INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGIES
BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **25**, No 4
Sofia • 2025 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2025-0036

A Novel Hybrid Deep Learning Framework for Image Captioning Using Combined Semantic and Object-Level Features

Harshil Narendrabhai Chauhan, Chintan Bhupeshbhai Thacker

Computer Science and Engineering Department, Parul University, Vadodata, Gujarat, India
E-mails: harshil.chauhan18838@paruluniversity.ac.in chintan.thacker19435@paruluniversity.ac.in

Abstract**:** *Bridging the gap between visual data and human language has been increasingly looked at through the task of automatically generating descriptive captions for images. This places the work within important scopes of accessibility, multimedia search, and human-computer interaction. For this work, we propose a hybrid deep learning model that fuses high-level scene context with localized object information for quality captions. Global image features are obtained through an Xception network, while You Only Look Once, version 8* (*YOLOv8*) *is used to derive object-specific fine details. These visual features are merged and passed to a Bahdanau attention mechanism, which feeds an LSTM decoder to generate context-aware captions. The proposed method was tested on the Flickr8k dataset using BLEU and METEOR metrics; it showed promising improvements over traditional single-stream approaches. Results speak well of the model's ability to deliver better interpretability and accuracy in image captioning***.**

Keywords**:** *Image captioning, Object detection, Semantic features, Visual-text generation, YOLOv8 model.*

1. Introduction

Humans can easily perceive and describe visual information using natural language. Identifying objects and their interactions or narrating the context of a scene is very intuitive. On the other hand, granting machines such an ability has remained a great challenge and requires progress in both computer vision and Natural Language Processing (NLP). Image captioning tries to fill this gap by producing adequate textual descriptions of images automatically – a task that is important in applications ranging from assistive technologies for the visually impaired, content retrieval, medical imaging, to human-computer interaction [1, 2].

The main goal of an image captioning system is to convert image pixels into meaningful and understandable language. It normally proceeds in two steps: visual understanding and linguistic generation. In the first step, salient elements of the picture, such as objects, actions, spatial relations, and attributes, are identified with the help of deep learning-based feature extractors. In the second step, these extracted features are converted into descriptive sentences by language models. Fig. 1 shows the basic block diagram of the image captioning process. Recent advances in deep learning have significantly improved image captioning performance. The attention mechanism has especially transformed model performance because it enables models to pay more selective attention to relevant parts of an image while generating words. Methods such as Bahdanau attention and Luong attention provide dynamic alignment between visual features and the quality and contextual relevance [3, 4].

To improve caption quality, nowadays captioning frameworks often integrate object detection models such as YOLO [5], Faster R-CNN [6], and DETR [7], which help identify objects within an image. These models allow the captioning system to consider both global scene context and object information. This combination helps create captions that are more accurate, specific, and meaningful.

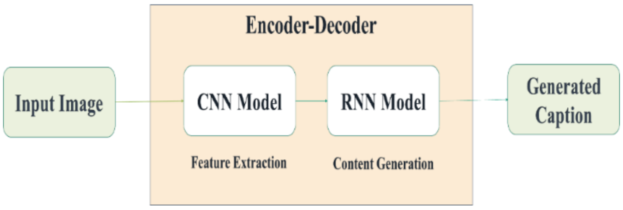


Fig. 1. Basic block diagram of image caption generation [8]

The paper is organized as follows. Chapter 2 provides a literature review of related work in image captioning. Chapter 3 presents the proposed methodology, including the workflow, detailed descriptions, and pseudocode. Chapter 4 describes the dataset, hyperparameters, and evaluation metrics used in this study. Chapter 5 discusses the results achieved by the proposed hybrid model and provides a comparative analysis with existing approaches. Finally, Chapter 6 concludes the study and outlines potential directions for future research.

2. Related work

This section highlights key advancements in the field of image captioning, covering both early methods and modern deep learning approaches. Initially, researchers used template-based and retrieval-based methods. Template-based techniques generated sentences using fixed sentence patterns with placeholders to fill in details from the image [8, 9]. While these methods produced grammatically correct captions, they were limited in flexibility. Due to their reliance on fixed formats, they struggled to adapt to diverse types of images, reducing their usefulness in real-world applications.

On the other hand, retrieval-based methods worked by searching the dataset for images similar to the input image and copying their existing captions. While this approach gave proper sentence structure, it often failed to provide accurate or meaningful descriptions when new or unseen images were used. Both of these early methods helped build the foundation for image captioning. However, they were not good at capturing deeper relationships between different objects and their properties in complex scenes.

With the advancement of deep learning, the encoder-decoder architecture has become the most widely used approach in image captioning. These models typically use Convolutional Neural Networks (CNNs) to extract image features and Recurrent

Neural Networks (RNNs) to generate captions, enabling end-to-end training of the entire system.

