IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151092

AI-Powered Image Captioning and Content Generation: Design, Implementation, and Evaluation

Hirek Praful Warhekar[1], Kunal Dinesh Afare[2], Manasvi Dharmendra Ingale[3], Sagar Sanjay Kinada[4], Anup G. Kadu[5]

B.Tech Student, Department of Artificial Intelligence & Data Science,

Prof. Ram Meghe Institute of Technology & Research, Badnera-Amravati, India[1]

B.Tech Student, Department of Artificial Intelligence & Data Science,

Prof. Ram Meghe Institute of Technology & Research, Badnera-Amravati, India[2]

B.Tech Student, Department of Artificial Intelligence & Data Science,

Prof. Ram Meghe Institute of Technology & Research, Badnera-Amravati, India[3]

B.Tech Student, Department of Artificial Intelligence & Data Science,

Prof. Ram Meghe Institute of Technology & Research, Badnera-Amravati, India[4]

Project Guide, Department of Artificial Intelligence & Data Science,

Prof. Ram Meghe Institute of Technology & Research, Badnera-Amravati, India[5]

Abstract: Image captioning models describe what a photograph shows, but people who publish images online usually need that description reshaped for a particular language, platform, and tone. This paper presents the design, implementation, and evaluation of a web application that bridges this gap. A BLIP captioning model fine-tuned on Flickr8k produces a base caption; a large language model served through the Groq API rewrites it for a selected platform, language (English, Hindi, or Marathi), and writing style; and a Pillow-based renderer places the result on the image. A Streamlit interface, a FastAPI backend, and a SQLite history store complete the system. On 1,000 Flickr8k test images the captioner obtained BLEU 0.2999, METEOR 0.5697, and CIDEr 0.5191. Ten-run timing tests gave a mean of 7.35 s for caption generation and 23.0 s end to end. Limitations of the evidence are discussed.

Keywords: Image captioning; BLIP; vision-language models; large language models; multilingual content generation; Flickr8k

I. INTRODUCTION

Images dominate what people publish online, yet an image rarely travels alone. A photograph posted to a professional network needs a measured description, the same photograph on a photo-sharing site calls for a lighter tone, and a reader who relies on a screen reader needs a literal account of what is visible. Automatic image captioning, which joins computer vision with natural language generation, supplies the first step of this chain by describing an image in a sentence [1], [2]. The field has moved from convolutional–recurrent encoder–decoders [1], through attention over image regions [3], [4], to transformer-based vision–language models pre-trained on large image–text collections [5], [6], [7].

A single descriptive sentence, however, is rarely the artefact a content creator needs. Turning it into a publishable caption means rewriting it in the audience's language, following the conventions of a platform, choosing a register such as formal, humorous, or promotional, and finally placing the text on the picture. Captioning research mostly evaluates the first step against reference sentences, while work on multilingual [8], [9], style-controlled [10], [11], and instruction-following [12], [13] generation has developed the remaining steps along separate lines. In the literature reviewed for this study we did not find a documented, openly described pipeline that joins these steps in one application, although we do not claim that none exists.

 This work is licensed under a Creative Commons Attribution 4.0 International License

DOI: 10.17148/IJARCCE.2026.151092

This paper reports the design, implementation, and evaluation of such a pipeline. A BLIP captioner [6], fine-tuned on Flickr8k [14], produces a base caption. The base caption is passed, together with the user's choice of platform, language, and writing style, to a large language model reached through the Groq API, which writes the final caption. A Pillow-based renderer then places that caption on the original image using one of six layouts. A Streamlit front end, a FastAPI back end, and a SQLite history store form the application around these models. The design keeps the two generative steps apart on purpose: the vision model is the only source of statements about the image, and the language model is told to change wording and tone but not content. The contributions are:

(1) an end-to-end architecture, described component by component from the implemented code, that turns an uploaded image into a platform-, language-, and style-specific caption and a captioned image;
(2) a fine-tuning and evaluation record for BLIP on a 6,000/1,000/1,000 Flickr8k split, with BLEU, METEOR, and CIDEr measured on the 1,000 test images;
(3) a grounding-oriented prompting strategy for the language-model step, with qualitative examples in English, Hindi, and Marathi;
(4) timing measurements for the generation API, the rendering API, and the complete interface, together with an account of what the evidence does and does not show.

Section II reviews related work, Section III describes the system and the experimental procedure, Section IV presents the results and discusses limitations, and Section V concludes.

