OPEN ACCESS

Volume: 13

Special Issue: 2

Month: January

Year: 2026

E-ISSN: 2582-0397

P-ISSN: 2321-788X

Citation:
K, Dhamayandhi, et al. "Automatic Image Captioning Using Convolutional Neural Network and Long Short-Term Memory Techniques." *Shanlax International Journal of Arts, Science and Humanities*, vol. 13, no. 2, 2026, pp. 224–36.

DOI:
https://doi.org/10.34293/sijash.v13iS2-i4-Jan.10607

Automatic Image Captioning Using Convolutional Neural Network and Long Short-Term Memory Techniques

Dhamayandhi K
Department of Information Technology
Avinashilingam Institute for Home Science and Higher Education for Women
Coimbatore, Tamil Nadu, India

Dr. T. Jayamalar
Department of Information Technology
Avinashilingam Institute for Home Science and Higher Education for Women
Coimbatore, Tamil Nadu, India

Dr. N. Krishnaveni
Department of Information Technology
Avinashilingam Institute for Home Science and Higher Education for Women
Coimbatore, Tamil Nadu, India

Lavanya C
Department of Information Technology
Avinashilingam Institute for Home Science and Higher Education for Women
Coimbatore, Tamil Nadu, India

Abstract
Image captioning is the task of automatically generating natural language descriptions based on image content, with applications in social media, e-commerce, and content creation. Classic approaches would involve reinforcement learning and multimodal transformers, but they demand large datasets and huge computational resources and are not suited to handle complex scenes. To satisfy these challenges, this work adopts a hybrid approach that employs the task of CNNs for visual feature extraction and uses LSTMs for sequential caption generation. More in detail, rich image features were extracted using a pre-trained Inception V3 model, while an LSTM was adopted for synthesizing image captions. Image caption generation was realized based on both greedy and beam search strategies. Finally BLEU score and visualization demonstrate the effectiveness of the model in generating captions similar to the reference descriptions

Keywords: Image Captioning, Convolutional Neural Network (CNN), Long Short-Term Memory Network (LSTM), Computer Vision, Bleu Score

Introduction

The task of automatically describing the contents of an image in a human-like manner is a primary concern in the area of image captioning. The application is important in contemporary digital systems, allowing for dynamic interactions in various fields such as social media, e commerce, and media automation. Traditional techniques, including reinforcement-based systems and multimodal transformers, often depend on extensive

datasets and powerful hardware, which can be resource-intensive and less accessible in many real-world scenarios. A more robust and efficient solution involves integrating Convolutional Neural Networks (CNNs) with Long Short Term Memory (LSTM) networks. CNNs are adept at analyzing visual data and identifying key features, while LSTMs are designed to understand and generate well-formed text sequences. By combining these strengths, this approach provides improved accuracy, better handling of visual complexity, and a more streamlined method for generating descriptive captions from images. The proposed project focuses on developing an automatic image captioning system using deep learning techniques. A pre-trained InceptionV3 network is used to obtain the image features from the input images, and an LSTM network is used to obtain the captions from images. To process and tokenize the input texts, the model makes use of the obtained image features to train it. Whereas during caption generation, it can either rely on greedy or beam search methods. Greedy search chooses words based on their maximum probability at different time steps to generate the caption step by step, whereas beam search searches for different words and only chooses the top k most likely paths to generate the caption. To test how well the model performs, it makes use of its BLEU scores by comparing them to reference captions. Moreover, image visualization capabilities are used to show how it generates images and their respective captions. Despite its importance, manually writing captions for the vast and growing number of images shared online every day is unrealistic and time-consuming. With millions of images uploaded daily to platforms like social media and e commerce websites, content creators face difficulties in meeting the demand for fast, accurate, and consistent captions. This can impact productivity and limit user engagement. For businesses, especially in the online retail space, well-formed image descriptions are critical for enhancing search engine visibility and improving the overall customer experience. The primary objective of this project is to develop an automatic image captioning system that effectively integrates CNNs and LSTMs to produce meaningful and accurate descriptions of images. Supporting objectives include extracting relevant image features using a pre-trained InceptionV3 model, generating descriptive captions through LSTM networks, constructing a functional model that unites vision and language components, and implementing decoding techniques such as greedy search and beam search to optimize caption quality and relevance.

