# Image Captioning with Deep Learning: A CNN-RNN Approach

1st HRITIK ROSHAN SHAW
*Centre of Excellence in Artificial Intelligence*
*IIT Kharagpur*
KHARAGPUR, INDIA
hritikroshanshaw.24@kgpian.iitkgp.ac.in

2nd AJAY VAMSI VELAMARTHI
*Centre of Excellence in Artificial Intelligence*
*IIT Kharagpur*
KHARAGPUR, INDIA
av.24@kgpian.iitkgp.ac.in

***Abstract*—This project explores a deep learning-based approach to image captioning aimed at generating natural language descriptions of images. The proposed model integrates Convolutional Neural Networks (CNNs) for visual feature extraction and Recurrent Neural Networks (RNNs) with Long Short-Term Memory (LSTM) units for language modeling. Key features include multilingual translation, text-to-speech (TTS), and Braille encoding, enhancing accessibility for visually impaired users. The model was trained on the Flickr8k dataset and evaluated using BLEU metric and human feedback, demonstrating strong performance on simple images but challenges with complex scenes.**

## I. Introduction

Image captioning is a multidisciplinary challenge at the intersection of computer vision and natural language processing (NLP). It involves generating coherent, meaningful descriptions for visual content, combining the ability to understand image features with the capacity to articulate this understanding in human language. Applications range from aiding visually impaired individuals and enhancing multimedia search engines to enabling human-robot interaction.

The motivation for this project arises from the World Health Organization's estimate that over 285 million people globally suffer from visual impairments. By generating descriptive captions for images, translating them into multiple languages, and converting them into Braille or audio, this project seeks to bridge the gap between visual content and accessibility, empowering visually impaired users to engage with the visual world.

### A. objective:

Generate coherent, contextually accurate captions for images. Extend captions to Braille, multilingual text, and speech formats for accessibility.

### B. scope:

Accurately describe objects, actions, and their relationships in various scenes. Ensure inclusivity through multiple accessibility formats.

## II. Related Work

### A. Notable Existing Works

The problem of generating natural language descriptions from visual data has long been studied in computer vision. Here are some notable works related to this work, Farhadi et al.: Proposed scene triplet-based descriptions (e.g., object-action-scene), limited by its rigid structure and inability to generalize.Kiros et al.: Introduced a neural network model to predict the next word in a sequence, but lacked the flexibility to handle unseen scenarios or complex relationships. Mao et al.: Utilized RNNs to generate descriptions by sequentially predicting words, improving fluency but struggling with intricate scenes involving multiple objects. Show and Tell Model: Introduced by Vinyals et al., this model was one of the first to use CNNs for feature extraction and LSTMs for caption generation. Attention Mechanisms: Xu et al. developed attention-based models to allow the decoder to focus on specific parts of the image. Transformers for Image Captioning: Recent models incorporate transformers for more efficient language generation, providing improved contextual understanding.

### B. Challenges in Previous Methods

Reliance on templates or predefined structures, limiting diversity in captions.

Inadequate handling of complex object interactions and novel compositions.

The proposed model addresses these limitations by leveraging a CNN-RNN architecture for end-to-end training, enabling it to generate flexible, human-like descriptions.

## III. Dataset

When selecting or preparing a dataset for training, it is essential to consider several critical characteristics:

a. Diversity of Captions: Variety in Phrasing: Each image should have captions that are not identical, providing varied language structures and expressions. Rich Vocabulary: Captions should include a broad vocabulary to ensure the model learns to generalize language effectively. Multiple Captions per Image: Helps the model understand different ways an image can be described, leading to better caption generalization. Why: This enables the model to learn the mapping between visual features and complex language constructs, improving

Identify applicable funding agency here. If none, delete this.

its ability to generate diverse and contextually appropriate captions.

