Taylor & Francis
Taylor & Francis Group

Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: 0005-1144 (Print) 1848-3380 (Online) Journal homepage: www.tandfonline.com/journals/taut20

Image captioning deep learning model using ResNet50 encoder and hybrid LSTM–GRU decoder optimized with beam search

P. V. Kavitha & V. Karpagam

To cite this article: P. V. Kavitha & V. Karpagam (2025) Image captioning deep learning model using ResNet50 encoder and hybrid LSTM–GRU decoder optimized with beam search, Automatika, 66:3, 394-410, DOI: 10.1080/00051144.2025.2485695

To link to this article: https://doi.org/10.1080/00051144.2025.2485695

Published online: 16 Apr 2025.

AUTOMATIKA
2025, VOL. 66, NO. 3, 394–410
https://doi.org/10.1080/00051144.2025.2485695

Image captioning deep learning model using ResNet50 encoder and hybrid LSTM–GRU decoder optimized with beam search

P. V. Kavitha and V. Karpagam

Department of Artificial Intelligence and Data Science, Sri Ramakrishna Engineering College, Coimbatore, India

ABSTRACT

Image captioning is a fascinating and fast-evolving research project that integrates two domains: Natural Language Processing and Computer Vision. Creating appropriate captions is a difficult task due to the many activities portrayed in the backdrop image. To mitigate these drawbacks, the envisioned work employs a ResNet50 encoder for image feature extraction and a Hybrid LSTM–GRU decoder optimized with Beam Search to produce text descriptions. Beam search is a search technique that enables caption generation with higher quality and consistency by investigating many paths in the search space and choosing the most likely option based on a score or probability. The findings compare CNN models such as VGG16, InceptionV3, ResNet50 and DenseNet121 with language model LSTM in terms of loss and accuracy on the Flickr8k dataset. To further boost the performance of caption quality, the proposed method uses ResNet50 + Hybrid LSTM–GRU with Beam search, which produces a good accuracy of 0.8932 and a lower loss of 0.4013 on the Flickr8k dataset. The proposed method, ResNet50 + hybrid LSTM–GRU with Beam Search, beats the findings of the aforementioned encoder–decoder models with Greedy Search in terms of the BLEU score of 0.6034.

ARTICLE HISTORY
Received 26 September 2024
Accepted 25 March 2025

KEYWORDS
Deep learning; greedy search and beam search; convolutional neural network; Resnet50; Hybrid LSTM–GRU

1. Introduction

The technique of producing captions automatically from provided photographs is known as image captioning. Not only does it comprehend the image's contents, but it also annotates them using a caption format [1]. Existing image captioning methods often struggle with accurately capturing the full context of complex scenes, especially when images contain ambiguous or multiple interpretations. The challenge lies in effectively integrating visual and textual modalities to produce high-quality, coherent captions. It uses an encoder–decoder paradigm to convert the input image into written descriptions. The encoder model uses computer vision techniques to handle the image element, while the language decoder model uses natural language processing techniques to handle language descriptions. An image caption must not only list the things in the picture but also describe the backdrop environment and the attributes of each object. After that, it produces accurate, concise, appropriate, grammatically and semantically sound textual descriptions. Figure 1 depicts the workflow of image captioning.

Even with major advances in the deep learning and language processing domains, the technique of creating captions from image data is not aligned with the actual content of the image. Creating accurate captions is still a difficult task because of the multitude of activities portrayed in the background image. Typically, an encoder is a CNN model to obtain visual features. Based on visual data, the RNN model is employed as a language decoder to produce text captions. One type of RNN structure that can stop gradient fading is called long-term memory (LSTM). LSTM helps with the vanishing gradient issue in RNNs, although it is not a perfect solution. Because of this, tests are conducted on specific LSTM versions, including Bi-LSTM and Stacked LSTM, although the outcomes are not more precise or pertinent to the image. Consequently, to enhance caption quality, the modified RNN structure GRU is employed.