Related Work

This section outlines different image captioning models, summarizing the main techniques used, their advantages, and their challenges. The approaches range from basic encoder decoder models to more advanced methods using attention, transformers, and graph-based networks. Khaing, P. P. [1] conducted a comparative study titled “Attention-based Deep Learning Model for Image Captioning,” implementing an attention-based CNN-RNN framework. This study investigates how different deep learning architectures enhanced with attention mechanisms perform in the task of generating descriptive and contextually relevant image captions. It contributes by analyzing model behaviors across different datasets, but the scalability and performance vary depending on the image complexity and dataset characteristics. Yu et al. [2] introduced the work “Multimodal Transformer with Multi-view Visual Representation for Image Captioning,” employing a multimodal transformer architecture integrated with multi-view visual representations. This method enhances visual understanding by capturing features from different perspectives, leading to improved caption quality. However, the added complexity of handling multiple visual streams can significantly increase computational demands. Shaikh, F. [3] developed the study “Solving an Image Captioning Task Using Deep Learning,” utilizing a CNN LSTM pipeline with an attention mechanism. The model serves as a more practical introduction to image captioning, where it shows in a realistic manner how deep learning frameworks can be applied to an application. However, the simplicity of this model and the limited experimentation might weaken complex or diverse datasets’ performances. Xiao et al. [4] proposed “Deep Hierarchical Encoder Decoder Network for Image Captioning,” which employs a hierarchical encoder-decoder structure to generate context rich captions. The model produces more coherent and detailed descriptions by capturing multi-level image

features. However, its hierarchical structure increases computational complexity and requires substantial training effort. Zeng et al. [5] presented the "S2 Transformer for Image Captioning," which involves a spatial and scale-aware transformation network for captioning. The model improves contextual understanding and attention distribution, leading to state-of-the-art results. Despite its strengths, the transformer may struggle with fine-grained object details in visually dense scenes and typically demands high computational resources. Sharma and Srivastava [6] introduced a novel method combining Graph Neural Networks with multilevel attention mechanisms in their work "Graph Neural Network-based Visual Relationship and Multilevel Attention for Image Captioning." This approach enhances caption quality by modeling object relationships and visual hierarchies. Cho and Oh [7] presented "Generalized Image Captioning for Multilingual Support," utilizing a multilingual encoder decoder model capable of generating captions in multiple languages. The system promotes broader applicability across linguistic contexts. However, maintaining semantic consistency across languages is challenging, and the model's performance can vary based on language-specific nuances. Lin et al. [8] proposed a domain-specific method titled "Skin Medical Image Captioning Using Multi-label Classification and Siamese Network." The model leverages a Siamese architecture and multi-label learning for generating descriptive captions tailored to medical images. While it improves accuracy in the medical domain, its design limits generalizability to other image types. Osman et al. [9] presented "A Survey on Attention-based Models for Image Captioning," offering a comprehensive analysis of various attention-based techniques used in the field. The work synthesizes recent advancements and categorizes methods based on architectural design and performance. However, as a survey, it lacks experimental validation or implementation of proposed ideas. Zhang et al. [10] developed a "Zero-Shot Image Caption Inference System Based on Pretrained Models," leveraging the use of vision and language pretrained models to perform image captioning without needing to be trained on a particular domain. This zero-shot approach enables flexible and adaptive captioning across domains. However, its reliance on pretrained knowledge can limit accuracy when dealing with novel objects or rare visual contexts.

Key Observation

Recent image captioning methods such as reinforcement learning and multimodal transformers aim to improve caption accuracy and robustness. While some models focus on computational efficiency, others accept increased complexity to achieve higher performance. Despite advancements, challenges remain in handling dataset diversity, providing multilingual support, and reducing over reliance on pre-trained models. Nevertheless, image captioning techniques continue to find strong applicability in domains such as healthcare, e-commerce, and accessibility systems.

Methodology

Methodology Overview

The Image Caption Generator project intends to create an automatic system for generating descriptive image captions based on deep learning methodologies. It uses the Flickr8k dataset, which consists of a folder containing 8,091 images and a separate caption file. In the caption file, each image ID is associated with five different captions. The project begins with data preparation—loading and pre-processing the captions to ensure consistency and clarity. A tokenizer is then applied to transform words into unique integer indices, enabling the model to interpret and process the vocabulary effectively. For feature extraction, a pre-trained model of Inception V3 is used, and a 2048-dimensional feature vector for every image is obtained. This vector is fed into a Long Short-Term Memory (LSTM) network. The whole architecture is based on a Convolutional Neural Network (CNN) model for extracting the visual features and an LSTM model for generating the sequence. The overall methodology of the system is illustrated in Fig. 1.The model is trained with a generator which produces batches of image features and their respective text sequences, training the loss function in the process. The model, once trained, outputs captions for test images using greedy search or beam search.

Performance is measured in terms of BLEU scores, comparing generated captions to reference captions. The project also features visualization tools that show each image with its generated caption, enabling qualitative assessment of the results. This project illustrates the application of computer vision and natural language processing to develop an efficient image captioning system.