II. LITERATURE REVIEW

A. Encoder–Decoder and Attention-Based Captioning

Show and Tell paired a convolutional image encoder with a recurrent language decoder trained to maximise the likelihood of the reference caption, and so set the encoder–decoder pattern that later work followed [1]. Show, Attend and Tell let the decoder weigh different image regions as each word is produced [3], and bottom-up and top-down attention replaced the uniform grid with object-level region features [4]. Hossain et al. organise the deep-learning approaches of this period by architecture and training strategy [2].

B. Transformer-Based Vision–Language Models

The transformer [15] and its image counterpart [16] moved captioning towards models pre-trained on large image–text collections. CLIP learned aligned image and text representations from natural-language supervision [5]. BLIP combines understanding and generation in one pre-training framework and bootstraps cleaner training text by generating synthetic captions and filtering noisy ones [6]; its text side uses a BERT-style encoder [17]. BLIP-2 connects a frozen image encoder to a frozen large language model through a lightweight querying module [7], and InstructBLIP and LLaVA add instruction tuning so that one model can follow different textual requests about an image [12], [13]. Pre-trained checkpoints of this family are distributed through the Hugging Face Transformers library [18], which the present system uses.

C. Large Language Models and Instruction Following

Training language models on human instructions and feedback improves their ability to follow requests such as writing in a given tone [19]. Open-weight models in this class, including the 20-billion-parameter mixture-of-experts model gpt-oss-20b [20], make it practical to use a language model as a text-adaptation stage after a captioner, which is the role the Groq-hosted model plays in this work.

D. Multilingual and Style-Controlled Captioning

PaLI trained a jointly scaled language–image model that covers many languages [8]. mBLIP adapts the BLIP-2 recipe to multilingual use by re-aligning an image encoder with a multilingual language model [9], and ICU divides the task into English image captioning followed by language understanding in the target language [21]. A recent survey compares attention-based transformer captioners across languages [22]. Crossmodal-3600 offers human-written captions in 36 languages and includes Hindi but not Marathi [23]; work on Hindi captioning has relied on encoder–decoder models trained on translated MS COCO data [24]. For style, StyleNet generated humorous and romantic captions by exploiting unpaired stylistic text [10], and Shuster et al. conditioned captions on a personality trait [11].

E. Evaluation of Captions

Captioning systems are usually scored against human references with BLEU [25], METEOR [26], and CIDEr [27]. All three reward overlap with the reference wording, so a correct caption phrased differently can score poorly. CLIPScore

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151092

was proposed as a reference-free alternative [28]. A separate concern is object hallucination, where a caption mentions things that are not in the image, which standard metrics do not measure directly [29].

F. Research Gap

The studies above advance individual capabilities: caption accuracy, language coverage, style control, and instruction following. In the sources we examined, reports of how these capabilities are assembled into a usable workflow, with input validation, stored history, and a shareable image as output, were scarcer, and the hand-over from a captioner to a language model deserves explicit attention given the hallucination problem [29]. This paper contributes a documented implementation of such a workflow together with an account of its measured behaviour. It does not propose a new model, and it makes no claim of superiority over other systems.

III. METHODS AND MATERIALS

A. System Architecture

The application follows a client–server design (Fig. 1). A Streamlit client handles uploads, option selection, and display, and talks over HTTP to a FastAPI server that owns all models and storage. At start-up the server initialises the SQLite repository and loads the fine-tuned BLIP model once; the Groq client is created when the first generation request arrives. Table I lists the components and their responsibilities.