b. Image Diversity and Quality: Variety of Contexts: The dataset should include images depicting various scenes, objects, activities, and contexts (e.g., outdoor scenes, indoor activities, animals, vehicles). Resolution and Clarity: High-quality images with clear visuals are better for feature extraction by CNNs. Why: Diverse and high-quality images ensure the CNN can learn robust feature representations, making the model effective across various types of input images.

c. Annotation Quality: Human-Annotated Captions: Captions should be written by humans to maintain natural language use and avoid biases inherent in automated captioning systems. Consistency in Labels: Captions should be consistent in terminology while allowing variability in expression. Why: High-quality annotations enable the LSTM or RNN-based decoder to learn better sentence structures, improving the accuracy of generated captions.

d. Dataset Size and Balance: Large Dataset: Preferably, a dataset with a substantial number of images and captions (e.g., MS COCO) is ideal for training deep learning models to avoid overfitting and achieve generalization. Balanced Representation: The dataset should represent various categories and contexts equally to prevent the model from being biased toward more frequently occurring scenarios.

The Flickr8k, Flickr30k, and MS COCO datasets are well-suited for training image captioning models. Each dataset contains images paired with multiple human-written captions that describe the image's content.But here we are using Flickr8k because of following reasons:

Simpler and Faster Training: Reduced Computational Resources: Flickr8k's smaller size (8,000 images with captions) allows for faster training and requires less memory, making it suitable for environments with limited computational power. Quick Prototyping: Its size facilitates rapid validation of model architecture and hyperparameters before scaling up to larger datasets. Focused Learning:

High-Quality Annotations: Captions are well-annotated and concise, enabling the model to learn fundamental relationships between visual features and language. Consistent Data: The smaller dataset size allows easier verification and ensures consistent, high-quality training data. Reduced Risk of Overfitting (in Prototyping):

Smaller Model Requirements: Encourages the development of simpler models with fewer parameters, reducing overfitting in early stages. Baseline Development: Flickr8k serves as an excellent baseline to evaluate model components like CNN feature extractors and LSTM decoders before scaling. Ease of Experimentation:

Iterative Improvement: The dataset size permits iterative experiments, allowing quick adjustments in architecture and training strategies. Efficient Hyperparameter Tuning: Smaller datasets allow for more efficient tuning of learning rates, batch sizes, dropout, and other hyperparameters. Benchmarking for Comparisons:

Standard Benchmark: Flickr8k is a widely-used benchmark for early image captioning research, facilitating direct comparison with existing work. Transferability: Success on Flickr8k provides a reference point for adapting to larger datasets like MS COCO. Accessibility and Simplicity:

Easy to Obtain and Use: Accessible for researchers and students, making it an ideal starting point. Well-Documented: Established baselines and implementations are available for guidance.

Before training an image captioning model, it is crucial to preprocess both the images and captions to ensure consistency, optimal learning, and better performance. Below are the preprocessing steps typically applied and their purposes:

1. Image Preprocessing: Resizing: All images are resized to a uniform shape (e.g., 224x224 or 299x299 pixels) to match the input size requirements of pre-trained CNN models like VGG16, ResNet, or Inception. Normalization: Image pixel values are normalized (e.g., scaled between 0 and 1 or standardized using mean and standard deviation) to make the training process more stable and improve convergence speed. Why: Resizing ensures that all images fit the model architecture, while normalization helps in faster and more stable gradient descent during training.

2. Caption Preprocessing: Tokenization: Captions are broken down into individual words or tokens to prepare them for numerical representation. Special tokens such as ¡start¿ and ¡end¿ may be added to indicate the beginning and end of sentences. Lowercasing: All captions are converted to lowercase to maintain uniformity and reduce the vocabulary size. Removing Punctuation: Non-essential punctuation is removed to simplify the text and reduce vocabulary complexity. Padding: Captions are padded to a uniform length to fit into fixed-size input vectors for training the LSTM or RNN. Padding ensures batch processing can be performed efficiently. Why: These steps help the model learn from consistent and well-structured input. Tokenization enables word embedding and sequence modeling, while lowercasing and punctuation removal streamline the training by reducing the dimensionality of the vocabulary. Padding ensures that all input sequences have a uniform length for batch processing.