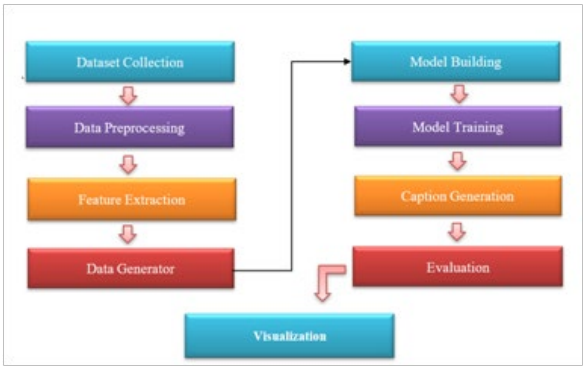


Figure 1 Methodology Diagram

Dataset Collection

The Flickr8k dataset, which was downloaded from Kaggle, is utilized for this project. The dataset consists of two main components: an images folder and a captions file. The image folder contains 8,091 .jpgs. These represent a large number of common scenes and activities; thus, the dataset is good for training models that generate descriptions or contextually relevant captions. The captions file, typically named captions.txt, provides five different textual descriptions for each image. These captions are expressed in natural language and are designed to convey a description of a corresponding image. Each caption is associated with an image identifier, enabling efficient mapping of images to their respective descriptions.

Data Preprocessing

This step ensures that the text data are clean, consistent, and in a usable form to be trained on the model. The key steps in the preprocessing pipeline are as follows:

Loading Captions:

Captions are read from the file and converted to lowercase to maintain uniformity. The function also skips the first line and returns a list of raw captions.

Tokenizing Captions

A tokenizer is created to convert each word into a numerical index, enabling the model to process the textual data. The tokenizer learns the vocabulary from the list of captions.

Cleaning Text

This function removes unwanted punctuation, numerical characters, and extra spaces from the captions, resulting in a cleaner and more standardized dataset.

Cleaning Captions

Each caption is further refined by extracting the relevant portion, often done by splitting the text using a delimiter such as a comma.

Creating Caption IDs

A new list is formed that combines the image ID with its corresponding cleaned caption. Each caption is wrapped with special tokens, such as “start” and “end,” to indicate the beginning and end of the sentence.

Feature Extraction

Feature extraction is a crucial step in machine learning and computer vision, where relevant information (or features) is extracted from raw data to be used for further analysis or modeling. In the scenario of image processing, the role of feature extraction is the identification and separation of key features or patterns in the images that assist in the processes of classification or recognition or captioning.

Image preprocessing

The input image is loaded and resized to a 299×299 pixel matrix corresponding with the expected input dimension for the InceptionV3 architecture. It is then converted into a numerical array format and reshaped to add an extra dimension, simulating a batch of images. The pixel values are normalized to fit the range required by InceptionV3, typically between −1 and 1. After these preprocessing steps, the image becomes suitable for feature extraction by the model.

InceptionV3 for Feature Extraction

Inception V3 is an optimized convolutional neural network created by Google. Inception V3 is trained on the enormous dataset of the ImageNet library, which has more than a million pictures belonging to 1,000 classes. Inception V3 is a deep learning algorithm commonly employed in the field of computer vision. In numerous applications in the field of computer vision, ranging from the description of pictures to studying the similarity of pictures, the final output is not needed. Something more significant than classification must be attained in the case of an image. Instead of the final layer in the Inception V3 algorithm, the penultimate layer provides a 2048-dimensional vector.

Data Organization for Training, Validation, and Testing

The system extracts visual features from all images and organizes them into three categories: training, validation, and testing. This categorization is essential for proper data management and ensuring the model receives the right type of data for each phase of training and evaluation. Once extracted, the features are stored separately for each category, allowing for easy access when needed during training and evaluation.

Typically, The dataset is split into three parts: 80% for training, 10% for validation, and 10% for testing. Such a distribution ensures that the model gets enough examples to learn properly, while at the same time it keeps enough data for both validation and testing. By storing the features in organized dictionaries, the system enables efficient use of the data throughout the model’s development process.

Data Generator

This process involves batching image-caption pairs, extracting features from images using a pre-trained InceptionV3 model, and tokenizing captions into input output pairs. TensorFlow’s tf.data API is used to efficiently handle the data pipeline, ensuring optimal performance through batching, shuffling, and prefetching. The goal is to prepare data for training a model, such as for image captioning, where it predicts the subsequent word of an image caption given its features and the former words.

