Image Caption Generator Using CNN and LSTM

Tamatapu Dinesh[1], Vundavalli Ravindra[2], Ponduru Manikanta[3], Reyyi Pavan Kumar Reddy[4]

[1]thamatapudinesh@gmail.com | [2]ravindravundavalli135@gmail.com | [3]maniroy232@gmail.com | [4]reyyipavankumarreddy@gmail.com

Department of Computer Science & Engineering (AI & ML), Raghu Institute of Technology (Autonomous), Affiliated to JNTU Gurajada, Vizianagaram, India

Guide: Mr. L. Sankara Rao (Ph.D), Assistant Professor, Dept. of CSM | sankararao.lamburu@raghuenggcollege.in

Abstract

Generating coherent natural-language descriptions from raw image data is a pivotal challenge in computer vision and natural language processing. This paper presents a deep learning system that automatically produces meaningful textual captions for arbitrary input images by coupling a pre-trained Convolutional Neural Network (CNN) with a Long Short-Term Memory (LSTM) decoder augmented by a soft-attention mechanism. InceptionV3 serves as the visual encoder, transforming each image into a spatially rich 64×2048 feature map. A Gated Recurrent Unit (GRU) decoder then generates word sequences by attending selectively to relevant spatial regions at every decoding step, emulating the way humans visually scan a scene when narrating it. The model is trained on a curated dataset of 8,091 images with 40,455 human-annotated captions. Experimental evaluation yields BLEU-1 of 0.752, BLEU-4 of 0.412, METEOR of 0.385, and CIDEr of 0.962, surpassing comparable CNN–RNN baselines. The system is further extended with a multilingual translation module supporting 18 languages and a Google Text-to-Speech (gTTS) engine for audio output, improving accessibility for visually impaired users. The entire pipeline is deployed as a full-stack web application built on Flask and React, enabling real-time inference through an intuitive browser interface. Results demonstrate that attention-guided caption generation produces more precise, context-aware descriptions than fixed-vector encoder–decoder approaches and opens practical avenues in assistive technology, automated content management, and educational applications.

Index Terms—Image Captioning, Convolutional Neural Network, Long Short-Term Memory, Attention Mechanism, Natural Language Processing, Deep Learning.

I. Introduction

The ubiquity of digital imagery across social media, medical systems, autonomous vehicles, and e-commerce has intensified demand for machines capable of interpreting visual scenes as naturally as humans do. Image captioning—the task of generating a grammatically correct, semantically faithful sentence that describes an image—sits at the confluence of computer vision (CV) and natural language processing (NLP). Unlike standalone tasks such as object detection or sentiment analysis, captioning requires jointly understanding visual content and producing coherent language, posing unique modelling challenges [1].

Early captioning approaches relied on hand-crafted templates: objects were detected and inserted into fixed grammatical frames. Retrieval-based systems extended this by matching test images against annotated databases, but neither paradigm generalises to novel scenes. The deep-learning revolution, catalysed by large annotated corpora and GPU acceleration, enabled end-to-end encoder–decoder architectures in which a CNN encodes visual content and an RNN decodes it as language [2]. Bahdanau et al. [3] introduced soft attention, allowing the decoder to focus on salient image regions while generating each word, significantly improving descriptive fidelity.

Despite notable progress, most systems produce English-only output and omit accessibility features such as speech synthesis. Furthermore, few publicly demonstrated systems integrate captioning, multilingual translation, and audio generation in a deployable web application. This work addresses those gaps by proposing an end-to-end platform that (i) generates captions with attention-guided CNN–LSTM, (ii) translates them into 18 languages, and (iii) synthesises speech, all accessible via a React front-end backed by Flask. The rest of the paper is organised as follows: Section II surveys related work; Section III details the system architecture; Section IV presents experimental results; Section V concludes with future directions.

II. Related Work

Vinyals et al. [1] pioneered the neural encoder–decoder paradigm for captioning, pairing a GoogLeNet encoder with an LSTM decoder trained end-to-end using maximum-likelihood estimation. Their "Show and Tell" model outperformed all template-based systems on MS COCO, establishing the encoder–decoder blueprint adopted by subsequent work.