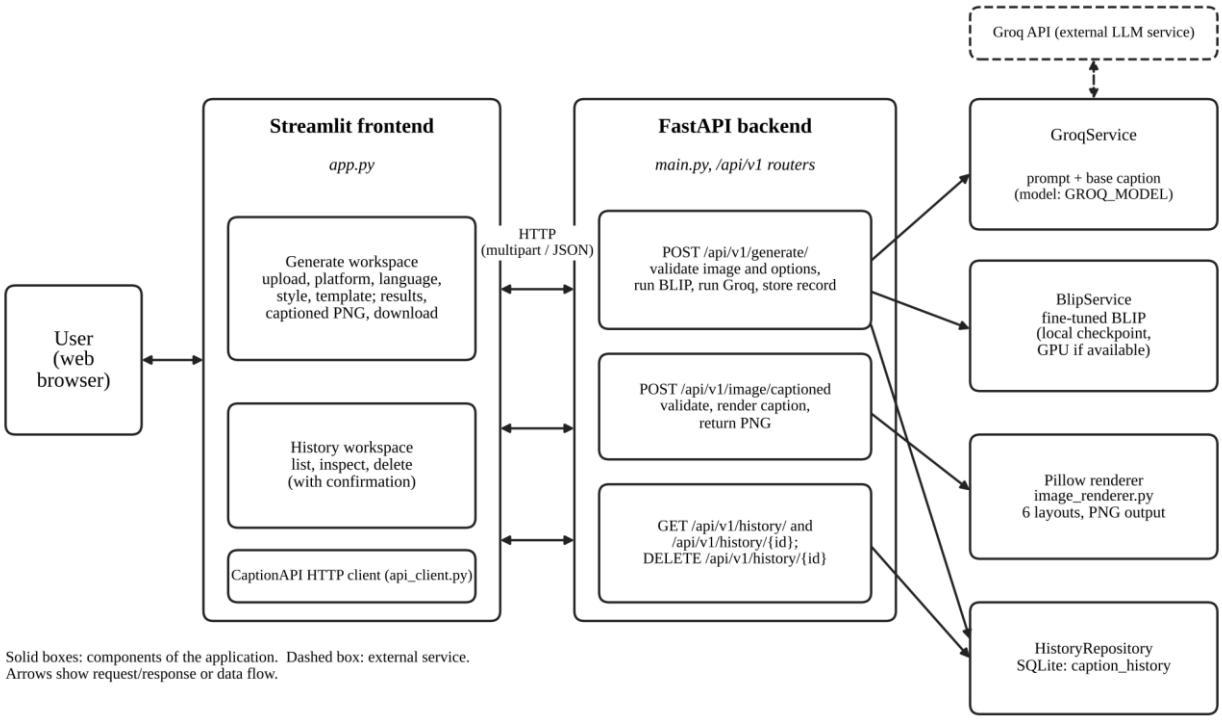


Fig. 1. System architecture of the application. Solid boxes are components of the system; the dashed box is an external service.

TABLE I. SYSTEM COMPONENTS AND RESPONSIBILITIES

Component	Technology	Responsibility
Frontend	Streamlit; requests-based client	Image upload and preview; selection of platform, language, style, and template; display of both captions, the captioned image, and its download; history browsing, inspection, and deletion.
API layer	FastAPI	Input validation and routing: POST /api/v1/generate/, POST /api/v1/image/captioned, GET and DELETE /api/v1/history/…, and GET /health.
Captioner	Fine-tuned BLIP (PyTorch, Transformers)	Base caption for an RGB image using beam search with five beams.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151092

Component	Technology	Responsibility
Content generator	Groq API; default model openai/gpt-oss-20b	Rewrites the base caption for the chosen platform, language, and writing style.
Renderer	Pillow	Draws the caption onto the original image with one of six layouts and returns a PNG.
History store	SQLite	Stores caption records (text and selected options; images are not stored).

B. Request Workflow

Fig. 2 shows the sequence for one generation. The user uploads an image and chooses a platform, a language, a writing style, and a visual template. The frontend sends the image and the four options in a single multipart request to the generation endpoint. The backend validates the request, produces the base caption with BLIP, passes it to the Groq model together with the options, and stores the completed record in SQLite. It returns both captions to the frontend as JSON. The frontend keeps the captions in session state and only then makes a second request to the rendering endpoint, sending the image again together with the generated caption, the language, and the template; the endpoint answers with a PNG. Because the two requests are independent, a rendering failure leaves the captions visible and the history record intact. The history record is written after both captions exist and before rendering takes place.