Model Building

The model is designed for image captioning, combining a CNN-based encoder with an LSTM-based decoder. The encoder extracts features from the input image, normalizes them, and processes them through dense layers to generate a fixed-size feature vector. The decoder takes the sequence of words (caption)

as input, embeds them into dense vectors, and passes them through an LSTM to capture the sequential dependencies between words. The output of the encoder and the feature vectors of the LSTM network are then combined and used for predicting the next element in the sequence with the help of a softmax activation function. The model uses categorical cross entropy loss with the Adam optimizer for optimization.

CNN with LSTM

- CNN Encoder: The encoder learns the pre-extracted image features through BatchNormalization and Dense layers. The image is converted to a compact feature vector that describes principal visual information.
- LSTM Decoder: The decoder accepts the input caption sequence, embeds every word into a vector, and passes the sequence through an LSTM layer. The layer identifies dependencies among words within the sequence.
- Concatenating Encoder and Decoder: The features of the image from the encoder and the output from the LSTM decoder are concatenated. This concatenated representation is fine-tuned with Dense layers and employed for predicting the next word in the sequence of captions using a softmax layer.

Model Training

This training setup uses two important callback strategies to improve the training process of the image captioning model:

Early Stopping Callback: This callback function ensures early stopping of the training process when no improvement in the validation loss occurs. It does this by setting patience = 3, such that training stops when no improvement is observed in 3 epochs. Besides, it ensures the best model parameters are used and no overfitting.

Learning Rate Scheduler Callback: A custom learning rate schedule is defined using an exponential decay function (lr * e^-0.6) to gradually reduce the learning rate with each epoch. This helps the model converge more smoothly and avoid overshooting optimal points.

The model is trained using the fit() function for up to 15 epochs, but with early stopping, it may halt sooner if the validation loss stops improving. The datasets are split into batches for both training and validation to ensure efficient memory usage and performance.

Caption Generation

After training, the model is able to generate captions for images. This is done using methods such as greedy search or beam search, where the model predicts the next word in the sequence based on the previously generated words and the visual features of the image.

Greedy Search and Beam Search

Greedy Search and Beam Search are methods used during caption generation (or any sequence prediction) to decide how to pick the next word.

Greedy Search: The greedy generator function generates captions starting with ‘start’, and at each step, it predicts the next word having the highest probability according to the model. It keeps adding the predicted words until it generates ‘end’ or reaches the maximum length, finally returning the complete caption without ‘start’ and ‘end’.

Beam Search: The beam_search_generator function generates a caption by exploring multiple possible word sequences at each step (instead of picking just the highest probability word like greedy search). It keeps the top K_beams most likely sequences based on their cumulative probability. At every step, it expands all current sequences by predicting next words, updates their probabilities, and keeps only the best K candidates. It continues until the maximum caption length is reached or an ‘end’ token appears, and then returns the most probable complete caption.

Evaluation

BLEU is an abbreviation for "Bilingual Evaluation Understudy". This is an evaluation measure used to judge how well automatically generated texts, like translation pairs or photo captions, are when compared to human-prepared reference texts. The value of a BLEU score is always between 0 and 1.

BLEU-1 and BLEU-2

- BLEU-1, which is more simpler, verifies word-level similarities within the created text and the actual texts.
- BLEU-2 evaluates two-word sequences (bigrams), making it slightly more challenging as it considers context in addition to individual words.

The expected BLEU score ranges for image captioning tasks are shown in Table 1.

Table 1 Normal BLEU Score Range for Image Captioning

BLEU TYPE	NORMAL RANGE
BLEU-1	0.5 -0.7
BLEU-2	0.3-0.5

BLEU Score Calculation

The calculation of BLEU scores involves several key steps to evaluate the performance of machine-generated captions against human-generated references. The process includes tokenization and the computation of BLEU-1 and BLEU-2 scores.

Steps Involved

Tokenization

- References: The human-generated captions are tokenized into individual words, creating a list of tokens for each reference caption.
- Hypotheses: The machine-generated captions (from Greedy and Beam Search) are also tokenized into individual words for comparison.

BLEU Score Calculation

- BLEU-1: Computes the matching of single-word (unigram) overlaps between automatically generated and human-written reference captions.
- BLEU-2: Calculates the overlap of two-word sequences (bigrams), considering context.

Comparison

The BLEU-1 and BLEU-2 scores are calculated for both Greedy Search and Beam Search algorithms to give an assessment of similarity between machine-made captions and human written reference captions.

Visualization

In this process, random test images are selected, and their corresponding human-generated captions are cleaned by removing the "start" and "end" tokens. Two captioning methods, Greedy Search and Beam Search, are then used to generate captions. BLEU scores (BLEU-1 and BLEU-2) were computed in order to compare the similarity between the proposed captions and human captions. Images are displayed alongside their respective captions, with the BLEU scores shown for each. This provides both a visual and quantitative evaluation of the model's performance.

