Image Caption Generation using CNNs and Transformers

Chaitanya Krishna Kommineni
Department of $School of Engineering and Applied Sciences (Data Science)$
University at Buffalo
Buffalo, NY 142603
{ckommine}@buffalo.edu

Abstract

This project aims to build an Image Captioning System that can automatically create meaningful descriptions for images by combining Computer Vision and Natural Language Processing techniques. A pretrained ResNet50 model is used to extract visual features from images, which are then passed into an LSTM model to generate captions. We used a smaller dataset of 500 images from the Flickr8k dataset to save time and computational resources. The system can be applied to help visually impaired individuals, improve image search systems, and manage digital content. The results show that the model can understand images and create relevant captions. In the future, adding attention mechanisms or transformer-based models could make it even better.

1 Introduction

The ability to generate meaningful captions for images is a challenging yet exciting task that combines the fields of Computer Vision and Natural Language Processing (NLP). An Image Captioning System aims to automatically describe the content of an image in natural language, which has a wide range of applications, including assisting visually impaired users, improving search engines, and enabling better digital content management. In this project, we develop an Image Captioning System that uses a pretrained ResNet50 model to extract visual features from images. These features are combined with captions processed using an LSTM-based model to generate sentences that describe the images. We use a subset of the Flickr8k dataset (limited to 500 samples) for training and evaluation, demonstrating how transfer learning and efficient preprocessing techniques allow the system to perform well even with limited data. This project not only provides a practical implementation of cutting-edge AI techniques but also highlights the potential of AI systems that can interpret visual content and express it in human language. The results demonstrate the system's ability to generate meaningful captions, laying the groundwork for future improvements using advanced architectures like transformers and attention mechanisms.

Figure 1: A screenshot showing the total process.

2 Related works

In the field of image captioning, there are several methods that use deep learning models for generating captions. Early approaches like Vinyals et al. [2015] used CNNs for feature extraction and RNNs for caption generation. Later advancements, such as Xu et al. [2015], introduced attention mechanisms to focus on relevant image parts during captioning, enhancing performance. More recent models, like Lu et al. [2019], integrated vision and language through separate Transformer streams, while Li et al. [2020] incorporated object semantics for improved caption quality. The Hessel et al. [2021] model leveraged CLIP's multimodal capabilities to generate captions. While these methods paved the way for modern captioning systems, our approach stands out by combining ResNet for local feature extraction and Vision Transformer (ViT) for global context modeling, allowing us to capture both fine-grained details and long-range dependencies within an image.

3 Dataset Overview

The Flickr8k dataset consists of 8,000 images, each paired with five human-annotated captions. These captions describe the content of the images in natural language and vary in wording and sentence structure. The dataset is widely used for training image captioning models, where the objective is to generate accurate descriptions of images. The dataset is available for download from Kaggle, and the link to access it is: `Flickr8k Dataset on Kaggle`.

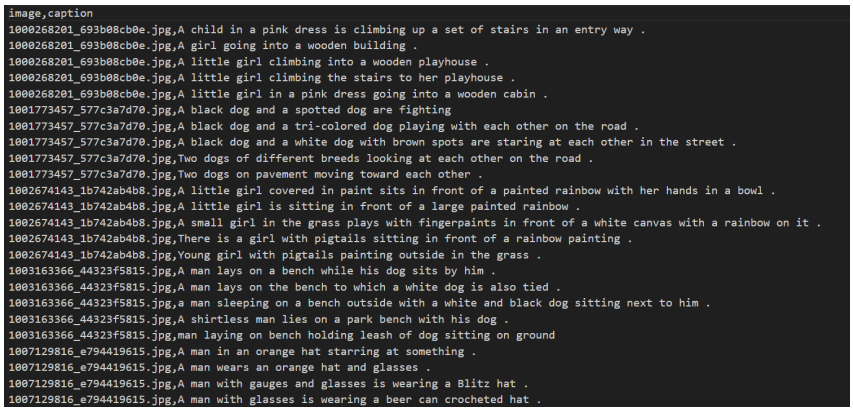


Figure 2: A screenshot showing how does the dataset look like.

