Computer Vision Final Project — Image Captioning

Mounya Inampudi, Rupesh Swarnakar, Junaid Ahmed Mohammed

2025-11-28

Contents

1 Abstract

In this project, we implement a complete image captioning system using a classical encoder–decoder framework enhanced with **soft attention**, following the **Show, Attend, and Tell** approach. We employ a pre-trained **ResNet-101 CNN** as the encoder to extract dense spatial features, and a **Bahdanau-style attention LSTM** as the decoder to sequentially generate captions. The pipeline includes detailed steps for data preprocessing, vocabulary design, CNN feature extraction, attention-based decoding, model training with teacher forcing, evaluation using BLEU metrics, and inference via beam search. Experiments on the **Flickr8k dataset** yielded a **BLEU-4 score of 0.2818**, which aligns well with established attention-based baselines. This report outlines the dataset, preprocessing strategy, model design, training procedure, evaluation metrics, results, key challenges, and potential avenues for future work.

2 1. Introduction

Image captioning is a multimodal task that bridges computer vision and natural language processing. It requires understanding the semantic content of an image and generating coherent textual descriptions. Traditional approaches adopt an **encoder–decoder** framework: a CNN encoder extracts visual features, while an RNN decoder generates captions sequentially. Incorporating **attention mechanisms** allows the model to dynamically focus on relevant regions in an image during decoding , improving both interpretability and descriptive accuracy, especially in complex scenes. In this work, we build an end-to-end pipeline encompassing dataset preprocessing, feature extraction, vocabulary construction, attention-based decoding, model training, and evaluation, providing hands-on experience in multimodal deep learning.

3 2. Dataset Description

The **Flickr8k dataset** consists of 8,000 images, each paired with five human-annotated captions. The dataset covers diverse real-world scenarios, including: - People performing various actions (e.g., running, sitting, interacting) - Animals such as dogs, horses, and elephants - Indoor and outdoor scenes - Vehicles, sports, food, and miscellaneous objects

Actual: a man on a motorcycle touches his knee to the ground during a sharp turn

The richness of the captions allows the model to learn robust visual–linguistic mappings. Evaluation is performed using **BLEU metrics**, which measure both lexical correctness and sentence fluency.

4 3. Data Preprocessing

Proper preprocessing is essential for stable training and accurate caption generation.

4.1 3.1 Caption Cleaning

Captions were normalized to reduce noise: 1. Converted all text to **lowercase** to avoid duplications caused by casing. 2. Removed **punctuation, special characters, and numbers**. 3. Collapsed multiple spaces into a single space. 4. Filtered out very short or meaningless tokens (e.g., single letters). 5. Added special tokens: `<start>` to indicate the beginning, and `<end>` to mark the end of a caption.

This ensures a standardized structure for each caption, which is crucial for consistent model training.

4.2 3.2 Vocabulary Construction

A word frequency threshold of ≥ 5 occurrences was applied, selecting only meaningful tokens and limiting vocabulary size to around 2,000 words. Special tokens `<pad>`, `<start>`, `<end>`, and `<unk>` were manually included.

Benefits of frequency thresholding: - Reduces overfitting on rare words - Stabilizes the softmax output by limiting dimensionality - Improves training efficiency and reduces memory usage

4.3 3.3 Image Preprocessing

- Images were resized to 256×256 and normalized using ImageNet mean and standard deviation.
- Training images underwent **random cropping and horizontal flipping**; validation/test images used center cropping.
- Final tensors had shape $3 \times 224 \times 224$, suitable for the ResNet-101 encoder.

5 4. Model Architecture

The model follows the **Show, Attend, and Tell** paradigm, composed of three main components: encoder, attention mechanism, and decoder.

5.1 4.1 Encoder — ResNet-101 CNN

We used a pre-trained **ResNet-101** as the encoder. The global pooling and final fully connected layers were removed to retain spatial feature maps. Adaptive average pooling reshaped the features into a 14×14 grid (196 regions), each with 2048 dimensions.