3. Encoding Captions: Word Mapping: Each word in the tokenized captions is mapped to a unique integer index using a vocabulary dictionary. This encoding is necessary for training neural networks. Why: Neural networks cannot process raw text, so integer encoding facilitates embedding layers or direct input to RNNs.

4. Vocabulary Creation: Frequency Filtering: Words that appear below a certain frequency threshold are excluded to reduce the vocabulary size and improve learning efficiency. Why: This step prevents rare words from cluttering the training process and ensures the model focuses on learning more frequent, useful words.

5. Data Augmentation (Optional): Random Transformations: Techniques such as horizontal flips, random crops, and slight rotations can be applied to images to artificially expand the

training dataset. Why: Augmentation helps prevent overfitting by making the model more robust to variations in input images.

Purpose of Preprocessing: Ensures Uniformity: Standardizes image and caption formats for seamless input into the model. Improves Learning: Preprocessing steps like normalization and tokenization enhance the model's learning efficiency and stability. Reduces Complexity: Vocabulary filtering and lowercasing simplify the learning task by reducing unnecessary complexity. Prevents Overfitting: Data augmentation helps the model generalize better by exposing it to slight variations during training. Preprocessing lays the foundation for successful model training, enabling the model to focus on learning meaningful patterns from consistent and well-prepared input data. Proper preprocessing contributes significantly to the model's overall accuracy, efficiency, and performance.

## IV. Methodology

The methodology for developing an image captioning model encompasses several key stages, integrating computer vision and natural language processing techniques to generate descriptive captions for images and extend these captions to Braille and Hindi. Below is a comprehensive outline:

### *A. Model Architecture Overview*

Convolutional Neural Network (CNN) for Feature Extraction: A pre-trained CNN model (e.g., VGG16, ResNet, Inception) is used to extract high-level feature representations from images. Why: Transfer learning enables robust feature extraction without needing to train the CNN from scratch. Recurrent Neural Network (RNN) or Long Short-Term Memory (LSTM) for Caption Generation:

The LSTM network serves as the language model that takes the extracted image features as input and generates descriptive captions word by word. Why: LSTMs can effectively manage long-term dependencies, essential for coherent and contextually appropriate sentences.

The Detail overview is given below:

Step 1: Image Feature Extraction: Images are fed into a pre-trained CNN model, and feature vectors are extracted from a high-level layer, serving as input for the language model.

Step 2: Caption Tokenization and Embedding: Captions in the dataset are tokenized and converted into sequences of integer indices. An embedding layer maps these indices to dense vector representations.

Step 3: Model Integration: Extracted image features are combined with embedded caption sequences. Image features are passed through a fully connected layer and integrated with the LSTM for sequential caption generation.

Step 4: Sequential Caption Generation: The LSTM generates captions word by word during training using teacher forcing, where the actual word at each step is used as input to stabilize training.

Step 5: Training and Optimization: The model is trained with categorical cross-entropy loss and optimized using methods like Adam or RMSprop to ensure convergence.

### *B. Preprocessing for Braille and Hindi Conversion*

Converting Captions to Braille: Once a caption is generated, it is converted to Braille ASCII using a predefined mapping where each character is matched to its corresponding Braille symbol. This step ensures that visually impaired users can access the captions in a readable format. The Braille representation helps make the generated captions accessible through Braille displays or embossed prints.

Implementation: A Python dictionary maps English characters, numbers, and punctuation to their Braille ASCII equivalents.The function text_to_braille_ascii() processes the caption and converts each character

Converting Captions to Hindi: Generated captions are also translated into Hindi using an automated translation tool (e.g., googletrans library). Translating captions into Hindi ensures accessibility for Hindi-speaking users, broadening the usability of the model.