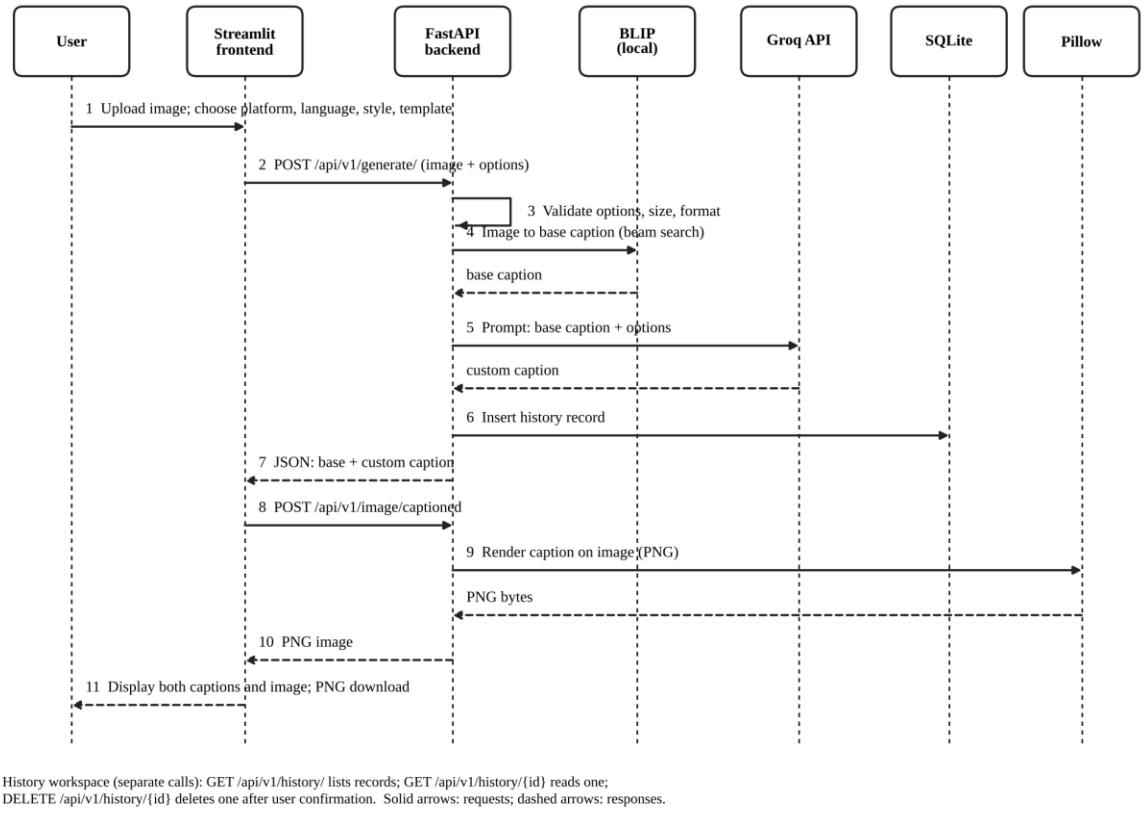


Fig. 2. Sequence of one caption generation. Steps 1–7 form the generation request, steps 8–10 the rendering request, and step 11 the display of results.

C. Frontend

The Streamlit client has two workspaces, Generate and History, selected from a sidebar. The generation page accepts JPG, JPEG, PNG, and WebP files of up to 10 MB, previews the image, and offers four selectors: platform (Instagram, LinkedIn, Facebook, X), language (English, Hindi, Marathi), writing style (Normal, Formal, Creative, Humorous, Marketing, Travel, and Accessibility/Alt Text), and one of six visual templates. Results appear in two panels labelled with the model that produced them, followed by the captioned image and a download button. The HTTP client uses a 10 s connection timeout and a 120 s read timeout, and converts connection failures, timeouts, and server error details into readable messages. The history page lists records newest first in pages of 10, 20, or 50, shows the stored fields, loads a single record on request, and deletes a record after an explicit confirmation step.

DOI: 10.17148/IJARCCE.2026.151092

D. Backend and Input Validation

The generation endpoint checks that the platform, language, style, and optional template belong to the supported sets and rejects other values with status 422. It reads at most 10 MB plus one byte of the upload and answers 413 if the limit is exceeded. Pillow then opens the bytes, requires the detected format to be JPEG, PNG, or WebP (415 otherwise), verifies the file, and reopens it as an RGB image; an unreadable file gives status 400. Failures of the later stages are mapped to explicit statuses: 503 if the Groq key is not configured or the history database is unavailable, and 502 if the Groq provider returns an error. The rendering endpoint applies the same size and format checks and additionally validates the caption (1–2,000 characters), the language, the template, and the canvas preset. Uploaded images are processed in memory; the reviewed generation and rendering code does not write them to disk.

E. Base Caption Generation

A single service object loads the BLIP processor and model from a local checkpoint directory (configurable, defaulting to models/blip-flickr8k), moves the model to a GPU when one is available and otherwise uses the CPU, and switches it to evaluation mode. For each request the processor converts the image into tensors and the model generates a caption by beam search with five beams and at most 50 new tokens, without gradient computation. The decoded text, stripped of special tokens, is the base caption.

F. Dataset and Fine-Tuning

Fine-tuning used the Flickr8k collection [14] as distributed in the Hugging Face repository intro/flickr8k. Each record holds an image, a split label, and five human-written captions. The repository provides 6,000 training, 1,000 validation (dev), and 1,000 test images (Table II), which together form the 8,000 images used here. The dataset build metadata lists 92 further records excluded because they carry no split label, and no missing image files.