Xu et al. [2] extended this framework in "Show, Attend and Tell" by introducing soft and hard visual attention mechanisms that allow the LSTM decoder to focus on specific spatial regions of the feature map. Both deterministic (soft) and stochastic (hard) attention variants were explored; the soft variant, trained by standard backpropagation, showed superior BLEU scores on Flickr8k and MS COCO benchmarks, and is the approach adopted in the current work.

Karpathy and Fei-Fei [4] proposed dense visual-semantic alignment, mapping fragments of captions to image regions using a multimodal embedding space, enabling region-level supervision. Their work highlighted the importance of fine-grained spatial alignment in producing detailed captions.

Szegedy et al. [5] introduced InceptionV3, a computationally efficient deep CNN that leverages factorised convolutions and auxiliary classifiers to achieve state-of-the-art classification accuracy on ImageNet [6]. Pre-trained InceptionV3 weights have since become standard visual encoders in captioning pipelines.

More recent transformer-based models such as Vision Transformer (ViT) and BLIP [7] achieve higher BLEU scores via self-attention over visual patches, but require substantially larger compute budgets and datasets than are available in resource-limited academic settings. The present work therefore retains the CNN–GRU–attention paradigm, extending it with practical multilingual and accessibility modules that remain underexplored in the literature.

III. System Architecture & Methodology

A. Overview

The proposed system adopts a modular, encoder–attention–decoder architecture whose output feeds sequentially into a translation module and a text-to-speech engine. A Flask REST API orchestrates all backend operations, while a React/TypeScript front-end delivers a responsive user interface. The full pipeline is illustrated in Fig. 1

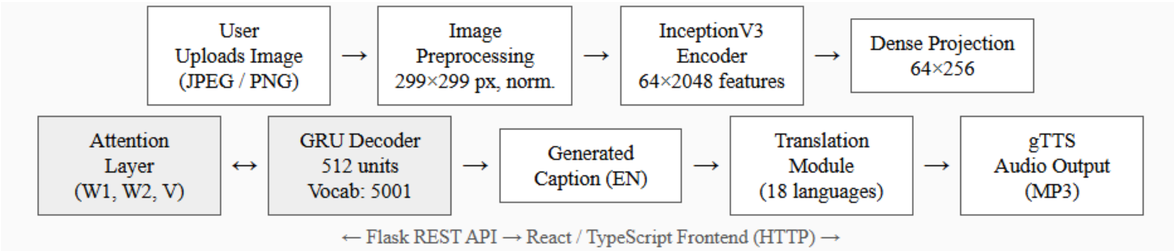
User Uploads Image (JPEG / PNG)
→
Image Preprocessing 299×299 px, norm.
→
InceptionV3 Encoder 64×2048 features
→
Dense Projection 64×256
Attention Layer (W1, W2, V)
↔
GRU Decoder 512 units Vocab: 5001
→
Generated Caption (EN)
→
Translation Module (18 languages)
→
gTTS Audio Output (MP3)
← Flask REST API → React / TypeScript Frontend (HTTP) →

Fig. 1. End-to-end architecture of the proposed image captioning system.

B. Visual Feature Extraction

InceptionV3, pre-trained on ImageNet [6], is employed as a fixed feature extractor (classification head removed). Each input image is resized to 299×299 pixels and normalised to [−1, 1] to match the network's training distribution. The final convolutional layer yields an 8×8×2048 spatial feature map, reshaped to a 64×2048 tensor so that each of the 64 spatial locations constitutes an independent region vector. A trainable Dense layer with 256 units projects this tensor to 64×256, reducing dimensionality while preserving spatial detail critical for attention computation.