Sasibhooshan, Kumaraswamy and Sasidharan [10] introduced a model that combined a Wavelet transform-based CNN with a Virtual Attention Prediction Network (VAPN). The encoder in their model was designed to capture both spatial and channel-wise relationships in the image. For caption generation, they used LSTM units in the decoder. While their model showed promising results, it faced limitations in accurately detecting objects and handling visually complex scenes with many moving elements. The authors suggested that employing more powerful models, such as transformer-based networks, could improve performance. They also recommended exploring video captioning as a future direction.

Al-Malla, Jafar and Ghneim [11] proposed an attention-based encoder-decoder model that utilized Xception for image feature extraction and YOLOv4 for object detection. Their model was evaluated on popular datasets like MSCOCO and Flickr30K, showing improved accuracy compared to earlier methods. However, the authors acknowledged that there remains room for improvement and suggested that more advanced models and enhanced sentence generation techniques could further boost performance.

Wang et al. [12] introduced the Multilayer Dense Attention (MDA) model, which enhanced the standard encoder-decoder framework by incorporating dense attention layers at multiple stages of the CNN. This design allowed the model to capture both global scene context and fine-grained image details. They used Inception-V3 for feature extraction and an LSTM network for caption generation. Their model achieved strong performance on the MSCOCO and Flickr8k datasets, outperforming several previous approaches. These results emphasize the effectiveness of multi-level attention mechanisms in producing more accurate and context-aware image captions.

An attention-based approach was created especially for Hindi language generation by Dhir et al. [13]. In order to match image regions with corresponding Hindi words, their model used CNNs for image feature extraction and an LSTM decoder directed by an attention mechanism. The research tackled the dearth of non-English captioning models and showed encouraging outcomes on a customized dataset, suggesting that multilingual systems for low-resource languages are feasible.

Mishra et al. [14] came up with a Hindi captioning mechanism for images using a dynamic convolutional-based encoder-decoder. They were able to capture spatial dependencies in the image more effectively by replacing conventional CNNs with dynamic convolutions. The model performed better on Hindi datasets, thereby making an important contribution in the area of captioning tools for native languages.

Xu et al. [15] proposed the “Attend and Tell” model, which was among the first to apply visual attention in neural captioning. Their model used a combination of CNN-LSTM, where at each word generated, the decoder could selectively focus on different regions of an image. This allowed for more accurate and expressive descriptions as it resulted from this dynamic alignment.

Vinyals et al. [16] proposed the “Show and Tell” model, where they used a CNN to understand the image and an LSTM to generate the caption. Both parts were trained together in one system. This model worked well on the MSCOCO dataset. However, it did not use any attention mechanism, so it looked at the full image equally all the time. Because of this, it was not very effective in understanding complex images with many details.

Long [17] demonstrated the effectiveness of Convolutional Neural Networks (CNNs), particularly DenseNet, in extracting complex image features due to their dense connectivity and efficient gradient flow. Attention mechanisms, such as channel and spatial attention, have further enhanced model performance by directing focus to the most informative regions of an image. Although primarily applied to Facial Expression Recognition (FER), these techniques are highly relevant to image captioning tasks that require detailed spatial and semantic understanding. This motivates the adoption of hybrid CNN-attention architectures for improving visual caption generation.

Image captioning has evolved significantly from basic rule-based methods to highly sophisticated deep learning architectures. Nevertheless, research gaps persist. Many models struggle to accurately detect fine-grained object relationships and interpret scenes containing multiple objects, overlapping regions, or alphanumeric information. Furthermore, the generation of generic or repetitive captions remains an issue, indicating a lack of commonsense reasoning and contextual understanding in many current models.

3. Proposed methodology

This paper presents a novel hybrid approach for generating the caption from the input image. Meaningful visual features are extracted from the image using the Xception CNN model along with the You Only Look Once, version 8 (YOLOv8) object detection model, which provides a rich feature map. Extracted features are given as an input to the Attention model, which focuses on the useful part of the image, and later on, using an LSTM model sequence by sequence captions should be generated. The proposed hybrid model is illustrated in Fig. 2.