GRU extracts text context information more effectively, handles longer text sequences more efficiently and solves the gradient diminishing problem. Because of the GRU architecture's simplicity, model parameters are greatly reduced. Training efficiency can be significantly increased, and the process is straightforward. Since there are no output gates in these long- and short term memory networks, all the memory unit's contents are recorded into the network at each step. GRU versions such as Stacked GRU and Bi-GRU are also tested. However, the findings are not more accurate or relevant to the image, and the captions are not sufficient. The proposed method replaces the traditional image captioning model by combining the ResNet50 Cum Hybrid LSTM–GRU model with Beam Search decoding to improve the quality of the visual description, it

CONTACT P. V. Kavitha kavithapv572@gmail.com Department of Artificial Intelligence and Data Science, Sri Ramakrishna Engineering College, Coimbatore - 641022, Tamilnadu, India

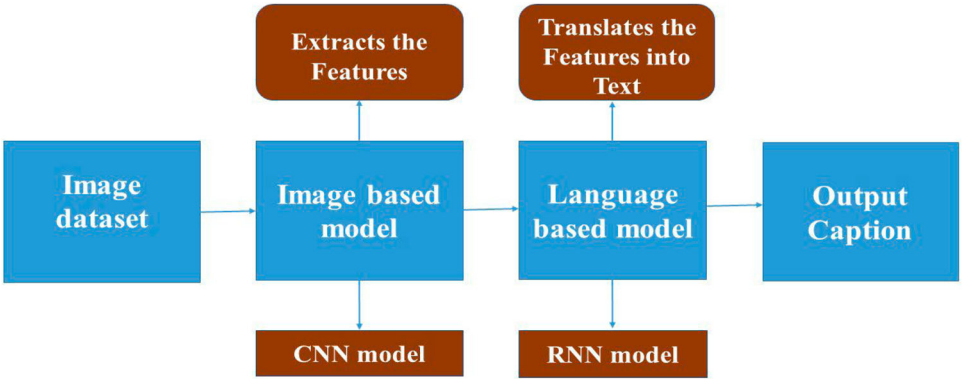

Figure 1. Image captioning workflow.

fully capture the visual features and generate a more precise description.

1.1. Challenges in enhancement of high-quality caption

The intention of this proposed approach is to build a picture captioning model with good-quality captions using the Transfer learning CNN model ResNet50 with Hybrid LSTM–GRU. One of the challenges behind a good-quality caption is selecting the appropriate CNN model that captures the full context of an image, including nuanced details and relationships between objects. The CNN model needs to understand the spatial and semantic relationships within an image. On the other hand, it is challenging to create captions that accurately convey the desired idea in the case of images containing ambiguous or complex scenes that can be interpreted in multiple ways. Integrating information from both visual and textual modalities effectively is crucial for generating high-quality captions. Therefore the enhancement of caption quality is predicted based on the concept of the Beam Search decoding algorithm, which is explained in the upcoming section.

1.2. Research significance

VGG16 and ResNet50 were chosen for image feature extraction due to their proven effectiveness in capturing rich, hierarchical visual features. VGG16, with its simple architecture and deep convolutional layers, excels in feature extraction for general image classification tasks. ResNet50, on the other hand, utilizes residual connections to combat vanishing gradients, making it highly effective for training deeper networks and achieving better performance on complex image data. While models like EfficientNet may offer better efficiency, VGG16 and ResNet50 were selected based on their established success in image captioning tasks and their ability to provide high-quality visual features for the proposed model.

The importance of this research is based on the Transfer Learning CNN model cum Hybrid LSTM–GRU model with Beam Search decoding algori thm to build high caption quality. This work mainly contributes on understanding the integration of various CNN models such as VGG16, InceptionV3, ResNet50 and DenseNet121 with a language model, LSTM. ResNet50 has lower loss and more accuracy than the other models for improved caption quality. The focus of this study is to enhancing accuracy and relevance of generated captions by capturing the detailed visual features and effectively integrating them with the textual descriptions.

- Apart from the ResNet50 model, LSTM variations, including Stacked LSTM and Bi-LSTM, and GRU variations, including Stacked GRU and Bi-GRU models, have also been thoroughly investigated to properly extract the equivalent visual features to produce captions from the image features.
- Furthermore, this experimental work focuses on the ResNet50 model with a hybrid combination of LSTM and GRU to achieve good accuracy and low loss, which significantly improves the caption quality to be more accurate and pertinent to the image.
- By leveraging the strengths of ResNet50 for image feature extraction and the Hybrid LSTM–GRU for language generation, the model addresses limitations in existing methods. The paper provides an improved efficiency in caption quality, attaining higher accuracy and lower loss on the Flickr8k dataset compared to traditional approaches.

The overview of this work is explained in the succeeding sections. We discuss similar studies of image captioning in Section 2. Section 3 discusses the backdrop of image captioning encoder–decoder models and decoding methods such as Greedy Search and Beam Search. Section 4 introduces the background of our envisioned captioning method, which is then explained

in Section 5. Sections 6 and 7 discuss their experimental results, its discussions and their conclusions respectively.

