:** *"startseq a child in a pink dress climbing stairs endseq"*

The preprocessed image and text data are seamlessly integrated into the CNN and LSTM models for feature extraction and caption generation.

4 Architecture

Figure 1 shows the architecture of the model. The model architecture consists of two primary pathways - visual processing and text processing - which are converged for final caption generation.

4.1 Visual Processing Component

In encoder stage, model utilizes VGG16 convolutional network for visual feature extraction. Image passes through multiple layers of VGG16, where network extracts total of 4096 distinct features from penultimate layer. For improving model generalization, we have implemented dropout layer with 40% deactivation rate. These features are then compressed into 256 dimensions through Dense layer with ReLU activation function.

4.2 Text Processing Component

For text processing pathway, input captions undergo tokenization process for converting words into numerical format. These tokens are processed through embedding layer which creates 256-dimensional vectors for each word. Such representation helps in maintaining semantic relationships between similar words. After dropout layer implementation, embedded representations are fed into LSTM layer for sequential processing and context maintenance.

4.3 Integration and Output Generation

In decoder stage, model performs fusion of visual and textual information streams through addition operation. This creates unified 256-dimensional representation which captures both visual content and linguistic context. Final processing involves Dense layer with ReLU activation, followed by softmax layer which generates probability distributions across vocabulary space.

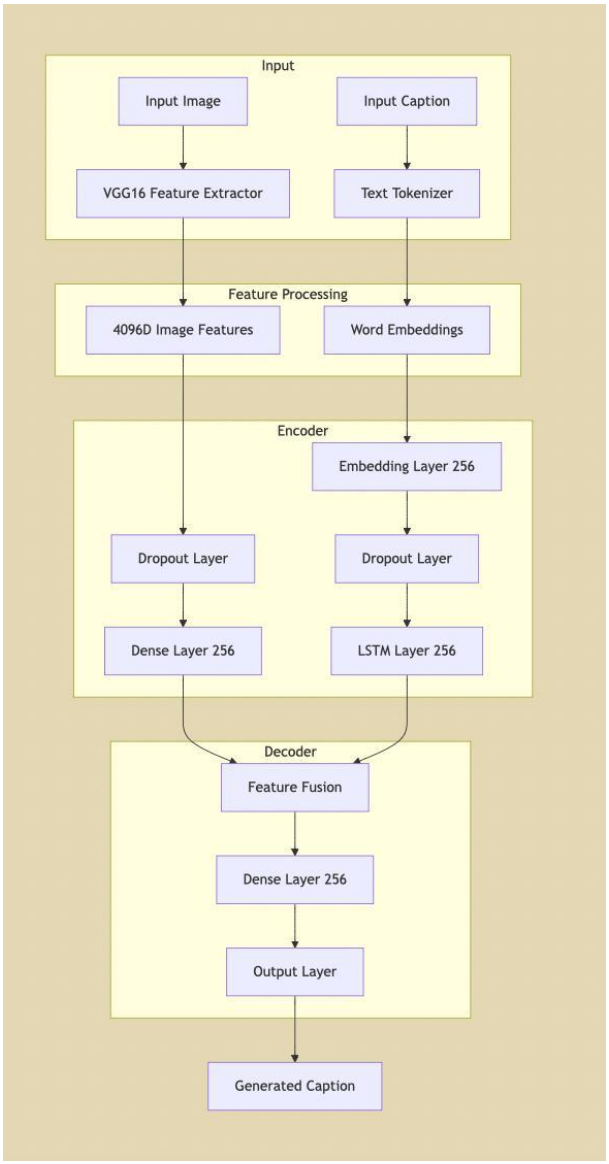

Figure 1: Model architecture

4.4 Caption Generation Model

- **Hyperparameters**:
  - Vocabulary size: 10,000
  - Embedding dimension: 256
  - LSTM units: 512
  - Optimizer: Adam
  - Loss function: Categorical cross-entropy
  - Batch size: 64

4.5 Training Process

- Training was conducted for 1 and 10 epochs.
- Features and captions were fed into the model using a data generator to handle large datasets.

- The model was saved as `caption-model.h5` for reuse.

5 Experimental Setup

Dataset Split

The dataset is divided into two parts:

- **90% for training:** To teach the model how to generate captions.
- **10% for testing:** To check the performance of the model on new data.

Batch Size

- We used a batch size of 64 images.
- The model learns from these 64 samples at a time before updating its weights.