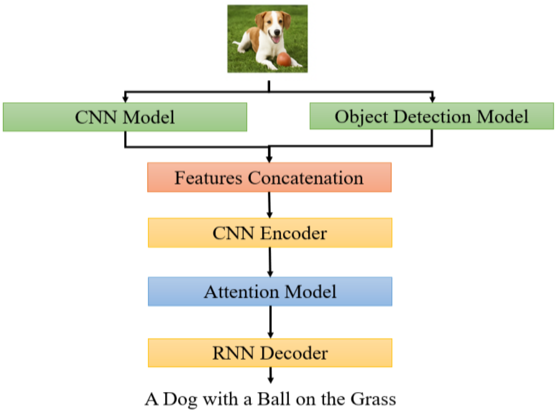


Fig. 2. Proposed hybrid deep learning model

3.1. Feature extraction

In this study, we utilize the pre-trained Xception network, also known as the Extreme Inception model [18], to derive meaningful visual features from the input images. This architecture is designed to perform depth-wise separable convolutions, allowing it to apply one convolutional filter per channel, thereby efficiently capturing spatial structures. To adapt it for feature extraction, the final classification layers are omitted, and the output from the last convolutional stage is retained as the semantic representation of the image. Complementing the CNN-based features, we also integrate the YOLOv8 object detection model [19], developed by Ultralytics, to incorporate object-specific information. For this work, we employ the YOLOv8x.pt configuration to detect prominent regions in each image. The model returns bounding boxes, class probabilities, and category labels. A confidence threshold of 0.2 is applied to filter out unreliable predictions and retain only the most relevant object instances. This combination of global scene context from Xception and localized object cues from YOLOv8 forms the backbone.

3.1.1. Xception model

The Xception (Extreme Inception) architecture is a convolutional neural network that advances the Inception family of models by relying exclusively on depth-wise separable convolutions. It was proposed by Chollet [18] in 2017, showing in Fig. 3, with the key hypothesis that spatial and cross-channel correlations in feature maps can be entirely decoupled, thus allowing for more efficient and effective feature learning.

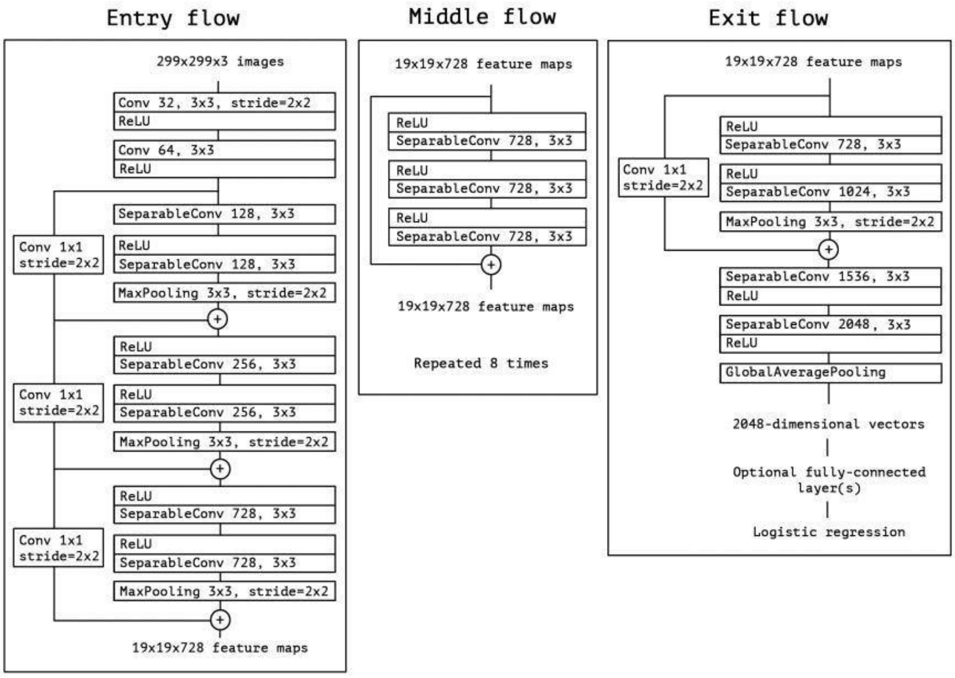


Fig. 3. Xception model architecture [29]