2. Related work

The current section describes the previous works of image captioning. The conventional approaches to image captioning are template and retrieval-based. A template-based approach uses fixed slot templates to generate captions. These slots are occupied with three tags, such as object, action, scene. This approach uses predefined templates to create grammatically correct captions but cannot produce variable-length captions [2]. The retrieval-based approach extracts the k closest photos connected with their captions from the pre-constructed image caption repository, and it is unable to build fresh captions [3]. Deep learning has made considerable advances, particularly in image captioning models that integrate convolutional and recurrent neural networks. This approach has more flexibility, can produce more diverse phrases using the data, and performs better than systems based on templates and retrieval [4]. The most commonly used technique for describing images is the CNN encoder–RNN decoder technique, which was invented by Mao et al. [5]. The image features are extracted from a CNN encoder to create feature vectors, and then from an RNN decoder to create textual descriptions. The RNN network has the flaws of gradient expansion or gradient disappearance, despite the fact that this model has increased the effect of picture description to a certain level. The recurrent neural network is unable to learn connection information or manage such "long-term dependency" when solving long-distance information.

Vinyals et al. [6] used the LSTM model instead of RNN to rectify the flaws of gradient expansion and improve the image captions. LSTM can capture only one-way timing information, and it does not concentrate on the words that have the most impact on the outcome. TT-LSTM, a two-layer LSTM model, was used for image caption synthesis. It uses XceptionNet for visual feature extraction and two LSTM layers for text/captions generation [7]. XuJia et al. [8] proposed a model to generate long phrases. To tackle the problem of producing short words, it provides more global semantic information as input to the LSTM block. To extract semantic information, it additionally combines normalization techniques and an additional RNN hidden layer. Wang et al. [9] proposed a CNN encoder model and a Bi-LSTM decoder model. To learn the image textual descriptions, it uses past and future data in a high-level semantic space. Yao et al. [10] proposed a unique architecture, LSTM with Copying Mechanism (LSTM-C), which includes copying concepts into CNN and RNN image captioning frameworks. Following that, this model employs a language decoder RNN with a copying mechanism to choose words from objects and position them appropriately in the text phrase. Alam et al. [11] used par-inject architecture and examined five image encoders for image captioning, including Vgg16, InceptionV3, ResNet50, Densenet201 and Xception, and calculated loss and accuracy throughout training. To create sentences from visual features, an LSTM decoder has been used. Resnet50 outperformed all other encoders in terms of accuracy. Rahman et al. [12] created an encoder ResNet50 model for visual feature extraction and a decoder bidirectional LSTMs model for generating the corresponding Bangla captions. Yucong et al. [13] proposed a technique for image semantic description that combines Bi-GRU and the Attention model. CNN starts by obtaining the image's global attributes. Second, CNN has implemented a channel attentiveness mechanism that gives channel information more weight. Finally, Bi-GRU – Attention is utilized to create a semantic description that can accurately capture the content of the image utilizing the feature information that has been examined. Verma et al. [14] proposed VGG16 Hybrid Places 1365 encoder model and LSTM decoder model for generating textual descriptions from picture features in the labelled Flickr8k and MSCOCO Captions datasets.

Yan et al. [15] introduced AICRL, a one-joint model that employs a ResNet50 encoder and LSTM decoder with soft attention. This model generates captions from photos in the MSCOCO 2014 dataset. Zhang et al. [16] proposed a semantic element identification framework for image captioning, which includes an object detection module to anticipate image sections and LSTM model to provide local descriptions for these regions. Semantic features are created using local, global descriptions and object areas. The proposed Element Embedding LSTM model predicts language descriptions by merging CNN and semantic characteristics. Chang et al. [17] proposed a captioning model that uses VGG16 and LSTM with attention to give textual descriptions of images. Additionally, the encoder model now includes Mask R-CNN with OpenCV, which recognizes objects and analyses colours. Zhang et al. [18] projected a gLSTM approach, where the visual features of RoI are used as guiding information for gLSTM for creating more accurate captions. Ding et al. [19] suggested a new bottom-up visual attention mechanism that detects regions by combining basic-level properties like image quality with high-level properties like face recognition. These feature vectors with weight matrices adjusted by the Gaussian filter are passed into the decoder model to produce image textual descriptions.