Experimental Results and Discussion

Loading and Tokenizing Captions

Loading Caption: Captions are loaded by function load_captions(file_path), which reads the captions from a specified file, omits the first line, and lowercases each caption. A list of these captions was then returned.

Tokenizing Captions: The function tokenize_captions(captions) turns the captions into an integer sequence using a tokenizer. An integer ID was assigned to each distinct word in the caption. This is helpful when processing captions in machine-learning models.

Cleaning the Caption

The captions are cleaned by removing punctuation, numbers, and extra spaces, ensuring the text is formatted for further processing. A few examples of the cleaned captions after preprocessing are shown in Fig. 2.

```
['a child in a pink dress is climbing up a set of stairs in an entry way',
 'a little girl climbing into a wooden playhouse',
 'a little girl in a pink dress going into a wooden cabin',
 'a black dog and a tricolored dog playing with each other on the road',
 'two dogs of different breeds looking at each other on the road',
 'a little girl covered in paint sits in front of a painted rainbow with her hands in a bowl',
 'a small girl in the grass plays with fingerpaints in front of a white canvas with a rainbow on it',
 'young girl with pigtails painting outside in the grass']
```

Figure 2 Sample cleaned captions after preprocessing

Processing captions with Start and End

A “start” token is inserted at the beginning of each image caption, and an “end” token is appended at the end. This process helps the model identify the boundaries of each caption during training. The modified captions are then stored in a list along with their corresponding image identifiers. A sample of the processed captions is shown in Fig.3.

```
(['1000268201_693b08cb0e.jpg\tstart a child in a pink dress is climbing up a set of stairs in an entry way end\n',
  '1000268201_693b08cb0e.jpg\tstart a little girl climbing the stairs to her playhouse end\n',
  '1001773457_577c3a7d70.jpg\tstart a black dog and a tricolored dog playing with each other on the road end\n',
  '1001773457_577c3a7d70.jpg\tstart two dogs on pavement moving toward each other end\n',
  '1002674143_1b742ab4b8.jpg\tstart a small girl in the grass plays with fingerpaints in front of a white canvas with a rainbow on it end\n',
  '1003163366_44323f5815.jpg\tstart a man lays on a bench while his dog sits by him end\n',
  '1003163366_44323f5815.jpg\tstart a shirtless man lies on a park bench with his dog end\n'],
 40455)
```

Figure 3 Processing captions with start and end tokens

Visualizing Images with Captions

A collection of images and their corresponding captions was visualized using a custom function. After grouping captions by image identifiers, the function displays each image along with its associated captions. The number of images shown can be adjusted dynamically. Each image is plotted, and its captions are arranged in a grid format below it, as illustrated in Fig. 4.

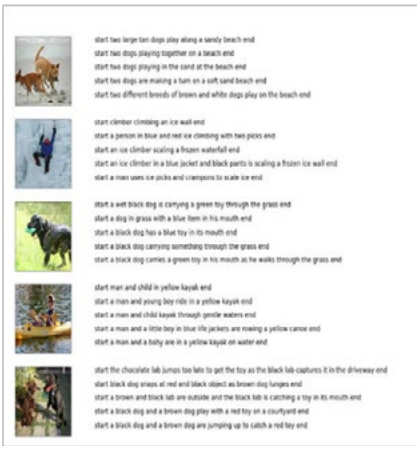


Figure 4 Visualizing Images with Captions

Caption length distribution visualization

The process generates a histogram to visualize the distribution of caption lengths (in terms of word count) within the dataset, providing insights into the variation across captions. The resulting distribution is shown in Fig. 5.

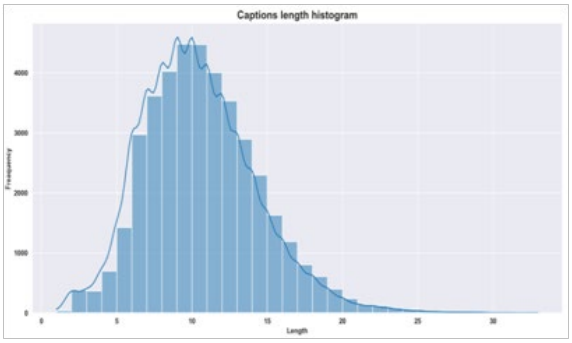


Figure 5 Caption length distribution

Word occurrences visualization in captions

Combines all captions into one large string and counts the frequency of each word using a counter function. The 30 most common words were selected and their frequency values were normalized to a range between 0 and 1. A pie chart was generated, where each slice represents the frequency of a word. The color of the slices was based on the frequency of the words, as shown in Fig. 6.