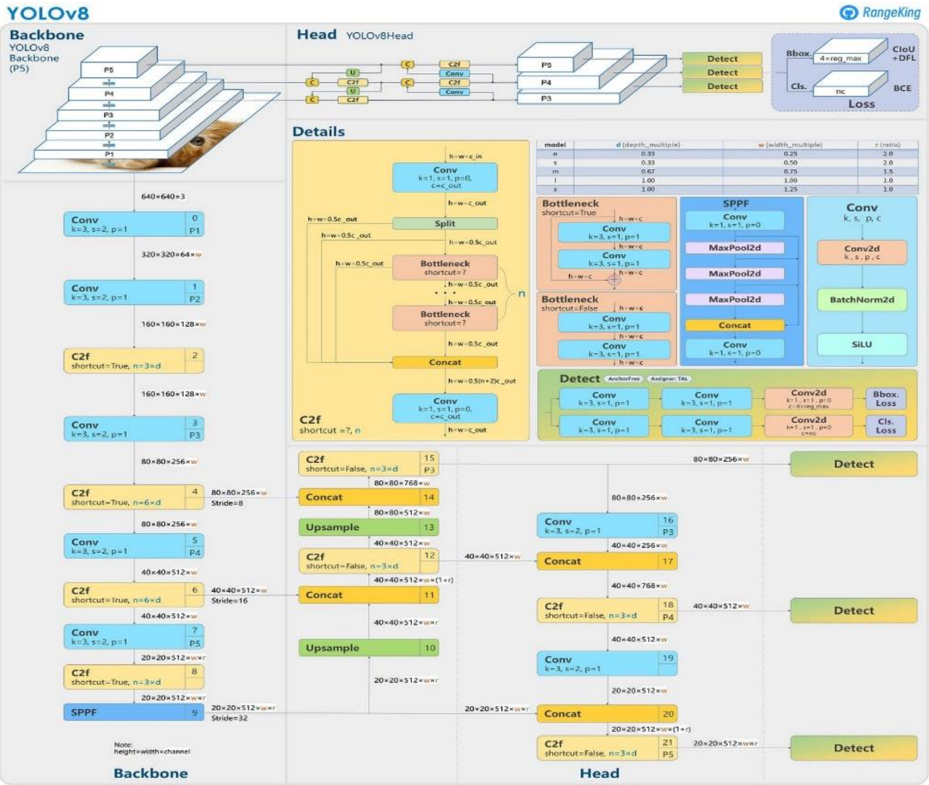
The Xception architecture is composed of a total of 36 convolutional layers, which serve as the primary feature extraction component of the network. These layers are structured into 14 modular blocks, many of which incorporate linear residual connections, a design approach inspired by the ResNet family of models. Rather than relying on conventional convolution operations, Xception employs a depth-wise separable convolution strategy. This technique breaks down standard convolutions into two distinct steps: a depth-wise convolution, which applies spatial filtering independently to each input channel, and a pointwise convolution, which combines the outputs across channels. This architectural choice enhances both computational efficiency and feature discrimination.

$$F_{\mathrm{s}} = \mathrm{Xception}(I) \epsilon R^{M\times d_{\mathrm{s}}}, \tag{1}$$

where F_{s} represents high-level semantic features, M is the number of spatial positions (flattened spatial dimensions), and d_{s} is the feature depth.

3.1.2. YOLOv8 object detection model

To enhance the semantic richness of image features and mimic human visual attention, we incorporate YOLOv8 (You Only Look Once, version 8) [19] as a pre-processing stage in our image captioning pipeline. YOLOv8, developed by Ultralytics, is the latest and most refined iteration of the YOLO object detection family. It is a unified, real-time object detector that balances accuracy, speed, and efficiency.



model	d (depth_multiple)	w (width_multiple)	r (ratio)
n	0.33	0.25	2.0
s	0.33	0.50	2.0
m	0.67	0.75	1.5
l	1.00	1.00	1.0
x	1.00	1.25	1.0

Fig. 4. Yolov8 model architecture [19]

YOLOv8 introduces several key improvements over its predecessors, including a fully convolutional backbone and head, anchor-free detection, and decoupled classification and regression heads. In our image captioning pipeline, YOLOv8 is employed alongside Xception to provide a dual-perspective representation of the visual content. While Xception captures global semantic features of the entire image, YOLOv8 contributes localized object-level features, enriching the model's understanding of specific entities present within the scene. Fig. 4 shows the object detection with the class label and confidence score using Yolov8x.pt model.

(2) $$F_{\mathrm{o}} = \mathrm{Yolov8}(I) \epsilon R^{N\times d_{\mathrm{o}}},$$

where N is the number of objects detected and d_{o} is the object feature dimensionality.

3.2. Feature concatenation

After extracting global semantic features from the image using the Xception model and localized object-level features using the YOLOv8 object detection model, we employ a feature concatenation strategy to combine both representations into a unified visual descriptor. This integration allows the model to simultaneously leverage high-level scene context and detailed object-level information. The global image feature vector $F_{\mathrm{s}} \epsilon R^{M\times d_{\mathrm{s}}}$ is obtained by passing the input image I through the Xception network. Here, M denotes the flattened spatial positions, and d_{s} is the feature depth.

Simultaneously, the YOLOv8 model processes the same input image to detect multiple objects, each with an associated confidence score, class label, and bounding box. The regions corresponding to the detected bounding boxes are cropped and passed through the CNN model to extract localized object features. The resulting YOLO-based feature matrix is denoted as $F_{\mathrm{o}} \epsilon R^{N\times d_{\mathrm{o}}}$. Finally, we concatenate the scene-level and object-level features along the spatial dimension to form a unified feature tensor. The final feature tensor has a size of (101×2048).