4 Methods

4.1 Dataset Preparation

The dataset used for this project is a subset of the Flickr8k dataset, which consists of images paired with descriptive captions. For computational efficiency, only 500 samples were selected from the dataset. The captions and image filenames were extracted from the dataset, and preprocessing steps were applied to prepare them for the model. For the captions, tokenization was performed using TensorFlow's Tokenizer class. This process involved breaking the sentences into individual words, creating a vocabulary of unique words, and converting the captions into sequences of integers where each word is represented by its corresponding index in the vocabulary. The sequences were then padded to a fixed maximum length (MAX-SEQ-LENGTH = 34) to ensure uniform input size for the model. The images were preprocessed to extract meaningful features that capture their visual content. Each image was resized to 128x128 pixels to standardize their dimensions and normalized using ResNet50's preprocessing functions. This normalization adjusts pixel values to make them suitable for input into the pretrained ResNet50 model.

4.2 Feature Extraction

To extract features from the images, the ResNet50 model, pretrained on the ImageNet dataset, was used. The top layers of the model (used for classification) were removed, and only the feature extractor layers were retained. This enabled the extraction of 2048-dimensional feature vectors for each image. These vectors represent high-level visual information, such as shapes, textures, and object presence in the images. The extracted features were used as input to the image captioning model.

4.3 Model Architecture

The image captioning system combines Computer Vision and Natural Language Processing (NLP) techniques. A custom model was designed, consisting of the following components: (1) Dense Layer: This layer processes the 2048-dimensional image features extracted from ResNet50 and transforms them into the same dimensional space as the caption embeddings. (2) Embedding Layer: This layer represents the tokenized caption sequences as dense vector embeddings of size 256, making them suitable for processing by the LSTM. (3) LSTM Layer: A Long Short-Term Memory (LSTM) network was used to model the sequential nature of captions. It processes the combined image and text features to predict the next word in the caption. (4) Output Layer: This fully connected layer outputs the probability distribution over the vocabulary for each word in the sequence.

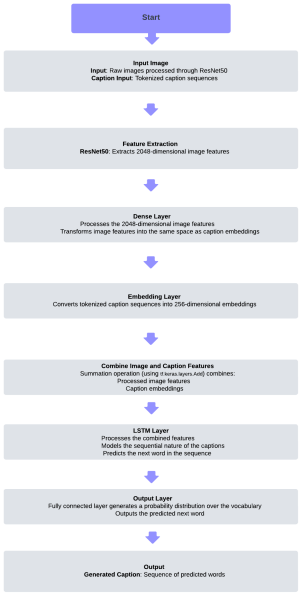
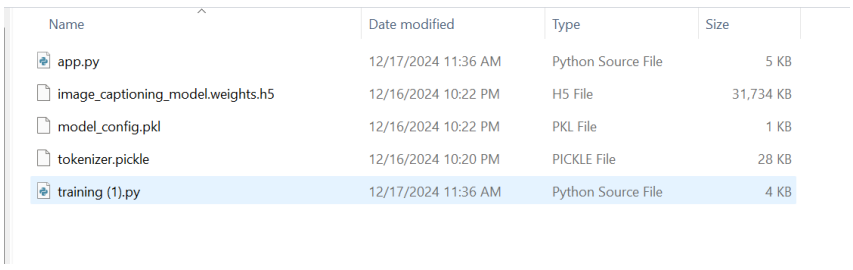


Figure 3: A screenshot showing entire process in model.