Al-Malla et al. [20] suggested a captioning encoder–decoder model with object detection features that employs YOLOv4 to increase the quality of output captions. The encoder employs Xception, and the decoder employs attention to GRU architecture. Tejal et al. [21]

suggested a captioning approach that includes data gathering, uncaptioned image selection, appearance and texture extraction, and artificial picture caption creation. After gathering data from two public sources, ARO (Adaptive Rain Optimization) helps to choose the images with non-captions. The SDM (Spatial Derivative and Multiscale) method extracts appearance features, whereas the WPLBP method extracts texture features. The ECANN (Extended Convolutional Atom Neural Network) model uses a caption reuse mechanism to choose the most accurate caption by merging the CNN and LSTM architectures.

3. Proposed methodology

Image captioning is the technique of building written descriptions for images. The CNN architecture extracts image feature vectors, and these captured features are given into the LSTM architecture to generate the text description. To ensure an input data quality and improve the model performance, the data pre-processing process is more essential. Initially, the band-pass filter utilized for concentrating the frequencies important for the speech analysis, to eliminate the unwanted noise from the audio signals. It determined using Equation (1) as

$$y(t) = \int_{-\infty}^{\infty} x(t-\tau)h(\tau)d\tau \tag{1}$$

where $x(t)$ refers the input signal and $\boldsymbol{h(\tau)}$ denotes the impulse response. Normalization utilized for ensured all audio signals on the same scale following noise reduction. It divides by the signal with the maximum absolute value as Equation (2),

$$x_{norm}(t) = \frac{x(t)}{\max(|x(t)|)} \tag{2}$$

The time domain signal transformed into the frequency-domain information by extracting the characteristics by using the Short-Time Fourier Transform (STFT). It is mathematically represented as Equation (3):

$$X(t,f) = \int_{-\infty}^{\infty} x(\tau)w(\tau-t)e^{-j2\pi f\tau}d\tau \tag{3}$$

where $w(\tau - t)$ refers the window function. These pre-processing steps are more significant for improving the signal quality and offers a feature that fed into the ML models are informative.

3.1. Transfer learning CNN models – encoder

This section deals with transfer learning CNN encoder models such as VGG16, InceptionV3, ResNet50 and DenseNet121. The LSTM decoder model has been investigated in this section. LSTM models such as Stacked LSTM and Bi-LSTM, GRU models such as Stacked GRU and Bi-GRU have been explored. The proposed approach, i.e. ResNet50 with Hybrid LSTM–GRU, is explained in this section.

The CNN feature extraction model extracts notable feature vectors from images, often in a vector form of fixed-length. Transfer learning is a method for representing features from a pre-trained model that avoids training a new model from scratch. A model that is already trained is frequently built on ImageNet, and the weights obtained from the trained model can be utilized with a custom neural network for any other similar application. This strategy not only shortens training time but also eliminates generalization errors. The transfer learning CNN encoder models used are VGG16, InceptionV3, ResNet50 and DenseNet121 are discussed below.

3.1.1. VGG16

VGG16 is a prominent CNN architecture [22] that is widely used for image classification. It is used in image captioning to extract information from source photos. The VGG16 model extracts high-level visual features from an input image, these image features are passed to LSTM which in turn helps to construct the caption. VGG16 uses 14 million pictures and 1000 classes. The architecture and its workflow are enclosed in the supplementary information (1.1 – VGG16)

3.1.2. Inceptionv3

InceptionV3 [23] is a deep CNN architecture optimized for image classification tasks. It is a 48-layer network that is trained on the ImageNet dataset for 1000 classes. The pre-trained weights of this network capture high-level visual features, which are then passed into LSTM to generate natural language descriptions. The architecture and its workflow are enclosed in the supplementary information are included in supplementary information (1.2 – InceptionV3).

3.1.3. Resnet50

It is known as a residual network consisting of 50 layers deep [24]. It uses skip connections to solve the problem of vanishing gradient. The addition of input to a convolution block is called a skip connection. It is the only algorithm that trains using F(X) rather than the result "Y". To put it another way, F(X) = 0 implies that Y = X.

3.1.3.1. Skip connection. Figure 2 illustrates the skip connection.

Skip Connection is a connection that skips some layers of the model. With the skip connection, the output is: $Y = F(X) + x$.

ResNet50 consists of two blocks. They are:

1. Identity Block
2. Convolution Block

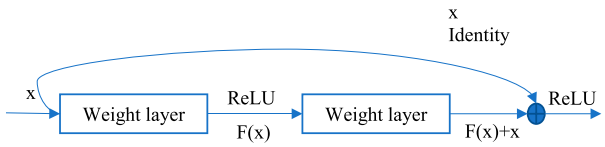

Figure 2. Skip connection of ResNet50 for solving vanishing gradient problem.