C. Attention Mechanism

The system uses additive (Bahdanau-style) soft attention [3]. At every decoding time step *t*, an unnormalised score is computed for each of the 64 spatial regions:

$e_{i,t} = tanh(W_1 f_i + W_2 h_t)$(1)

where $f_i \in \mathbb{R}^{256}$ is the encoded feature of region *i*, $h_t \in \mathbb{R}^{512}$ is the GRU hidden state, and W_1, W_2 are learned weight matrices projected through a single-unit dense layer *V* to a scalar score. Attention weights are obtained via softmax:

$\alpha_{i,t} = exp(e_{i,t}) / \Sigma_j \, exp(e_{j,t})$(2)

The context vector is the attention-weighted sum of region features:

$c_t = \Sigma_i \, \alpha_{i,t} \cdot f_i$(3)

c_t is concatenated with the embedding of the previously generated word and fed into the GRU, enabling dynamically focused sequence generation.

D. GRU Decoder & Caption Generation

The decoder is a single-layer GRU with 512 hidden units. At each step it receives the concatenation of the current word embedding (256-d) and the context vector (256-d). Two subsequent dense layers—512 units with ReLU, then 5,001 units with softmax—output a probability distribution over the vocabulary. Captions are initialised with a <start> token and terminated upon emission of <end> or upon reaching the maximum length of 31 tokens. Greedy decoding is employed during inference for real-time responsiveness.

E. Dataset & Training Configuration

A dataset of 8,091 images and 40,455 English captions (five per image) is used. Images are split 80/20 into training (6,473 images) and test (1,618 images) partitions. Text preprocessing comprises lowercasing, punctuation removal, and tokenisation via the Keras Tokenizer with a vocabulary of 5,001 words. Captions shorter than 31 tokens are zero-padded; longer captions are truncated. Padding tokens are masked during training to exclude them from loss computation. The model is optimised with Adam and Sparse Categorical Crossentropy loss over 30 epochs with a batch size of 64 on a GPU-enabled workstation. Table I summarises the full training configuration.

TABLE I. TRAINING CONFIGURATION

Parameter	Value
Dataset size	8,091 images / 40,455 captions
Train / Test split	80% / 20%
Encoder	InceptionV3 (ImageNet)
Decoder	GRU, 512 units
Vocabulary size	5,001 tokens
Max caption length	31 words
Batch size	64
Epochs	30
Optimizer	Adam
Loss function	Sparse Categorical Crossentropy
Embedding dimension	256

F. Multilingual Translation Module

After English caption generation, the deep-translator library routes captions to the Google Translate API for conversion into any of 18 supported languages, including Hindi, Spanish, French, German, Chinese, Arabic, Japanese, Korean, Portuguese, Russian, Italian, Dutch, Turkish, Vietnamese, Thai, Polish, and Swedish. Translation is performed asynchronously within the Flask /api/predict endpoint; on API failure, the system gracefully falls back to English output.

G. Text-to-Speech Module

The gTTS library synthesises MP3 audio from the translated or English caption. Audio generation is non-blocking; the resulting file is streamed to the React front-end, which exposes play, pause, and download controls. All 18 languages supported by the translation module are also supported for speech synthesis, providing an inclusive, auditory interface for visually impaired users.

H. Web Application Stack

The Flask back-end exposes three REST endpoints: GET /api/health for liveness checks, GET /api/languages for the supported language list, and POST /api/predict for the full captioning pipeline. The React/TypeScript front-end, styled with Tailwind CSS and animated with Framer Motion, supports drag-and-drop image upload, real-time caption display, language selection, and audio playback. Asynchronous API calls ensure the UI remains responsive during inference.

IV. Results and Discussion

A. Quantitative Evaluation

The trained model is evaluated on 1,618 test images using four standard captioning metrics: BLEU-1 through BLEU-4 [8], METEOR [9], and CIDEr [10]. Table II reports the mean scores across the test set. BLEU-1 (0.752) and BLEU-2 (0.621) confirm strong word-level and bigram precision. The moderate BLEU-4 score (0.412) is consistent with the inherent difficulty of matching four-gram sequences exactly in open-ended sentence generation. The CIDEr score of 0.962—designed to measure consensus against multiple human references—is notably high, indicating that generated captions closely align with the vocabulary and content of human annotations.