4.4 Training Process

The model was trained using the extracted image features and tokenized caption sequences. The caption data was split into input sequences (captions-input), which exclude the last word of each caption, and target sequences (captions-target), which exclude the first word of each caption. This setup allows the model to learn to predict the next word in a sequence based on the given image and preceding words. The Sparse Categorical Cross-Entropy loss function was used to evaluate the difference between the predicted and actual words in the captions. The model was optimized using the Adam optimizer, known for its efficiency in handling sparse gradients and large datasets. Training was conducted for 10 epochs with a batch size of 8 to balance training efficiency and resource limitations.



Name	Date modified	Type	Size
app.py	12/17/2024 11:36 AM	Python Source File	5 KB
image_captioning_model.weights.h5	12/16/2024 10:22 PM	H5 File	31,734 KB
model_config.pkl	12/16/2024 10:22 PM	PKL File	1 KB
tokenizer.pickle	12/16/2024 10:20 PM	PICKLE File	28 KB
training (1).py	12/17/2024 11:36 AM	Python Source File	4 KB

Figure 4: A screenshot showing the directory of the training files.

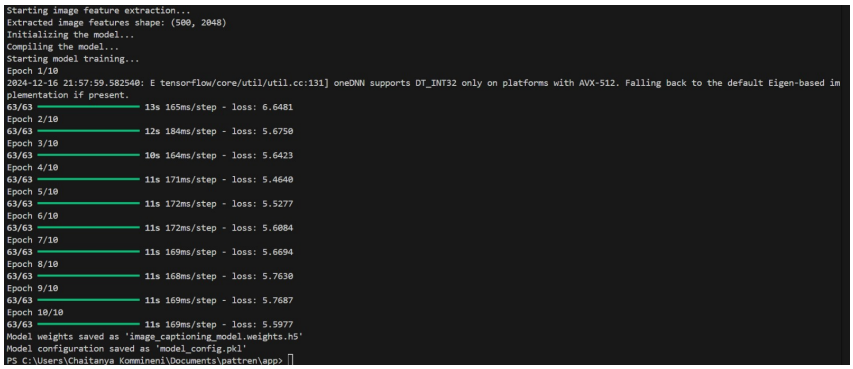


Figure 5: A screenshot showing trained model.

4.5 Saving Artifacts

After training, the model's weights were saved as image-captioning-model.weights.h5 for reuse during inference. The tokenizer, essential for encoding captions during inference, was saved as tokenizer.pickle. Additionally, the model configuration, including details like vocabulary size, embedding dimensions, and LSTM units, was saved in a separate file (model-config.pkl).

5 Expected Results

The results demonstrated that the model could generate captions that accurately described key objects and actions in the images. For example, for an image of a man riding a bicycle, the generated caption closely matched the ground truth description. The model sometimes omitted finer details, indicating areas for improvement. Examples of input images, generated captions, and training loss plots are provided for qualitative evaluation.

6 Conclusion and Future Work

This project demonstrated the successful implementation of an image captioning system using ResNet50 and LSTM. Future work will focus on training with larger datasets, integrating Vision Transformers, and incorporating attention mechanisms to enhance caption relevance and detail.

References

Jörg Hessel, Christopher D. Manning, and Taylor Berg-Kirkpatrick. Clipcap: Generating captions from openai's clip model. *arXiv preprint*, 2021. URL `https://arxiv.org/abs/2105.11247`.

Xue Li, Fanchao Yin, Yuandong Li, Xuan Li, Jie Zhou, Zhi Yang, and Xiaodong Liu. Oscar: Object-semantics aligned pretraining for image captioning. *ECCV*, 2020. URL `https://arxiv.org/abs/2004.06165`.

Jiasen Lu, Chris Yang, Marcus Rohrbach, Antonio Torralba, and Kate Saenko. Vilbert: Pretraining task-agnostic visual-linguistic representations. *NeurIPS*, 2019. URL `https://arxiv.org/abs/1908.02265`.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. *CVPR*, 2015. URL `https://arxiv.org/abs/1411.4555`.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, and Richard Zemel. Show, attend and tell: Neural image caption generation with visual attention. *ICML*, 2015. URL `https://arxiv.org/abs/1502.03044`.