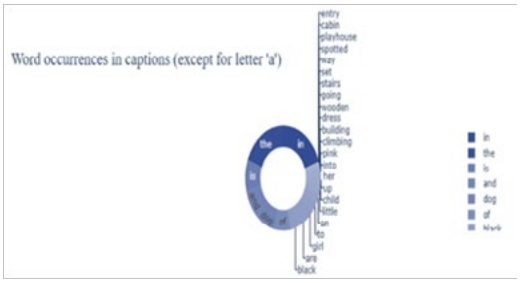


Figure 6 Word occurrences visualization in captions

Tokenization and Vocabulary Size Calculation

This step tokenizes the captions and converts them into unique numerical indices. The vocabulary size is then determined by counting the total number of unique words, with an additional entry included for padding. This calculated size is essential for constructing the embedding layer of the model and managing the entire vocabulary.

Image processing

An image is loaded and resized to 299×299 pixels, then converted into an array. An additional dimension is added to match the input shape required by the model, followed by InceptionV3-specific preprocessing. This prepares the image for further analysis or model input, as shown in Fig.7.

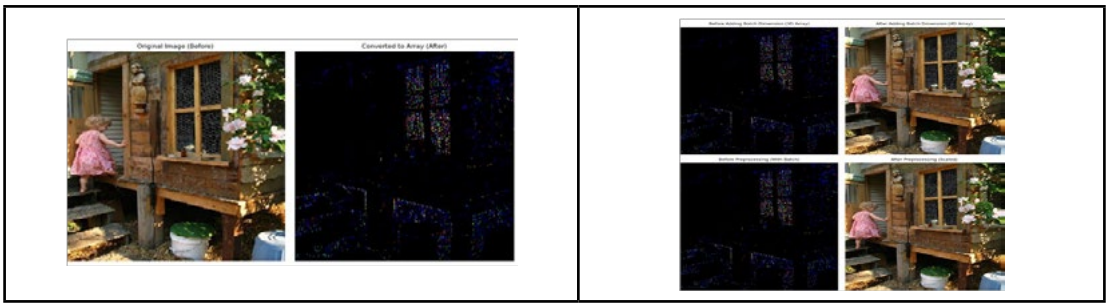


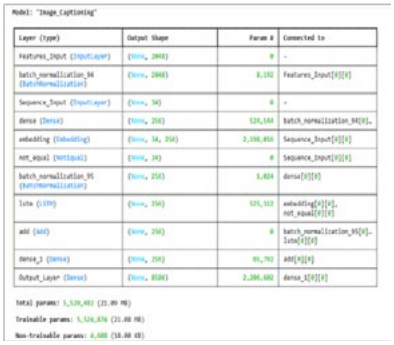
Figure 7 Image processing steps (a) Original image and conversion to array, (b) Image representation before preprocessing with batch dimension, and final preprocessed image after InceptionV3-specific scaling

Image Feature Extraction with InceptionV3

The InceptionV3 model is employed for obtaining image features, which involves preprocessing the image by resizing and normalizing and then feeding the image into the model. This is followed by stripping the final output layer and using the output of the second-to-last layer as the features of the image.

Building the Image Captioning Model

A model is constructed to process image features and caption sequences. The encoder processes image features using dense and batch normalization layers. The decoder is a neural network that processes caption inputs using an embedding layer followed by a Long Short-Term Memory (LSTM) network. The outputs of these components are combined and passed through dense layers to predict the next word in the sequence. The model is trained using a categorical cross-entropy loss function with the Adam optimizer and learning rate clipping, as illustrated in Fig. 8.


Model: "Image_Captioning"

Layer (type)	Output Shape	Param #	Connected to
Features_Input (InputLayer)	(None, 2048)	0	-
batch_normalization_94 (BatchNormalization)	(None, 2048)	8,192	Features_Input[0][0]
Sequence_Input (InputLayer)	(None, 34)	0	-
dense (Dense)	(None, 256)	524,544	batch_normalization_94[0]...
embedding (Embedding)	(None, 34, 256)	2,198,016	Sequence_Input[0][0]
not_equal (NotEqual)	(None, 34)	0	Sequence_Input[0][0]
batch_normalization_95 (BatchNormalization)	(None, 256)	1,024	dense[0][0]
lstm (LSTM)	(None, 256)	525,312	embedding[0][0], not_equal[0][0]
add (Add)	(None, 256)	0	batch_normalization_95[0]... lstm[0][0]
dense_1 (Dense)	(None, 256)	65,792	add[0][0]
Output_Layer (Dense)	(None, 8584)	2,206,600	dense_1[0][0]