Fine-tuning strategy: - **Freeze early layers** to preserve generic visual features. - **Fine-tune the last two convolutional blocks** for Flickr8k-specific adaptation.

Output shape: `(batch_size, 196, 2048)`

5.2 4.2 Attention Mechanism — Bahdanau Additive Attention

The attention module allows the decoder to focus on relevant image regions at each time step: 1. Compute alignment scores between decoder hidden state and encoder features. 2. Apply softmax to obtain attention weights. 3. Compute the context vector as a weighted sum of encoder features. 4. Concatenate the context vector with the current word embedding before feeding it to the LSTM.

Advantages: - Provides visual interpretability - Handles complex or cluttered scenes - Improves descriptive accuracy by emphasizing salient objects

5.3 4.3 Decoder — LSTM with Attention

The decoder is an **LSTMCell** with attention. Key components: - Embedding layer (256-dimensional) - LSTMCell for sequential decoding - Attention gate modulating the context vector - Linear + softmax layer for vocabulary prediction - Dropout (0.5) for regularization

Teacher forcing was used during training to feed ground-truth words to the model, improving convergence and stability.

5.4 4.4 Beam Search Inference

Beam search with a size of 3–5 was used during inference to maintain multiple caption hypotheses, producing more fluent and accurate captions compared to greedy decoding.

Example: - Greedy: "A dog running in the grass." - Beam: "A brown dog is running across a grassy field."

6 5. Training Procedure

6.1 5.1 Training Configuration

Component	Value
Loss Function	Cross-Entropy + Attention Regularization
Optimizer	Adam
Learning Rate	4e-4
Batch Size	32
Epochs	60
Gradient Clipping	5.0
Hardware	NVIDIA RTX GPU

6.2 5.2 Overfitting Prevention

- Data augmentation (random cropping/flipping)
- Dropout in decoder
- Attention regularization
- Limited CNN fine-tuning
- Monitoring validation BLEU scores

7 6. Evaluation Metrics

BLEU-1 to BLEU-4 metrics were used to measure n-gram precision, capturing both object-level accuracy and sentence-level fluency.

8 7. Results

8.1 7.1 BLEU Score Summary

Metric	Score
BLEU-1	0.6689
BLEU-2	0.5038
BLEU-3	0.3818
BLEU-4	**0.2818**

8.2 7.2 Final Image Captions

Here we present the three final results for the actual and generated captions.

Generated: a man is playing with a dog on the beach
Actual: a man holds a ball in the air for a brown dog to catch on the beach

Generated: a person is surfing in the ocean
Actual: a surfer on a blue surfboard is falling off of it as he hits a wave

Generated: a little boy is standing in front of his head
Actual: a child with a skull on his shirt is sitting in front of some plants and a building and is holding onto handlebars

9 8. Discussion

9.1 Strengths:

- Accurate object recognition and description
- Attention provides visual interpretability
- Beam search improves fluency
- Teacher forcing stabilizes learning

9.2 Challenges:

- Small or occluded objects may be missed
- Limited vocabulary can lead to repetitive captions
- LSTM struggles with **long-range dependencies**
- Small dataset increases overfitting risk

9.3 Potential Improvements:

- **Transformer-based decoder**
- Vision Transformer or DETR encoder
- **CLIP embeddings** for multimodal alignment
- Larger datasets (MS-COCO)
- Additional metrics like CIDEr and SPICE
- **Scheduled sampling** to reduce exposure bias

10 9. Conclusion

We presented a full pipeline for image captioning using an encoder–decoder framework with attention. The BLEU-4 score of 0.2818 demonstrates reasonable performance and establishes a baseline. This framework lays the groundwork for further improvements using transformer-based models, larger datasets, and advanced multimodal embeddings.

11 References

Xu, K. et al., "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention," ICML, 2015.

He, K. et al., "Deep Residual Learning for Image Recognition," CVPR, 2016.

Papineni, K. et al., "BLEU: A Method for Automatic Evaluation of Machine Translation," ACL, 2002.

Flickr8k Dataset, Kaggle

PyTorch Documentation: https://pytorch.org