(3) $$F = \mathrm{Concat}(F_{\mathrm{s}}, F_{\mathrm{o}}) \epsilon R^{(M+N)\times d}.$$

3.3. CNN Encoder

To effectively integrate the visual features into the language generation process, we employ a CNN Encoder module that transforms the high-dimensional image features into a more compact, semantically meaningful representation suitable for sequence modelling. Given that the visual features from the Xception model and YOLOv8 object detection network have already been extracted and stored, the CNN Encoder does not perform direct convolutional processing. Instead, it acts as a projection layer that maps the pre-extracted visual features into a lower-dimensional embedding space.

The transformed output $F_{\mathrm{e}} \in R^{B\times E}$, where B is the batch size and E is the embedding dimension, serves as the input to the attention mechanism and RNN decoder. This lightweight encoder design ensures that the model efficiently adapts pre-extracted features to a form that is compatible with the decoder, while also retaining the semantic and structural integrity of the visual content.

3.4. Attention model

Attention works like a human eye – focusing only on the parts of the scene that are important while describing an image. Here Bahdanau Attention model introduced by Bahdanau, Cho and Bengio [20], used to focus on the different parts of the image feature map at each time step while decoding. The image feature and the current hidden state are combined. This combination is processed to produce attention scores, which tell the model how important each region of the image is for the current word.

These scores are converted into attention weights through a softmax function, ensuring they sum to one. Using these weights, the model computes a context vector – a weighted average of the image features, emphasizing the important parts of the image for that particular step. Mathematical representation given below:

(4) $$\text{Score} = \tanh\,(W_1\,F + W_2\,h),$$

(5) $$\alpha = \text{softmax}(V \times \text{score}),$$

(6) $$c = \sum_i \alpha_i\,F_i,$$

where F is the set of encoder features, h is the decoder's hidden state, α are the attention weights, and c is the context vector passed to the decoder.

3.5. RNN Decoder

In this model, the Long Short-Term Memory (LSTM) generates one word at a time, maintaining sequence structure and memorizing long-term dependencies. At each time step, the LSTM receives two inputs: the Context vector and the word embedding. Context vector computed by the Bahdanau Attention mechanism and Word embedding, which is a dense representation of the previously generated word or the start token ⟨start⟩ at the initial state.

Think of the LSTM here as the "brain" that remembers what words have already been generated and guides what word should come next, while paying attention to different parts of the image at each step. Dropout (0.5) and recurrent dropout (0.3) are applied to regularize the LSTM and avoid overfitting. L2 regularization is applied on the dense layers following the LSTM to ensure weight penalties during optimization. The hidden states are explicitly initialized to zeros at the start of each new caption generation.

Pseudo code of proposed model

```
Input: Image I
Output: Caption C = {$w_1$, $w_2$, ..., w_T}
Step 1. Feature Extraction
  $F_s$ ← Xception(I)              // Global scene features
  $F_o$ ← YOLOv8(I)                 // Object-level features
Step 2. Feature Concatenation
  $F$ ← Concat($F_s$, $F_o$)             // Unified visual representation
Step 3. CNN Encoder
  $E$ ← CNN_Encoder($F$)
Step 4. Attention Model
  For each decoding step $t$:
    $\alpha_t$ ← BahdanauAttention($E$, $h_{t-1}$)
```

```
        $c_t \leftarrow \sum (\alpha_t \odot \text{E})$                // Context vector
**Step 5. RNN Decoder**
    Initialize $h_0$, $c_0 \leftarrow$ zeros
    $w_0 \leftarrow \langle\text{start}\rangle$, $C \leftarrow \{\}$
    For $t = 1$ to $T$:
        $x_t \leftarrow \text{Embed}(w_{t-1})$
        $z_t \leftarrow \text{Concat}(c_t, x_t)$
        $h_t, c_t \leftarrow \text{LSTM}(z_t, h_{t-1}, c_{t-1})$
        $\hat{y}_t \leftarrow \text{Softmax}(h_t)$
        $w_t \leftarrow \text{argmax}(\hat{y}_t)$
        Append $w_t$ to $C$
        If $w_t == \langle\text{end}\rangle$, break
Return Caption $C$
```

4. Implementation details

This section explains how the hybrid image captioning model was trained and tested. It includes information about the dataset used, how the data was divided for training and testing, the hyperparameters selected, and the methods used to check the model's performance. The main goal is to keep the process clear and repeatable, so that others can easily understand and compare the results with earlier research work.

4.1. Dataset details

The model is tested using the Flickr8k dataset [21], which has 8,087 real-life images. Each image comes with five different captions, written by different people. This dataset includes a variety of scenes, like indoor, outdoor, and action-based situations, which makes it a good choice to test image captioning models. To keep the training smooth and the results fair, the dataset is divided properly as shown in Table 1.