1. *Identity Block:* This block is used when input size = = output size, x is added to the output layer. Figure 3 represents the Identity block.

The input and output sizes are equal, the Identity Block is used that enables the input x to be added directly to block's output. It facilitates the learning of residual functions while preserves input spatial dimensions. It utilizes the convolutional layers, batch normalization and ReLU activations.

2. *Convolution Block:* This block is used when input size! = output size, a convolution layer is added to the output layer.

Convolution block is depicted in Figure 4.

The input and output sizes vary and a convolutional layer required to correct the dimensions, the Convolution Block is used. The convolutional layers, batch normalization and ReLU activation functions are included in this block [25]. They are padding the input value and performing 1 × 1 convolution. For padding the input value, the output size is computed using Equation (4).

$$\frac{x+2p-f}{s}+1 \times \frac{x+2p-f}{s}+1 \tag{4}$$

where "x", "p", "f" and "s" represent input size, padding, number of filters and stride, respectively. For 1 × 1 convolutional layers, the output size is calculated using Equation (5).

$$\frac{x}{2} \times \frac{x}{2} \tag{5}$$

The pooling layer (2 × 2) reduces the size of the image. The fully connected layer is skipped, as image feature vectors are given to LSTM [26]. The ResNet50 architecture is depicted in Figure 5.

ResNet50 architecture consists of input layer which takes an RGB image size (224, 224, 3). It consists of convolution layer containing 7 × 7 kernel with 64 extra kernels using a 2 stride, max pooling layer of 2 strides, 3 rounds of 9 layers: 3 × 3, 64 kernels, 1 × 1, 64 kernels, and 1 × 1, 256 kernels, 4 repetitions of 12 layers: 1 × 1, 128 kernels, 3 × 3, 128 kernels and 1 × 1 512 kernels, 6 repetitions of 18 layers: 1 × 1, 256, 2 kernels, 3 × 3, 256 kernels and 1 × 1, 1024 kernels, and 3 repetitions of 9 layers: 1 × 1, 512 kernels, 3 × 3, 512 kernels and 1 × 1, 2048 kernels. LSTM model uses 49 feature vectors of size 2048 as input, removing the need for a fully connected layer and pooling [27].

3.1.4. Densenet121

DenseNet [28] is a type of neural network model used for visual feature extraction. It concatenates the output of the next layer with the previous layer's output. It uses composite function and was created primarily to improve the vanishing gradient-induced reduction in accuracy in high-level neural networks. The network has N $(N+1)/2$ direct connections. N is the number of architectural layers. The feature vectors from this model are passed to the LSTM model to produce caption. The architecture and the explanation of the model are enclosed in supplementary information (1.3 – DenseNet121).

3.2. Language decoder models

3.2.1. Long short term memory (LSTM)

The language model uses LSTM for caption generation [29]. It is an RNN approach that consists of three gates. Cell state is the information that flows through

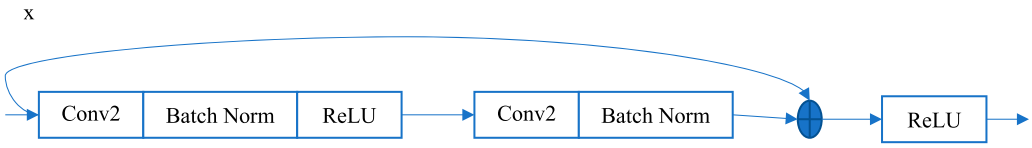

Figure 3. Identity block of ResNet50.

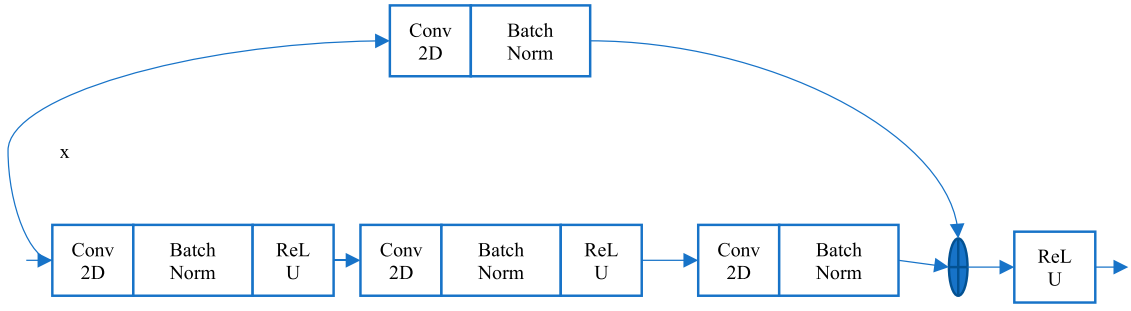

Figure 4. Convolution block of ResNet50.