TABLE II. FLICKR8K SPLIT USED IN THE EXPERIMENTS

Subset	Images	Share	Use in this work
Training	6,000	75.0%	Fine-tuning
Validation (dev)	1,000	12.5%	Defined, but not used during training
Test	1,000	12.5%	Final quantitative evaluation
Total included	8,000	100%	

Images were converted to RGB, and every (image, caption) pair was treated as a separate training example. The BLIP processor resizes images to 384×384 pixels and tokenises captions to a fixed length of 64 tokens; the caption tokens also serve as labels, with padding positions ignored in the loss. The starting point was the pretrained checkpoint Salesforce/blip-image-captioning-base, loaded as a conditional-generation model; no layers were frozen in the training notebook. Training used the Hugging Face Trainer on a streamed dataset with the settings in Table III, on a Tesla T4 GPU in Google Colab. The run took 2 h 53 min. With 6,000 images and up to five captions each there are up to 30,000 training pairs; at an effective batch of eight, 7,500 optimiser steps correspond to two passes over them. The logged training loss fell from 11.28 at step 100 to 5.97 at step 7,500. We did not verify how the Trainer normalised these values under gradient accumulation and therefore use them only as evidence that optimisation progressed, not as absolute quantities.

TABLE III. FINE-TUNING CONFIGURATION

Setting	Value
Base model	Salesforce/blip-image-captioning-base
Hardware / software	Google Colab, NVIDIA Tesla T4; PyTorch 2.11.0 (CUDA 13.0); Hugging Face Trainer
Batch size	2 per device, 4 accumulation steps (effective batch 8)
Optimiser, learning rate	AdamW (adamw_torch), 5×10^{-5}, 300 warm-up steps
Training length	7,500 optimiser steps (about two passes over the pairs)
Precision and memory	FP16 mixed precision; gradient checkpointing enabled
Caption length	64 tokens (padding and truncation)
Checkpointing	Every 1,000 steps, keeping two; final model saved after step 7,500

The notebook builds a validation dataset from the dev split but does not pass it to the Trainer as an evaluation set. No validation loss or metric was therefore computed during training, no checkpoint was selected on validation data, and the

 This work is licensed under a Creative Commons Attribution 4.0 International License

DOI: 10.17148/IJARCCE.2026.151092

model evaluated below is the one saved at the last step. The unmodified pretrained checkpoint was not evaluated, so the benefit of fine-tuning cannot be quantified from the present evidence.

G. Caption Adaptation with the Groq API

The adaptation stage sends two messages to a chat-completion model on the Groq platform. The model name is read from configuration and defaults to openai/gpt-oss-20b [20]. The system message casts the model as a multilingual social-media copywriter rather than a translator, so that Hindi and Marathi output is written natively instead of translated word for word. It states that the BLIP caption is the only source of facts about the image and forbids adding objects, people, activities, places, weather, time of day, emotions, or relationships that the caption does not support; it asks for hashtags that do not imply unsupported details, for a short output when the base caption is short, and, for the Accessibility/Alt Text style, for a concise literal description. It also requires the reply to contain only the caption. The user message supplies the base caption, the platform, the output language, and the style, and asks for one caption with suitable emojis and, on Instagram, at most three hashtags. Decoding uses a temperature of 0.25 and a limit of 800 completion tokens. The language model receives only text, never the image, so the grounding depends entirely on the base caption and on the model's compliance with the instructions.

H. Caption Rendering

The renderer composes the final image in memory with Pillow and returns a PNG at the original pixel dimensions, after applying the image's EXIF orientation. Six templates are available: a gradient overlay, a minimal footer, and layouts styled for Instagram (dark panel with an accent bar), LinkedIn (light inset panel with a vertical rule), Facebook (cream inset panel with a marker), and X (dark panel with a left bar). The font size is searched from a maximum, bounded by the image size, down to a readable minimum, and the first size at which the wrapped caption fits within the image is used; if none fits, an error is returned instead of clipping the text. Line breaks are computed from rendered pixel widths, and emoji are drawn with a separate colour emoji font. Hindi and Marathi text requires Pillow's complex-text shaping support and a Devanagari-capable font, and the renderer reports an error when either is missing.

I. History Storage

Each completed generation is stored in a SQLite table caption_history with an integer key, the base and generated captions, the platform, language, writing style, an optional visual template, and a UTC timestamp, plus an index on the timestamp. Every repository operation opens its own connection and commits or rolls back as a unit. The history endpoints list records newest first with a limit of 1–100 and a zero-based offset, return one record or 404, and delete one record with status 204 or 404. The history is shared by the application instance and is not tied to user accounts.

J. Evaluation Procedure

Captioning accuracy. The fine-tuned model captioned all 1,000 test images with beam search (five beams, at most 40 new tokens); each image has five human reference captions. Predictions and references were saved, and BLEU was computed at corpus level with the Hugging Face evaluate library, which reports four n-gram precisions and a brevity penalty. METEOR and CIDEr scores are those given in the project's evaluation report. BLEU measures modified n-gram precision against the references [25]; METEOR aligns words using exact, stem, and synonym matches and combines precision and recall [26]; CIDEr weights n-grams by TF-IDF so that wording shared by many references of an image counts for more [27]. Their scales differ, so the three values are not compared with one another.