Total params: 5,529,481 (21.09 MB)
Trainable params: 5,524,876 (21.08 MB)
Non-trainable params: 4,608 (18.00 KB)

Figure 8 Building Image Captioning Model

Training the Image Captioning Model

The trained model is completed with the data that is preprocessed for Early Stopping so as not to overfit the data and Learning Rate Scheduler to schedule the rate of learning. Training can be done until the loss functions are not improved in the validation data from a maximum of 15 epochs, as shown in Fig. 9.

```
Epoch 1/15
119/119 ━━━━━━━━━━━━━━━━━━━━ 1069s 9s/step - loss: 5.2136 - val_loss: 3.7396 - learning_rate: 0.0055
Epoch 2/15
119/119 ━━━━━━━━━━━━━━━━━━━━ 1235s 10s/step - loss: 3.3702 - val_loss: 3.3706 - learning_rate: 0.0030
Epoch 3/15
119/119 ━━━━━━━━━━━━━━━━━━━━ 1263s 11s/step - loss: 2.9633 - val_loss: 3.2727 - learning_rate: 0.0017
Epoch 4/15
119/119 ━━━━━━━━━━━━━━━━━━━━ 1540s 13s/step - loss: 2.7492 - val_loss: 3.2505 - learning_rate: 9.0718e-04
Epoch 5/15
119/119 ━━━━━━━━━━━━━━━━━━━━ 1260s 11s/step - loss: 2.6125 - val_loss: 3.2507 - learning_rate: 4.9787e-04
Epoch 6/15
119/119 ━━━━━━━━━━━━━━━━━━━━ 1349s 11s/step - loss: 2.5309 - val_loss: 3.2496 - learning_rate: 2.7324e-04
Epoch 7/15
119/119 ━━━━━━━━━━━━━━━━━━━━ 1252s 11s/step - loss: 2.4810 - val_loss: 3.2508 - learning_rate: 1.4996e-04
Epoch 8/15
119/119 ━━━━━━━━━━━━━━━━━━━━ 1395s 12s/step - loss: 2.4511 - val_loss: 3.2549 - learning_rate: 8.2297e-05
Epoch 9/15
119/119 ━━━━━━━━━━━━━━━━━━━━ 2053s 17s/step - loss: 2.4332 - val_loss: 3.2585 - learning_rate: 4.5166e-05
```

Figure 9 Training the Image Captioning Model

Plotting Training and Validation Loss

The plots of training and validation loss are plotted across the epochs to visualize the learning progress of the model. This visualization aids in determining whether a model is overfitting or underfitting by comparing the loss trends over time, illustrated in Fig.10.

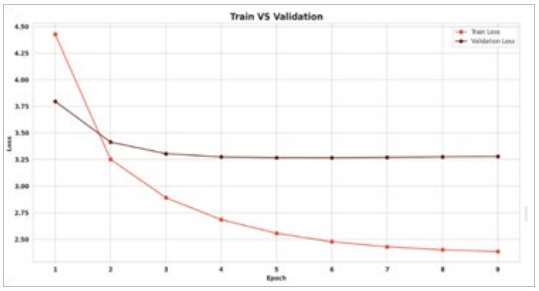
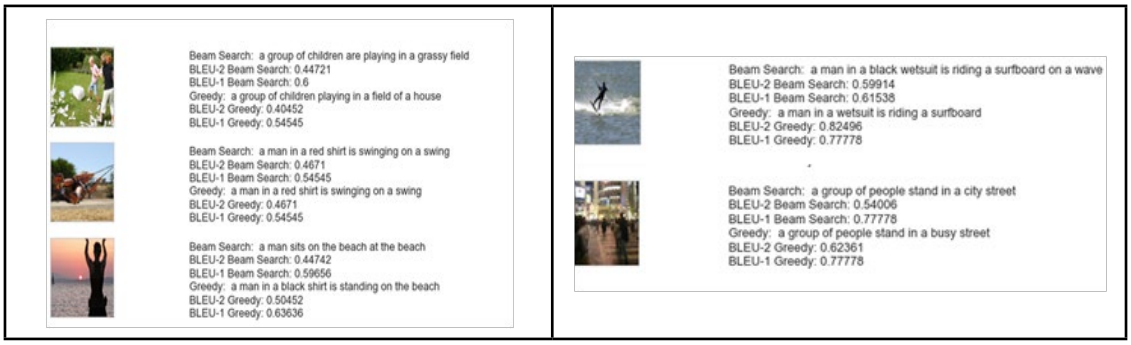


Figure 10 Plotting training and validation loss

Visualizing Generated Captions and Bleu Scores

Random test images are selected, and the original image is displayed alongside captions generated using greedy search and beam search. Their corresponding BLEU scores are shown for evaluation. This makes it easy to evaluate the quality of model-generated captions and the ground truth captions as shown in the Fig. 11.