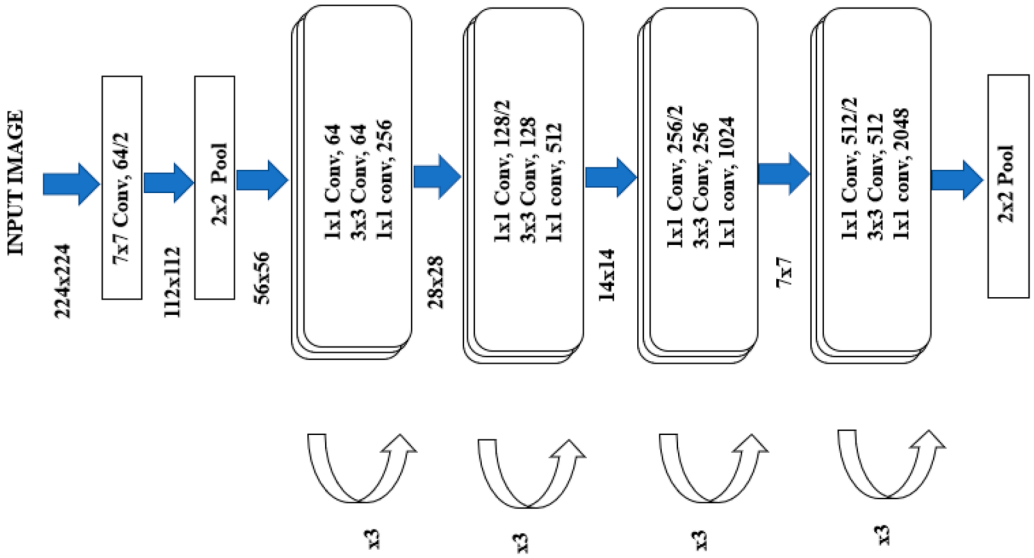

Figure 5. ResNet50 encoder model for image feature extraction.

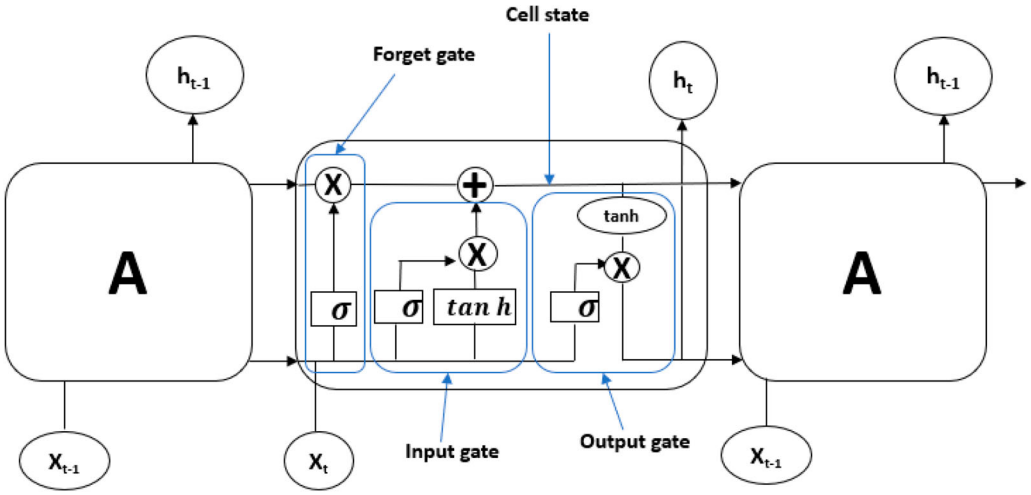

Figure 6. LSTM decoder model for caption generation.

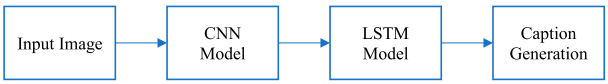

Figure 7. Image captioning – CNN model (ResNet50) for extraction of image features with simple LSTM for generation of captions.

the path. Gates let information through the cell state. LSTM architecture is illustrated in Figure 6. It consists of three gated units such as forget gate, input gate and output gate [30]. The working model of LSTM decoder is enclosed in the supplementary information (2.1 – Long Short Term Memory (LSTM)).