Table 1. Flick8k dataset split

Dataset	Training split	Validation split	Testing split	Total images
Flickr8K	80%	10%	10%	8092

4.2. Hyperparameters

Selecting the right hyperparameters is very important to help the model learn properly and avoid problems like overfitting or underfitting. In this work, the model was fine-tuned by trying different settings to find a good balance between training stability and caption quality. The final values were chosen by keeping in mind both the accuracy of the results and the computational limits. The final set of hyperparameters used for training is shown in Table 2.

Table 2. Hyperparameters used to train the proposed model

Parameter	Value
Batch size	64
Learning rate	0.00001
Epoch	30
Optimizer	Adam
Vocabulary size	10000

4.3. Evaluation parameters

The model's performance was evaluated by the two most popular and widely used evaluation parameters, BLEU [22] and METEOR [23]. BLEU measures n-gram overlapping between reference captions and model-generated captions where unlike BLEU, which focuses on precision, METEOR combines both precision and recall, offering a more nuanced measure that better correlates with human judgment. It rewards partial matches and penalizes incorrect word ordering, making it more sensitive to linguistic quality. The mathematical formula for both parameters is as follows:

$$\text{BLEU} = \text{BP} \times \exp\left(\sum_{n=1}^{N} w_n \log p_n\right), \tag{7}$$

$$\text{METEOR} = \frac{10 \times \text{Precesion} \times \text{Recall}}{\text{Recall} + 9 \times \text{Precesion}}. \tag{8}$$

5. Implementation details

The experimental results demonstrate that combining features of the CNN model with the object detection model improves the quality of generated captions. The code is implemented using the Python programming language. The proposed model has achieved a BLEU 1: 69.99, a BLEU 2: 55.46, a BLEU 3: 37.99, a BLEU 4: 23.05, and a METEOR: 42.85 score, which indicates that the model has outperformed in terms of caption generation. Fig. 5 illustrates the score achieved by the proposed model.

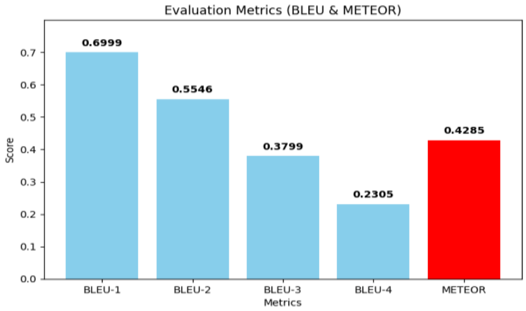


Fig. 5. Evaluation score achieved by the proposed model

5.1. Comparative analysis

To assess the performance of the proposed image captioning model, a comparative evaluation was performed using a set of established benchmark models. The comparison relied on widely accepted metrics, namely the BLEU scores (B1 through B4) and the METEOR score (M). BLEU focuses on the degree of n-gram overlap between the generated captions and the reference annotations, with higher-order scores like BLEU-3 and BLEU-4 providing insight into the model's ability to maintain fluency and contextual relevance. In contrast, METEOR accounts for both precision and recall, offering a more refined assessment by rewarding partial matches and penalizing incorrect word orders – making it better aligned with human judgment of caption quality. The results of this comparative study are detailed in Table 3, and their visual representation is depicted in Fig. 6.

Table 3. Comparative analysis of image captioning models

Model	B1	B2	B3	B4	M
Deep attention [10]	54	22	7.9	3	20
Google NIC [16]	63	41	27	-	-
Log bilinear [23]	65.6	42.4	27.7	17.7	17.31
Hard attention [15]	67	45.7	31.4	21.3	20.30
Proposed Novel Model	**69.99**	**55.46**	**37.99**	**23.05**	**42.85**