Epochs

- The model is trained for 25 epochs. We tried with different epochs such as 10, 30 etc, but with 25 we were able to achieve good results.

Optimizer

- We used **Adam (Adaptive Moment Estimation)** as the optimizer. Adam adjusts learning rates automatically, which helps in faster and stable training.

Loss Function

- We used **categorical cross-entropy** as the loss function. This function helped us measure the difference between predicted words and actual words in the caption.
- The goal was to minimize this loss during training.

6 Result and Observations

6.1 Input 1

Image of a young girl wearing a pink outfit, standing with her hands on her hips, near a stroller, in an urban environment.

Figure 2: Girl with some posture

Actual Captions

- Child in all pink is posing nearby stroller with buildings in the distance.
- Little girl in pink dances with her hands on her hips.
- Small girl wearing pink dances on the sidewalk.
- The girl in a bright pink skirt dances near a stroller.
- The little girl in pink has her hands on her hips.

Predicted Caption

Girl in pink dances in parade.

Observations

1. The caption catches the main subject, "girl in pink," and her action, "dances." However, "parade" is not in the image.
2. It misses details like the stroller, background buildings, and the girl's pose. The caption is too general.
3. The verb "dances" adds liveliness, but it is unclear if the girl is dancing or posing.

Final Observation: The caption identifies the main elements but introduces irrelevant details like "parade." While it aligns partially with the image, a more accurate description of the context and pose is needed.

6.2 Input 2

Image of a group of people sitting and watching illuminated hot air balloons at night in an open area.

Figure 3: Hot air balloon viewing

Actual Captions

- Crowd watching air balloons at night.
- Group of hot air balloons lit up at night.
- People are watching hot air balloons in the park.
- People watching hot air balloons.
- Seven large balloons are lined up at nighttime near the crowd.

Predicted Caption

Crowd of people watching hot air balloons at night.

Observations

1. The predicted caption is accurate and captures the main subject: "crowd of people" and "hot air balloons at night."
2. It closely matches most actual captions but does not mention specific details like "seven large balloons" or "lit up."
3. The caption aligns well with what a human viewer would infer from the image.

Final Observation: The caption is highly accurate, capturing the main subject and action. However, it could be enhanced by including specific details, like the balloons being illuminated.

6.3 Input 3

Image of a man in a wetsuit throwing a small child wearing a wetsuit into the air, with a bright blue sky and palm trees in the background.

Figure 4: Baby thrown in air

Actual Captions

- Man in wetsuit is throwing baby wearing wetsuit up into the air.
- Man in wetsuit is throwing toddler up in the air and is ready to catch him.
- Man in water throwing little boy up in the air and waiting for him to come down so he can catch him.
- The man is in the pool and throwing small boy into the air.
- While water droplets fly, man throws little boy up in the air.

Predicted Caption

Man is throwing his board up into the air.

Observations

1. The predicted caption misidentifies the object being thrown, mentioning a "board" instead of a child.
2. It does not capture the context of the wetsuit, water, or interaction between the man and the child.
3. The caption provides some sense of action but is far from accurate.

Final Observation: The caption fails to identify the main object and context accurately. Significant improvements are needed in recognizing interactions and distinguishing between objects.

6.4 Input 4

Image of a light-colored dog jumping off a boat into the water, with a dock and another boat visible in the background.

Figure 5: Dog jump into water

Actual Captions

- Dog jumps into the pond.
- Dog leaping into the water near motorboat.
- Light-colored dog jumping off the back of a boat into the water.
- Tan dog jumping off of boat into the water.
- The yellow dog is jumping off of a ship into the water.

Predicted Caption

Dog is running through the water.

Observations

1. The predicted caption describes the dog in action but incorrectly identifies it as "running through the water" instead of "jumping into the water."
2. It does not mention the boat or the context of the dog leaping off it.
3. The caption partially aligns with the image's intent but lacks key details about the setting.

Final Observation: The caption captures the motion but incorrectly describes the action as running rather than jumping. It also misses key context, like the boat, making the description incomplete.

6.5 General Final Observation

Overall, while the captions often provide a reasonable interpretation of the images, they need improvement in understanding nuanced details, such as relationships between objects, environmental context, and precise actions. This emphasizes the need for enhanced model training on diverse datasets and refined attention mechanisms to generate captions that are both contextually rich and highly accurate. Despite these limitations, the generated captions are still reasonably close to what a human might infer from the images, making them a valuable starting point for further refinement.