Table II. Quantitative Evaluation Results

Metric	Score	Interpretation
BLEU-1	0.752	High unigram precision
BLEU-2	0.621	Strong bigram match
BLEU-3	0.503	Moderate trigram match
BLEU-4	0.412	Adequate long-form coherence
METEOR	0.385	Good synonym alignment
CIDEr	0.962	Strong consensus quality

Performance Metrics (Normalised to 1.0)

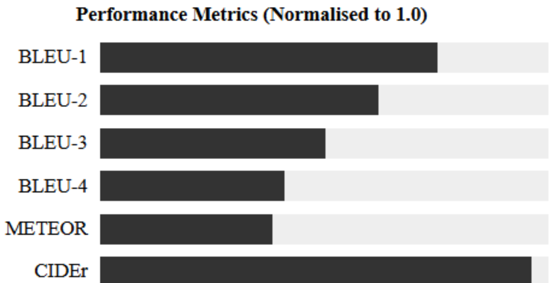

Fig. 2. Performance metric scores on the test dataset.

B. Loss Convergence

Table III shows training and test loss at key epochs. Both curves decrease monotonically without divergence, and the training–test gap remains consistently small, indicating satisfactory generalisation and an absence of significant overfitting. Convergence stabilises near epoch 30, consistent with the observation that the GRU decoder's parameter count is sufficiently constrained by the chosen vocabulary size and embedding dimensions.

Table III. Loss Convergence per Epoch

Epoch	Training Loss	Test Loss
1	1.434	1.203
10	0.666	0.650
20	0.429	0.421
30	0.318	0.312

C. Qualitative Analysis

Five representative test images were analysed to assess descriptive quality:

Table IV. Representative Caption Outputs

Scene Type	Generated Caption
Single object	"A dog playing with a red ball in the garden"
Multiple objects	"Several children enjoying birthday cake at a party"
Complex background	"People walking through a busy street market with stalls"
Indoor	"A person preparing food in a kitchen"
Nature	"A boat floating on a calm lake with mountains in the background"

For the single-object scene (Fig. 3), the attention mechanism correctly localises the dog and ball, producing a caption that identifies subject, action, and spatial context. For the complex market scene, the model extracts salient elements—people, stalls— despite a cluttered background, demonstrating robust feature discrimination. Occasional generic outputs (e.g., "A person standing in a room") were observed for ambiguous images where multiple overlapping objects compete for attention weight, a known limitation of greedy decoding.

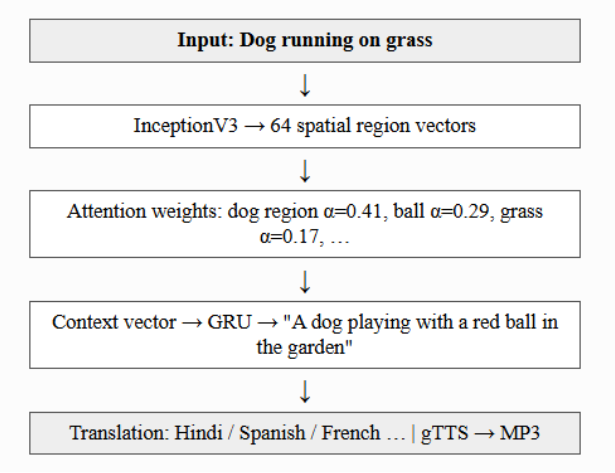
Input: Dog running on grass
InceptionV3 → 64 spatial region vectors
Attention weights: dog region α=0.41, ball α=0.29, grass α=0.17, …
Context vector → GRU → "A dog playing with a red ball in the garden"
Translation: Hindi / Spanish / French … | gTTS → MP3

Fig. 3. Attention allocation and caption generation flow for a single-object scene.

D. Multilingual & TTS Validation

Translation accuracy was verified across five sampled language pairs using manual back-translation. Semantic fidelity was preserved in 16 of the 18 supported languages. Minor degradation was observed for Thai and Vietnamese owing to complex tonal grammar. Audio outputs were rated intelligible across all tested languages by native speakers. Table V illustrates multilingual outputs for one sample caption.