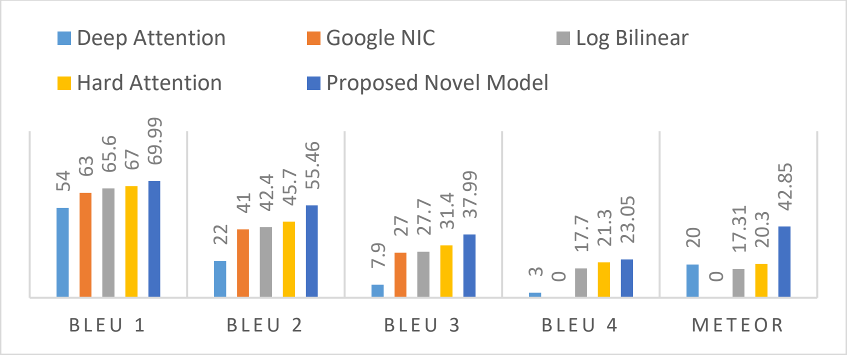


Fig. 6. Comparative analysis of Image Captioning Model

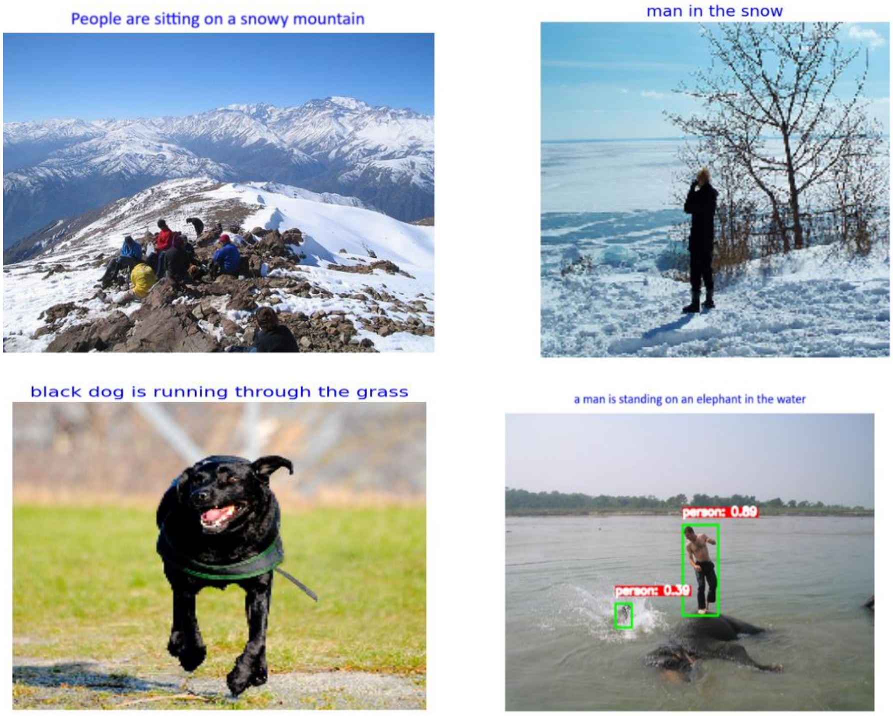

People are sitting on a snowy mountain
man in the snow
black dog is running through the grass
a man is standing on an elephant in the water
person: 0.89
person: 0.39

Fig. 7. Caption generated by the Proposed Hybrid Model

6. Conclusion and future work

This study proposes a hybrid image captioning approach that integrates CNN-based global features and YOLOv8-derived object features, enhanced by Bahdanau attention and LSTM decoding. The model effectively captures both scene context and object-level details, producing more accurate and contextually rich captions than traditional single-stream methods. While the approach shows promising results, it has limitations. Its performance depends on the accuracy of object detection, and evaluation on a small dataset like Flickr8k may restrict generalizability. Additionally, the multi-model design increases computational complexity. Future work can explore advanced encoder-decoder architectures, hyperparameter optimization, and multilingual caption generation in languages such as Gujarati, Marathi, Tamil, Urdu, and Kannada.

References

1. Bai, S., S. An. A Survey on Automatic Image Caption Generation. – Neurocomputing, Vol. **311**, 2018, pp. 291-304. DOI: 10.1016/j.neucom.2018.05.080.
2. Makav, B., V. Kılıç. A New Image Captioning Approach for Visually Impaired People. – In: Proc. of 11th International Conference on Electrical and Electronics Engineering (ELECO'19), Bursa, Turkey, 2019, pp. 945-949. DOI: 10.23919/ELECO47770.2019.8990630.
3. Hossain, M. Z., F. Sohel, M. F. Shiratuddin, H. Laga. A Comprehensive Survey of Deep Learning for Image Captioning. – ACM Computing Surveys, Vol. **51**, 2019, No 6, pp. 1-36. DOI: 10.1145/3295748.
4. Wang, H., Y. Zhang, X. Yu. An Overview of Image Caption Generation Methods. – Computational Intelligence and Neuroscience, 2020, pp. 1-13. DOI: 10.1155/2020/3062706.
5. Redmon, J., S. Divvala, R. Girshick, A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. – In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779-788.
6. Ren, S., K. He, R. Girshick, J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. – Adv. NeuReal Inf. Process Syst., Vol. **28**, 2015, pp. 91-99.
7. Eluri, Y., N. Vinutha, G. S. Abhiram. Image Captioning Using Visual Attention and Detection Transformer Model. – In: Proc. of IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT'24), IEEE, July 2024, pp. 1-4.
8. Poddar, A. K., R. Rani. Hybrid Architecture Using CNN and LSTM for Image Captioning in the Hindi Language. – Procedia Computer Science, Vol. **218**, 2023, pp. 686-696. DOI: 10.1016/j.procs.2023.01.049.
9. Xiao, X., L. Wang, K. Ding, S. Xiang, C. Pan. Deep Hierarchical Encoder-Decoder Network for Image Captioning. – IEEE Transactions on Multimedia, Vol. **21**, 2019, No 11, pp. 2942-2956. DOI: 10.1109/TMM.2019.2915033.
10. Sasibhooshan, R., S. Kumaraswamy, S. Sasidharan. Image Caption Generation Using Visual Attention Prediction and Contextual Spatial Relation Extraction. – Journal of Big Data, Vol. **10**, 2023, No 18. DOI: 10.1186/s40537-023-00693-9.
11. Al-Malla, M. A., A. Jafar, N. Ghneim. Image Captioning Model Using Attention and Object Features to Mimic Human Image Understanding. – Journal of Big Data, Vol. **9**, 2022, No 20. DOI: 10.1186/s40537-022-00571-w.
12. Wang, E. K., X. Zhang, F. Wang, T. Y. Wu, C. M. Chen. Multilayer Dense Attention Model for Image Caption. – IEEE Access, Vol. **7**, 2019, pp. 66358-66368. DOI: 10.1109/ACCESS.2019.2917771.
13. Dhir, R., S. K. Mishra, S. Saha, P. Bhattacharyya. A Deep Attention-Based Framework for Image Caption Generation in the Hindi Language. – Computación Y Sistemas, Vol. **23**, 2019, No 3. DOI: 10.13053/cys-23-3-3269.