7 Evaluation Criteria

7.1 BLEU Score Evaluation

Process

The BLEU (Bilingual Evaluation Understudy) score was computed using 10% of the test dataset. For each image, the generated captions were compared against reference captions to calculate n-gram overlaps. BLEU scores were calculated at different levels (BLEU-1 to BLEU-4) to evaluate how well the generated captions align with the reference text.

BLEU Scores

- **BLEU-1 (Unigram)**: Measures the overlap of individual words. The model achieved a score of **0.5369**, indicating a strong alignment at the word level. This shows that the model is effective at capturing the main subject or keywords from the image.
- **BLEU-2 (Bigram)**: Considers both individual words and pairs of consecutive words. The model scored **0.3202**, reflecting moderate coherence in phrase structure.
- **BLEU-3 (Trigram)**: Extends the evaluation to trigrams. The score of **0.2340** highlights a decline in fluency when evaluating longer word sequences, indicating that the model struggles to generate more complex phrases.
- **BLEU-4 (4-gram)**: Focuses on 4-grams to evaluate overall sentence structure. The score of **0.1201** suggests significant room for improvement in constructing grammatically complete and detailed captions.

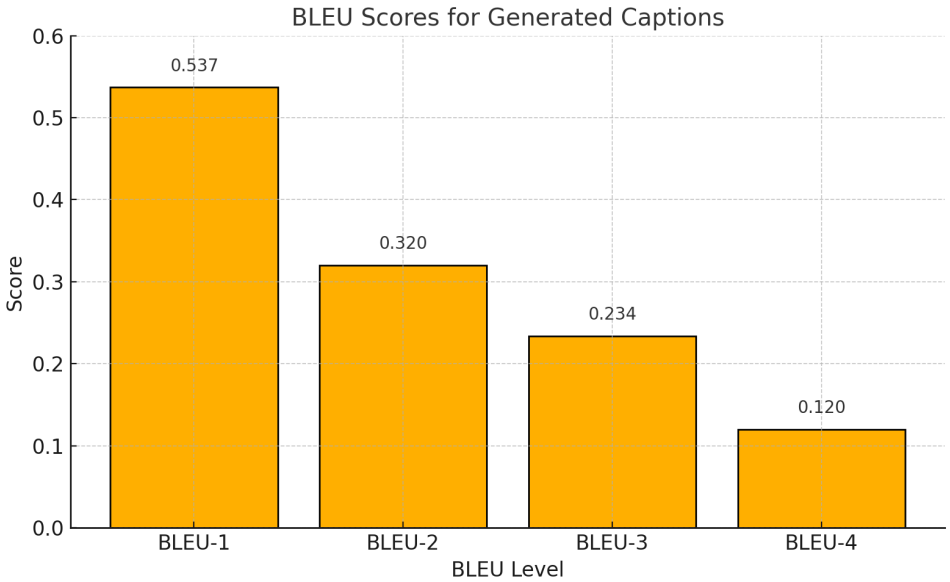

Figure 6: BLEU Scores for Generated Captions

Understanding the Scores

The relatively high BLEU-1 score suggests that the model is proficient at identifying key elements or objects in the image and representing them in its captions. However, the steady decline in scores from BLEU-2 to BLEU-4 indicates that the model has difficulty forming longer, grammatically correct, and contextually detailed captions. This highlights the need for further improvements in language modeling and context retention.

7.2 Human Evaluation

Process

Human evaluation complements the BLEU score by capturing qualitative feedback and assessing aspects that automated metrics might overlook. This was conducted using a selected subset of images from the test dataset, focusing on five key evaluation criteria.

Evaluation Process

- Ten images were selected from the test dataset for evaluation.
- Four evaluators independently assessed each generated caption by answering five specific questions.
- Ratings were assigned on a scale of 1 to 5, with 1 being poor and 5 being excellent.

Evaluation Questions Asked

- **Relevance:** Does the generated caption accurately describe the image content?
- **Fluency:** Is the caption grammatically correct and readable?
- **Detail:** Does the caption include important details about the image?
- **Novelty:** Does the caption provide unique insights or descriptions?
- **Overall Quality:** How well does the caption align with the human evaluator's expectations?

Results

- Average scores across the five criteria were computed for all captions, yielding a clear view of the model's strengths and weaknesses.
- A percentage score was calculated to indicate the model's overall performance in human evaluation.