Here is the workflow of CNN with simple LSTM given in Figure 7.

In this model, the CNN (ResNet50) initially extracts the image features that passed to the LSTM model for generating the caption. The CNN concentrates on the learning, while the LSTM interprets the features and generates corresponding captions sequentially that increase the complexity. The LSTM is used to enhance the quality of generated captions by capturing the relationships among the objects and actions in the image [31].

3.2.2. Stacked long short term memory (stacked LSTM)

Stacked LSTMs [32] are now a credible solution for generating captions. A stacked LSTM architecture is an LSTM model that consists of many LSTM layers stacked one above the other. Simple LSTM struggles with CNN-learned hierarchical representations for caption generation [33]. Stacking LSTM layers can help the model catch hierarchical patterns and dependencies in picture features, helping it gain a more complete knowledge of the visual information. The CNN model with Stacked LSTM can model longer sequences, making it better suited to remembering essential visual information for creating captions. Figure 8 illustrates stacked LSTMs [34].

Here is the workflow of CNN with stacked LSTM given in Figure 9.

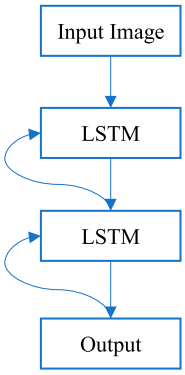

Figure 8. Stacked LSTM for caption generation.

ResNet50 utilized for capturing the high-level data by extracting the deep features from the input image. To enhance model's capacity to offer a more precise and cohesive captions, these features are subsequently through the stacked LSTM network that makes possible by the several LSTM layers. The model offers a comprehend intricate relationships of image and creates more contextually relevant to the captions due to the stacked LSTM architecture [35].

3.2.3. Bidirectional LSTM

It stores the information in both ways: from the past to the present and from the past to future. Bi-LSTM [36] is illustrated in Figure 10.

To integrate the contextual information from past and future, the Bi-LSTM model used in Figure 10 provides the sequential data both forward and backward. The model produces more precise captions due to this bidirectional flow that enhances the comprehension of sequence flow. The backward layer records the future context is more crucial for the caption generation task with the forward layer records before information. These two layers are combined, and the model's capacity to offer an insightful and well-informed result is improved.

Here is the workflow of CNN with Bi-LSTM given in Figure 11.

The ResNet50 model offers an illustration of Figure 11 that extracts the features from the image. A Bi-LSTM network carries out the bidirectional sequence learning then processes these characteristics. Bi-LSTM improves the caption generation process by offering the sequence in both directions, offering the model to capture the context from both previous and upcoming inputs. The model offers the more precise and context-aware captions for the two-way process [37].

3.2.4. Gated recurrent unit (GRU)

GRU comprises two gate mechanisms known as the Update gate and the Reset gate [38]. Figure 12 shows the architecture of the GRU. The working mechanism of GRU is enclosed in the supplementary information (2.2 Gated Recurrent Unit).

The GRU decoder model utilizes the retrieved image features to create the captions by processing the sequential data. An LSTM type as GRU becomes a computationally efficient with the fewer parameters to capture the long-term dependencies [39]. The controlling information flow with the update and reset gates, the model enables to concentrate on the pertinent data for creating the captions. This model learns to connect among the words and visual attributes to produce a textual descrip-

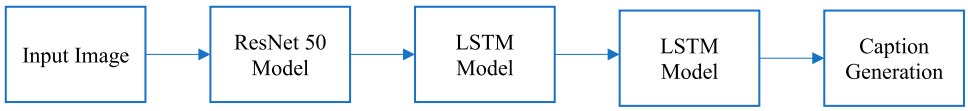

Figure 9. Image captioning – ResNet50 for image feature extraction with Stacked LSTM for caption generation.