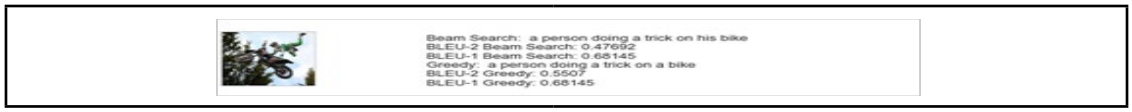


Figure 11 Visualizing Generated Captions and Bleu Score

Conclusion

Automatic image captioning based on Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks is an emerging area that combines the fields of computer vision and natural language processing. This allows the generation of descriptive textual captions from the image based on its content. In this work, the Flickr8k dataset—comprising thousands of images annotated with human-generated captions—was utilized for model training and evaluation. A pre-trained InceptionV3 model is used for high-level visual feature extraction from images. The extracted features are subsequently provided to an LSTM-based decoder, which models the sequential nature of language to generate coherent and semantically meaningful image captions The encoder-decoder architecture effectively captures both visual semantics and language generation patterns. To assess the performance of the model, BLEU (Bilingual Evaluation Understudy) scores were calculated to quantify the similarity between machine-generated and human-written captions. Furthermore, visualization of sample image-caption pairs was conducted to qualitatively analyze the model's outputs. This study demonstrates the capability of deep learning models, particularly CNN-LSTM combinations, in addressing multimodal learning problems. The results affirm the viability of such architectures for real world applications including assistive technologies, content indexing, and enhanced human-computer interaction.

Future Work

Although the current model demonstrates promising performance, several directions can be pursued to enhance its effectiveness and applicability:

- Larger Datasets: Incorporating more extensive and diverse datasets, including multilingual captions, can significantly improve the model's generalization capabilities.
- Attention Mechanisms: The proposed architecture can be enhanced through the integration of attention mechanisms that enable it to selectively focus on salient image regions during caption generation, thereby generating more accurate and contextually rich descriptions.
- Evaluation Metrics: Expanding the set of evaluation metrics beyond BLEU, such as incorporating ROUGE, METEOR, or CIDEr, may offer a more comprehensive assessment of caption quality.
- Real-time Applications: The model can also be used in real-life applications like automatic tagging in social platforms, vision-assistive technology for the visually impaired, and content creation in multimedia systems.

References

P. P. Khaing, "Attention-based deep learning model for image captioning: A comparative study," International Journal of Image, Graphics and Signal Processing, vol. 14, no. 6, pp. 1–10, 2019.

J. Yu, J. Li, Z. Yu, and Q. Huang, "Multimodal transformer with multi-view visual representation for image captioning," IEEE Transactions on Circuits and Systems for Video Technology, vol. 30, no. 12, pp. 4467–4480, 2019.

F. Shaikh, "Solving an image captioning task using deep learning," Analytics Vidhya, Apr. 16, 2018. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/04/solving-an-image-captioning-task-using-deep-learning/

X. Xiao, L. Wang, K. Ding, S. Xiang, and C. Pan, "Deep hierarchical encoder–decoder network for image captioning," IEEE Transactions on Multimedia, vol. 21, no. 11, pp. 2942–2956, 2019.

P. Zeng, H. Zhang, J. Song, and L. Gao, "S2 transformer for image captioning," in Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 2022, pp. 1608–1614.

H. Sharma and S. Srivastava, "Graph neural network-based visual relationship and multilevel attention for image captioning," Journal of Electronic Imaging, vol. 31, no. 5, pp. 053022-1–053022-12, 2022.

S. Cho and H. Oh, "Generalized image captioning for multilingual support," Applied Sciences, vol. 13, no. 4, p. 2446, 2023.

Y. Lin, K. Lai, and W. Chang, "Skin medical image captioning using multi-label classification and siamese network," IEEE Access, vol. 11, pp. 23447–23454, 2023.

A. A. Osman, M. A. W. Shalaby, M. M. Soliman, and K. M. Elsayed, "A survey on attention-based models for image captioning," International Journal of Advanced Computer Science and Applications, vol. 14, no. 2, 2023.

X. Zhang, J. Shen, Y. Wang, J. Xiao, and J. Li, "Zero-shot image caption inference system based on pretrained models," Electronics, vol. 13, no. 19, 2024.