Implementation: The translate_text() function uses the googletrans library to translate English captions to Hindi. The output is a linguistically accurate caption in Hindi, which can be displayed or processed further for other applications.

### *C. Model Evaluation Metrics*

BLEU Score: Evaluates the n-gram precision between generated captions and reference captions. CIDEr, METEOR, and ROUGE: Additional metrics that measure recall and precision, assessing how well the generated captions align with human annotations.

### *D. Training and Validation Strategy*

Data Splitting: A train-validation split ensures that the model's performance is monitored on unseen data during training. Early Stopping and Hyperparameter Tuning: Early stopping helps prevent overfitting, while parameters such as learning rate and LSTM units are fine-tuned for optimal results.

### *E. Model Inference*

Caption Generation: During inference, image features are passed to the LSTM, which predicts the next word step-by-step until an ¡end¿ token or a maximum length is reached. Beam Search: Beam search decoding can be used for generating higher-quality captions by exploring multiple paths and selecting the most probable sequence.

### *F. Post-Processing*

Braille Conversion: The generated English captions are converted to Braille using the predefined ASCII mapping for accessibility. Hindi Translation: The captions are translated into Hindi using a translation library to cater to non-English-speaking users.

Why This Methodology Works: Comprehensive Accessibility: Integrating Braille conversion and Hindi translation ensures that the image captioning system is inclusive, serving visually impaired and Hindi-speaking users.

Effective Model Structure: Combining CNNs for visual feature extraction and LSTMs for sequence modeling bridges the visual and linguistic aspects efficiently. Pre-trained Components: Using pre-trained CNNs accelerates training and provides robust feature extraction, improving overall performance.

## V. Experimental Results

The BLEU (Bilingual Evaluation Understudy) score is the primary quantitative metric used. It measures n-gram overlap between the generated captions and the reference captions (human-written). The BLEU score for our experiment is around 0.48 . Since we trained the model with Flickr8k dataset which have few number of images compared to Flickr30k, and MS COCO datasets that is why our model is not giving that much good prediction .Here is how our loss function in training and validation:

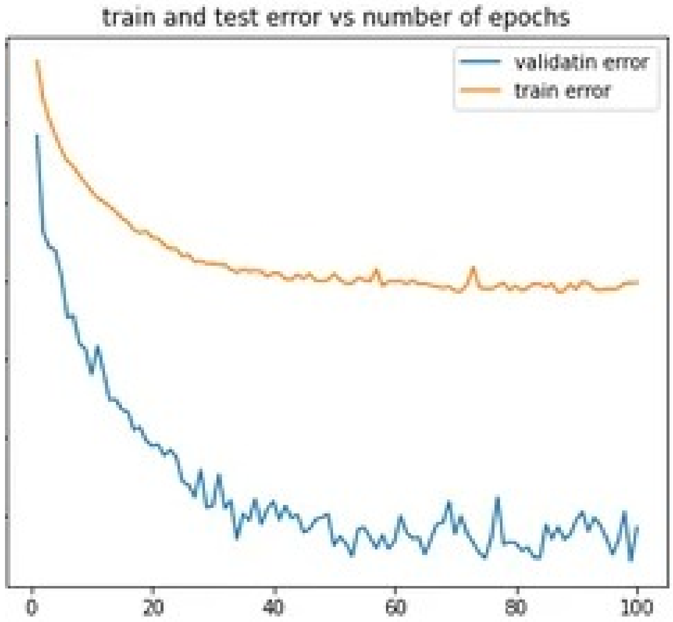
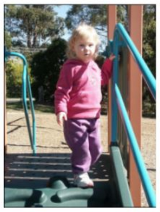


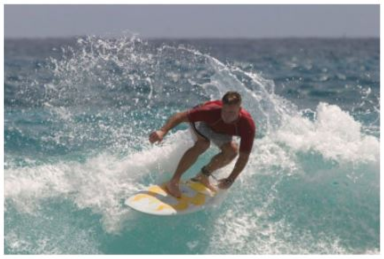
Fig. 1. Loss curve

But still the the model is performing good on training images . Here are few examples:



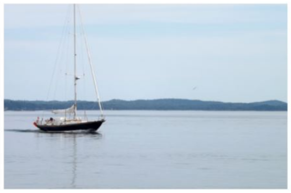
Predicted Caption: <SOS> a little girl in a pink shirt and jeans is jumping on a trampoline . <EOS>
Braille: ⠁ ⠇⠊⠞⠞⠇⠑ ⠛⠊⠗⠇ ⠊⠝ ⠁ ⠏⠊⠝⠅ ⠎⠓⠊⠗⠞ ⠁⠝⠙ ⠚⠑⠁⠝⠎ ⠊⠎ ⠚⠥⠍⠏⠊⠝⠛ ⠕⠝ ⠁ ⠞⠗⠁⠍⠏⠕⠇⠊⠝⠑ ⠲

Fig. 2. Predicting the caption of training image 1



Predicted Caption: <SOS> a man in a wetsuit is surfing in the ocean . <EOS>
Braille: ⠁ ⠍⠁⠝ ⠊⠝ ⠁ ⠺⠑⠞⠎⠥⠊⠞ ⠊⠎ ⠎⠥⠗⠋⠊⠝⠛ ⠊⠝ ⠞⠓⠑ ⠕⠉⠑⠁⠝ ⠲

Fig. 3. Predicting the caption of training image 2



Predicted Caption: <SOS> a man in a red shirt is standing in the sand with his arms outstretched . <EOS>
Braille: ⠁ ⠍⠁⠝ ⠊⠝ ⠁ ⠗⠑⠙ ⠎⠓⠊⠗⠞ ⠊⠎ ⠎⠞⠁⠝⠙⠊⠝⠛ ⠊⠝ ⠞⠓⠑ ⠎⠁⠝⠙ ⠺⠊⠞⠓ ⠓⠊⠎ ⠁⠗⠍⠎ ⠕⠥⠞⠎⠞⠗⠑⠞⠉⠓⠑⠙ ⠲

Fig. 4. Predicting the caption of one of the test image

## VI. Conclusion and Future work

### A. Conclusion

The CNN-RNN architecture demonstrates significant promise in automating image captioning, offering a robust foundation for accessibility solutions. The inclusion of Braille encoding, multilingual translation, and text-to-speech enhances usability for visually impaired individuals globally.

### B. Future Work

Extend the model to process videos and generate continuous captions. Develop interactive captions that respond to specific user queries. Incorporate text recognition from image regions containing written content. Improve caption quality for complex scenes through enhanced training and richer datasets.

## References

1.Farhadi, A., et al. ”Every Picture Tells a Story: Generating Sentences from Images.” ECCV, 2010.

2.Kiros, R., et al. ”Multimodal Neural Language Models.” NIPS Deep Learning Workshop, 2014.

3.Mao, J., et al. ”Deep Captioning with Multimodal Recurrent Neural Networks.” arXiv preprint arXiv:1412.6632, 2014.

4.Lin, T.-Y., et al. ”Microsoft COCO: Common Objects in Context.” ECCV, 2014.

5.Kingma, D. P., Ba, J. ”Adam: A Method for Stochastic Optimization.” arXiv preprint arXiv:1412.6980, 2014.

6.Vinyals, O., Toshev, A., Bengio, S., Erhan, D. (2015). Show and Tell: A Neural Image Caption Generator.

7.Xu, K., Ba, J., Kiros, R., et al. (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention.

8.Various online resources for Braille ASCII mappings.

## VII. Appendix

CRediT authorship contribution statement

**Ajay Vamsi Velmarthi:** Conceptualization, Methodology, Investigation, Validation, Writing – original draft, Visualization.

**Hritik Roshan Shaw:** Conceptualization, Software, Investigation, Validation, Writing – review  editing, Resources, Supervision.