14. Mishra, S. K., S. Sinha, S. Saha, P. Bhattacharyya. Dynamic Convolution-Based Encoder-Decoder Framework for Image Captioning in Hindi. – ACM Transactions on Asian and Low-Resource Language Information Processing, 2022. DOI: 10.1145/3573891.
15. Xu, K., et al. Attend and Tell: Neural Image Caption Generation with Visual Attention. – In: Proc. of 32nd Int. Conf. Mach. Learn. (ICML'15), Lille, France, 2015, pp. 2048-2057.
16. Vinyals, O., A. Toshev, S. Bengio, D. Erhan. Show and Tell: A Neural Image Caption Generator. – In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3156-3164.
17. Long, D. T. Efficient DenseNet Model with Fusion of Channel and Spatial Attention for Facial Expression Recognition. – Cybernetics and Information Technologies, Vol. **24**, 2024, No 1, pp. 171-189.
18. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. – In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1251-1258.
19. Ultralytics. Yolov8 Anchor-Free Bounding Box Prediction, Issue 189, 2023. **https://github.com/ultralytics/ultralytics/issues/189**
20. Bahdanau, D., K. Cho, Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. – arXiv preprint arXiv:1409.0473, 2014.
21. Hodosh, M., P. Young, J. Hockenmaier. Framing Image Description as a Ranking Task: Data, Models, and Evaluation Metrics. – Journal of Artificial Intelligence Research, Vol. **47**, 2013, pp. 853-899. DOI: 10.1613/jair.3994.
22. Papineni, K., S. Roukos, T. Ward, W.-J. Zhu. BLEU. – In: Proc. of 40th Annual Meeting on Association for Computational Linguistics (ACL'02), 2001, p. 311. DOI: 10.3115/1073083.1073135.
23. Banerjee, S., A. Lavie. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. – In: Proc. of ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, 2005, pp. 65-72.
24. Kiros, R., R. Salakhutdinov, R. Zemel. Multimodal Neural Language Models. – In: Proc. of International Conference on Machine Learning, (PMLR'14), June 2014, pp. 595-603.
25. Kamangar, Z. U., G. M. Shaikh, S. Hassan, N. Mughal, U. A. Kamangar. Image Caption Generation Related to Object Detection and Colour Recognition Using a Transformer-Decoder. – In: Proc. of 4th International Conference on Computing, Mathematics and Engineering Technologies (iCoMETs'23), Sukkur, Pakistan, 2023, pp. 1-5. DOI: 10.1109/iCoMET57998.2023.10099161.
26. Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin. Attention is All You Need. – arXiv. DOI: 10.48550/arXiv.1706.03762, 2017.
27. Mishra, S. K., Harshit, S. Saha, P. Bhattacharyya. An Object Localization-Based Dense Image Captioning Framework in Hindi. – ACM Transactions on Asian and Low-Resource Language Information Processing, Vol. **22**, 2022, No 2, pp. 1-15. DOI: 10.1145/3558391.
28. Kaur, M., H. Kaur. An Efficient Deep Learning Based Hybrid Model for Image Caption Generation. – International Journal of Advanced Computer Science and Applications, Vol. **14**, 2023, No 3. DOI: 10.14569/IJACSA.2023.0140326.

Received: 20.06.2025, First revision: 22.09.2025, Accepted: 25.09.2025