Timing. Three tests of ten runs each were performed with the same image, the Instagram platform, the English language, the Creative style, and the gradient_overlay template: the complete generation API (Test A), the rendering API with one fixed caption (Test B), and the complete flow through the Streamlit interface (Test C). The servers ran locally, with the Groq call crossing the internet. The hardware specification was not recorded.

Functional and qualitative evaluation. The application was exercised through its interface, and screenshots were taken of start-up, upload, option selection, caption generation, rendering, download, history retrieval, and record deletion. Six images that are not part of Flickr8k were captioned with different platform, language, and style combinations and the outputs were inspected by the authors. This is a demonstration of behaviour, not a user study or a systematic test suite.

IV. RESULTS AND DISCUSSION

A. Captioning Accuracy

Table IV gives the scores on the 1,000 test images. The BLEU n-gram precisions were 0.708, 0.401, 0.227, and 0.126 for unigrams to four-grams, with a brevity penalty of 1.0. The first figure means that about 71% of the words in the generated captions appear in at least one reference; precision falls as longer phrases are required, as expected for free-

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151092

form descriptions. The generated captions are also longer than the references: 12,152 tokens against 8,044 for the closest-length references, a ratio of 1.51, which the brevity penalty does not punish but which tends to lower precision.

TABLE IV. CAPTIONING RESULTS ON 1,000 FLICKR8K TEST IMAGES

Metric	Score	What it compares
BLEU (4-gram, corpus level)	0.2999	Overlap of word sequences with the references
METEOR	0.5697	Word alignment with the references, including stems and synonyms
CIDEr	0.5191	TF-IDF-weighted n-gram agreement with the references

These numbers describe how closely the generated captions follow the human wording; they do not show that each caption is visually correct. Metric values also depend on tokenisation and implementation details, so we do not set them against figures published for other systems. The notebook displays five test images, and two of them illustrate typical disagreements: a woman signalling to traffic was described as taking a picture of traffic, and two people on what the references call a dock were described as sitting on a bench by the water. In the other three the caption was consistent with the references. Five images are an illustration, not a measurement.

B. Response Time

TABLE V. RESPONSE TIME OVER TEN RUNS PER TEST (SECONDS)

Test	Mean	Median	Range
A: Complete generation API	7.349	7.350	6.32–8.50
B: Image-rendering API	13.550	13.555	13.16–14.05
C: End-to-end interface	23.008	23.275	20.83–25.45

Generation through the API took a mean of 7.35 s, which covers image validation, BLIP inference, the remote language-model call, and the database insert. Rendering took longer, 13.55 s, although it involves no neural model. We did not profile the renderer; its structure suggests candidate causes, since it loads fonts and re-wraps the text at each candidate font size and probes an emoji font, but this is a hypothesis to be tested. The end-to-end mean of 23.0 s is close to the sum of the two API means (20.9 s), and the remaining 2.1 s is consistent with the two sequential requests and interface overhead, which were not measured separately. Within each test the spread was moderate (Table V). The results come from one image and one configuration on one locally hosted deployment; no concurrency, CPU, or memory measurements were taken.

C. Functional Demonstration

The screenshots show the complete flow working in the running application: both servers starting, an image being uploaded and previewed, platform and style being chosen, the base and the Groq captions appearing side by side, the caption being drawn onto the image, and the PNG being offered for download. The history page lists earlier generations with timestamps and options, and a record can be deleted after a confirmation prompt, after which it disappears from the list. These screenshots document that the features operate; they are not a test suite and say nothing about behaviour on inputs that were not tried.

D. Qualitative Examples

Table VI lists the six examples. The Hindi and Marathi outputs are shown with an English gloss prepared by the authors; emoji produced by the model are omitted here for typesetting.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151092

TABLE VI. QUALITATIVE EXAMPLES (GLOSS IN BRACKETS FOR HINDI AND MARATHI)