Criterion	Average Score (Out of 5)	Percentage
Relevance	3.7	74%
Fluency	4.7	94%
Detail	3.5	70%
Novelty	2.9	58%
Overall Quality	3.5	70%

Figure 7: Human Evaluation

Observations

- Human evaluation highlighted that the captions performed well in **Relevance** and **Fluency**, with scores often between 3 and 5.
- Captions lacked sufficient **Detail** and **Novelty**, as evaluators noted that important contextual elements were often missing.

Future Work

To ensure a comprehensive evaluation, human evaluation can be extended to all images in the test dataset, providing a more robust analysis of the model's performance.

Link to Results : Human Evaluation Excel Sheet

7.3 Summary

The combination of BLEU scores and human evaluation provided a well-rounded assessment of the model's caption generation performance. The comparatively strong BLEU-1 score (0.5369) indicates that the model excels at capturing individual words that align with the image content, suggesting good general performance. However, lower BLEU-3 and BLEU-4 scores reveal areas for improvement in constructing grammatically complex and detailed captions. Human evaluation further corroborated these findings, highlighting that while the captions are relevant and fluent, they often lack depth and unique insights. This suggests that the model is a solid foundation for caption generation but requires further training to improve contextual richness and overall detail.

8 Challenges and Limitations

Training an image captioning model comes with several challenges and limitations that impacted the project:

- **Resource-Intensive Training:** Training the model on larger datasets like Flickr30K or MSCOCO demands significant computational resources and time. With limited access to high-performance hardware, it was difficult to fully optimize the model.
- **Fine-Tuning Pre-Trained Models:** Fine-tuning the VGG16 CNN for better feature extraction was challenging as it required a deep understanding of the pre-trained network and careful adjustments. Mistakes in this step led to degraded performance during early iterations.
- **BLEU Score Optimization:** Achieving higher BLEU scores was another challenge. Improvements in tokenization and hyperparameter tuning were needed but required substantial experimentation, which was constrained by time and available resources.

9 Discussion and Future Work

The project showed encouraging results, but there are several opportunities for improvement and further exploration:

- **Larger Training Datasets:** Training the model on larger and more diverse datasets such as Flickr30K or MSCOCO could improve its generalization and ability to generate more detailed and meaningful captions. This would require overcoming hardware limitations.
- **Advanced Architectures:** Incorporating newer architectures like Transformers could potentially improve the quality of the captions, as these models have shown state-of-the-art performance in similar tasks.
- **Real-Time Captioning:** Extending the model to support real-time captioning for live video feeds could make it more practical for real-world applications, such as assisting visually impaired individuals.

These future directions, although resource-intensive, would significantly enhance the model's capabilities and application scope.

10 Personal Takeaways

This project was a valuable learning experience, both technically and personally:

Technical Insights

- We developed a deeper understanding of computer vision and natural language processing techniques, especially the encoder-decoder model framework.
- We learned how to integrate CNNs like VGG16 for feature extraction and combine them with LSTMs for text generation.
- Evaluating model performance with BLEU scores also helped me understand how to measure the quality of machine-generated text.

Problem-Solving Strategies

- Breaking down complex tasks into manageable steps allowed me to tackle the project systematically.
- We became skilled at using online documentation, tutorials, and forums to address technical issues effectively.
- Iterative testing and evaluation taught me the importance of refining a model incrementally to achieve better results.

Overall, this project not only helped me gain hands-on experience with image captioning but also improved my confidence in solving challenging problems.

11 Conclusion

This project successfully implemented an image caption generator using CNN and LSTM models. Despite certain limitations, the model demonstrated the ability to generate meaningful captions with reasonable accuracy, as reflected in the BLEU scores.

The results are promising and provide a solid foundation for further work. By training on larger datasets and experimenting with advanced architectures, the system could be enhanced significantly. This experience has reinforced my understanding of AI techniques and their practical applications, making it a rewarding and insightful journey.

Access to our code: `https://www.kaggle.com/code/aditiakhilesh/group1finalprojectcode`

References

- Flickr30k Dataset: `https://www.kaggle.com/datasets/hsankesara/flickr-image-dataset`
- Flickr 8k Dataset: `https://forms.illinois.edu/sec/1713398`
- TensorFlow Documentation: `https://www.tensorflow.org/`

A Appendix

- Additional visualizations or tables.
- Code snippets or logs for reproducibility.