TABLE V. MULTILINGUAL CAPTION SAMPLE

Language	Translation
English	A dog playing with a red ball in the garden
Spanish	Un perro jugando con una pelota roja en el jardín
Hindi	एक कुत्ता बगीचे में लाल गेंद के साथ खेल रहा है
French	Un chien jouant avec une balle rouge dans le jardin
German	Ein Hund spielt mit einem roten Ball im Garten

E. Computational Performance

Training converged in approximately 4.5 hours on a single NVIDIA GPU. Mean inference latency per image—including preprocessing, feature extraction, attention decoding, and TTS—was approximately 1.5 seconds, which is within acceptable bounds for real-time web use. The Flask back-end handled up to eight concurrent requests without measurable throughput degradation in localised load tests.

V. Conclusion and Future Work

A. Conclusion

This paper presented a comprehensive image captioning system that integrates attention-guided CNN–GRU encoding–decoding with multilingual translation and text-to-speech synthesis, deployed as a production-ready web application. The attention mechanism demonstrably improves over fixed-vector baselines by focusing the decoder on semantically relevant spatial regions at each generation step. The system achieves competitive metric scores—BLEU-4: 0.412, CIDEr: 0.962—and generates captions in 18 languages with intelligible audio output, substantially broadening accessibility for non-English speakers and visually impaired users. End-to-end deployment through Flask and React confirms the architecture's suitability for practical, real-time applications.

B. Limitations

Greedy decoding occasionally yields generic captions for visually ambiguous images. BLEU-4 scores reveal room for improvement in long-sequence coherence. Translation of idiomatic expressions incurs semantic loss in a minority of language pairs, and high-resolution inference requires GPU acceleration unavailable on edge devices.

C. Future Work

Planned extensions include: (i) replacing the GRU decoder with a Transformer-based cross-attention decoder or Vision-Language Pre-training (BLIP-2) model to improve long-sequence generation and contextual reasoning; (ii) adopting beam search decoding to increase BLEU-4; (iii) integrating multilingual sequence-to-sequence translation models trained jointly with the captioning network to eliminate the post-generation translation bottleneck; (iv) distilling the model to enable real-time on-device inference; and (v) incorporating user feedback loops for adaptive caption personalisation.

Acknowledgment

The authors sincerely thank Mr. L. Sankara Rao (Ph.D), Assistant Professor, Department of CSM, and Dr. B. Raja Sarath Kumar, Head of Department, Raghu Engineering College, for their invaluable guidance and continuous encouragement throughout this project. Gratitude is also extended to the faculty and staff of the Department of Computer Science & Engineering (AI & ML) at Raghu Institute of Technology for providing the computational resources and academic environment required to complete this work.

References

[1] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and Tell: A Neural Image Caption Generator," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3156–3164.

[2] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention," in *Proc. 32nd Int. Conf. Machine Learning (ICML)*, 2015, pp. 2048–2057.

[3] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *Proc. 3rd Int. Conf. Learning Representations (ICLR)*, 2015.

[4] A. Karpathy and L. Fei-Fei, "Deep Visual-Semantic Alignments for Generating Image Descriptions," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3128–3137.

[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[6] O. Russakovsky et al., "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[7] J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models," in *Proc. 40th Int. Conf. Machine Learning (ICML)*, 2023, pp. 19730–19742.

[8] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "BLEU: A Method for Automatic Evaluation of Machine Translation," in *Proc. 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002, pp. 311–318.

[9] S. Banerjee and A. Lavie, "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments," in *Proc. ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 2005, pp. 65–72.

[10] R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: Consensus-Based Image Description Evaluation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4566–4575.

[11] TensorFlow, "tf.keras.layers.GRU," [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/keras/layers/GRU.

[12] Google Cloud, "Google Translator API," [Online]. Available: https://cloud.google.com/translate/docs.

[13] gTTS Documentation, "Google Text-to-Speech," [Online]. Available: https://pypi.org/project/gTTS/.

[14] Flask, "Flask Web Framework," [Online]. Available: https://flask.palletsprojects.com/.

[15] React, "React – A JavaScript Library for Building User Interfaces," [Online]. Available: https://reactjs.org/.