#	BLIP base caption	Platform / language / style	Groq-generated caption
1	a road near a cliff.	Instagram / English / Travel	Road vibes on the edge of the cliff #roadtrip #cliffside #travel
2	a closeup of a cat ' s face.	Instagram / English / Humorous	When you catch yourself looking at the camera like, “Did someone say treats?” #CatCloseup #FelineFeels #MeowMonday
3	a man in a black shirt is standing in a field.	Facebook / Hindi / Marketing	अपने स्टाइल को नई ऊँचाइयों पर ले जाएँ! हमारे काले शर्ट के साथ हर कदम में आत्मविश्वास। #BlackShirt #Style [Take your style to new heights! With our black shirt, confidence in every step.]
4	a group of people are sitting around a table in front of a large screen.	LinkedIn / Hindi / Formal	टीम के साथ बड़े स्क्रीन के सामने बैठकर रणनीतिक चर्चा कर रहे हैं, जिससे सहयोग और नवाचार को नई दिशा मिलती है। #Teamwork #Collaboration #Leadership [Sitting in front of a large screen with the team and holding a strategic discussion, which gives collaboration and innovation a new direction.]
5	a pink flower with drops of water on it.	X / Marathi / Normal	गुलाबी फुलावर पाण्याचे थेंब चमकत आहेत #गुलाब #फुल [Drops of water are glistening on the pink flower.]
6	two men are walking on a sand dune.	Facebook / Marathi / Creative	सँड ड्यूनवर पायांचे ठसे सोडत, मैत्रीचा प्रवास चालू! #सँडड्यून #साहस #मैत्री [Leaving footprints on the sand dune, the journey of friendship goes on!]

Several patterns appear. The base captions are short and generic. The adapted captions follow the requested platform and style: hashtags on Instagram, a formal register on LinkedIn, and a promotional tone for the Marketing style. In the two English examples the content stays close to the base caption. In some of the other cases the output goes beyond it despite the grounding instruction: the LinkedIn caption describes a team holding a strategic discussion, where the base caption mentions only people sitting around a table in front of a screen; the Marathi caption for the dune image speaks of friendship, a relationship the base caption does not state; and the Marketing caption presents the shirt as the writer’s own product. Whether such additions are acceptable depends on the use, but they show that prompt-level constraints reduce, and do not remove, embellishment. We did not measure how often this happens. Example 2 also shows a spacing artefact (“cat ' s”), which is consistent with the space-separated punctuation of the Flickr8k reference captions the model was trained on. No formal assessment of the fluency of the Hindi and Marathi text was carried out.

E. Limitations and Threats to Validity

Training and evaluation design. The validation split was not used, only one training run was made, no baseline (including the pretrained model) and no ablation were evaluated, and no variance across runs is available. The scores are a documented operating point, not evidence of improvement or of superiority over other models. Flickr8k is small and covers everyday scenes, while the qualitative examples come from other images, so how well the captioner generalises was not measured.

Metrics. BLEU, METEOR, and CIDEr measure agreement with reference wording and depend on implementation choices. The captioner was not scored with a hallucination-oriented or reference-free measure [28], [29], and the adapted captions in the three languages were not scored at all, so the quality of the final output rests on inspection of six examples.

External dependence. Caption adaptation relies on a third-party service, so availability, network latency, and any change to the hosted model affect both the output and the timings. The low temperature reduces but does not remove variation between runs.

Languages and rendering. Only English, Hindi, and Marathi are supported. Marathi is absent from the multilingual benchmark mentioned above [23], which complicates evaluation. Rendering Devanagari and emoji depends on font and shaping support in the host environment.

Deployment and users. The timing results come from one image, one configuration, and an unrecorded local machine. The history is shared and unauthenticated. No user study was done, so claims that the system saves effort or improves accessibility or engagement remain untested.



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026
DOI: 10.17148/IJARCCE.2026.151092

V. CONCLUSION AND FUTURE WORK

This paper described an application that turns an uploaded image into a platform-, language-, and style-specific caption and a captioned image, using a Flickr8k-fine-tuned BLIP model for the description, a Groq-hosted language model for adaptation, Pillow for rendering, and SQLite for history. On 1,000 test images the captioner reached BLEU 0.2999, METEOR 0.5697, and CIDEr 0.5191, and the complete flow took about 23 s per image in the tested configuration. Qualitative examples show the intended adaptation to platform and language, and also cases in which the adapted caption adds content that the base caption does not contain.

The most useful next steps follow from the limitations. They are to evaluate the pretrained model and a validation-monitored fine-tuning run as baselines, to add reference-free and hallucination-oriented measures, to assess the Hindi and Marathi output with native speakers, to profile and speed up the renderer, to avoid sending the image twice, to compare alternative language models, and to run a small user study of the workflow. Per-user history and authentication would be needed before wider deployment.

REFERENCES

[1] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 3156–3164.
[2] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, "A comprehensive survey of deep learning for image captioning," *ACM Comput. Surv.*, vol. 51, no. 6, pp. 1–36, 2019, doi: 10.1145/3295748.
[3] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, PMLR vol. 37, 2015, pp. 2048–2057.
[4] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 6077–6086.
[5] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. 38th Int. Conf. Mach. Learn. (ICML)*, PMLR vol. 139, 2021, pp. 8748–8763.
[6] J. Li, D. Li, C. Xiong, and S. Hoi, "BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *Proc. 39th Int. Conf. Mach. Learn. (ICML)*, PMLR vol. 162, 2022, pp. 12888–12900.
[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. Mach. Learn. (ICML)*, PMLR vol. 202, 2023, pp. 19730–19742.
[8] X. Chen *et al.*, "PaLI: A jointly-scaled multilingual language-image model," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023.
[9] G. Geigle, A. Jain, R. Timofte, and G. Glavaš, "mBLIP: Efficient bootstrapping of multilingual vision-LLMs," in *Proc. 3rd Workshop Adv. Lang. Vis. Res. (ALVR)*, 2024, pp. 7–25, doi: 10.18653/v1/2024.alvr-1.2.
[10] C. Gan, Z. Gan, X. He, J. Gao, and L. Deng, "StyleNet: Generating attractive visual captions with styles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 3137–3146.
[11] K. Shuster, S. Humeau, H. Hu, A. Bordes, and J. Weston, "Engaging image captioning via personality," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 12516–12526.
[12] W. Dai *et al.*, "InstructBLIP: Towards general-purpose vision-language models with instruction tuning," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 36, 2023, pp. 49250–49267.
[13] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 36, 2023, pp. 34892–34916.
[14] M. Hodosh, P. Young, and J. Hockenmaier, "Framing image description as a ranking task: Data, models and evaluation metrics," *J. Artif. Intell. Res.*, vol. 47, pp. 853–899, 2013, doi: 10.1613/jair.3994.
[15] A. Vaswani *et al.*, "Attention is all you need," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017.
[16] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.
[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.
[18] T. Wolf *et al.*, "Transformers: State-of-the-art natural language processing," in *Proc. Conf. Empirical Methods Nat. Lang. Process.: Syst. Demonstrations (EMNLP)*, 2020, pp. 38–45, doi: 10.18653/v1/2020.emnlp-demos.6.
[19] L. Ouyang *et al.*, "Training language models to follow instructions with human feedback," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 35, 2022, pp. 27730–27744.

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026
DOI: 10.17148/IJARCCE.2026.151092

[20] OpenAI, "gpt-oss-120b & gpt-oss-20b model card," *arXiv:2508.10925*, 2025.
[21] G. Wu, "ICU: Conquering language barriers in vision-and-language modeling by dividing the tasks into image captioning and language understanding," *arXiv:2310.12531*, 2023.
[22] I. A. Albadarneh, B. H. Hammo, and O. S. Al-Kadi, "Attention-based transformer models for image captioning across languages: An in-depth survey and evaluation," *Comput. Sci. Rev.*, vol. 58, art. 100766, 2025, doi: 10.1016/j.cosrev.2025.100766.
[23] A. V. Thapliyal, J. Pont-Tuset, X. Chen, and R. Soricut, "Crossmodal-3600: A massively multilingual multimodal evaluation dataset," in *Proc. Conf. Empirical Methods Nat. Lang. Process. (EMNLP)*, 2022, pp. 715–729, doi: 10.18653/v1/2022.emnlp-main.45.
[24] S. K. Mishra, G. Rai, S. Saha, and P. Bhattacharyya, "Efficient channel attention based encoder–decoder approach for image captioning in Hindi," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 21, no. 3, 2022, doi: 10.1145/3483597.
[25] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguist. (ACL)*, 2002, pp. 311–318, doi: 10.3115/1073083.1073135.
[26] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proc. ACL Workshop Intrinsic Extrinsic Eval. Measures Mach. Transl. Summ.*, 2005, pp. 65–72.
[27] R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 4566–4575.
[28] J. Hessel, A. Holtzman, M. Forbes, R. Le Bras, and Y. Choi, "CLIPScore: A reference-free evaluation metric for image captioning," in *Proc. Conf. Empirical Methods Nat. Lang. Process. (EMNLP)*, 2021, pp. 7514–7528, doi: 10.18653/v1/2021.emnlp-main.595.
[29] A. Rohrbach, L. A. Hendricks, K. Burns, T. Darrell, and K. Saenko, "Object hallucination in image captioning," in *Proc. Conf. Empirical Methods Nat. Lang. Process. (EMNLP)*, 2018, pp. 4035–4045, doi: 10.18653/v1/D18-1437.

 This work is licensed under a Creative Commons Attribution 4.0 International License