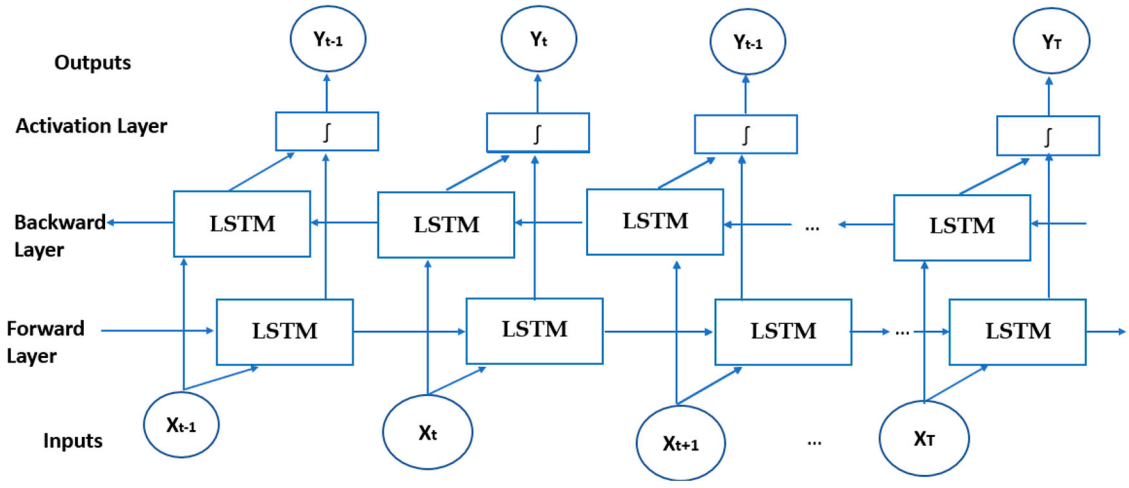

Figure 10. Bi-LSTM decoder model for caption generation.

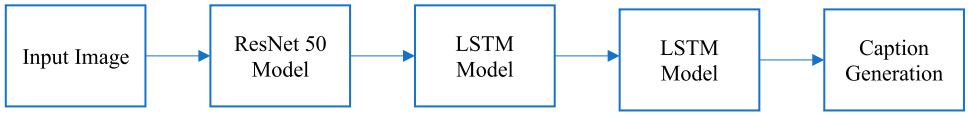

Figure 11. Image captioning – ResNet50 for feature extraction with Bi-LSTM for caption generation.

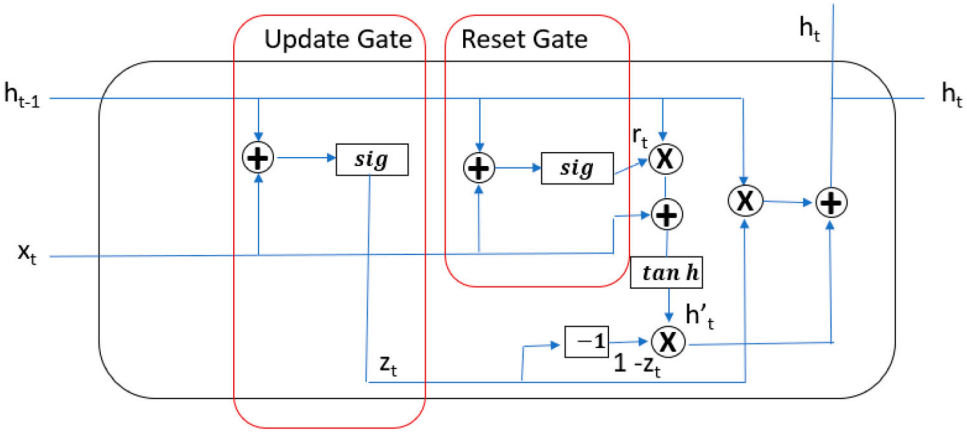

Figure 12. Gated Recurrent Unit (GRU) decoder for caption generation.

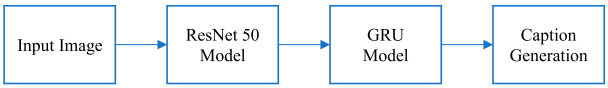

Figure 13. Image captioning – CNN (ResNet50) for feature extraction of images with simple GRU for text/caption generation.

tion. Here is the workflow of CNN with simple GRU, given in Figure 13.

The image is given to the CNN model to extract the feature vectors, and these features are given to the GRU model to generate captions/text. The CNN (ResNet50) takes the feature vectors from the input image and sends to GRU model for caption creation. The GRU model learns to temporal correlations among image and words by processing the features in a sequential manner. The gates as conventional LSTMs, GRU remains simpler but other has an efficient for jobs involving creation of captions. The feature vectors from the CNN-provided image, the GRU-based system effectively creates captions.

3.2.5. Stacked GRU

A sort of recurrent neural network (RNN) architecture known as a Stacked GRU includes multiple GRU layers arranged on top of one another [40]. Since the GRU layers are stacked on top of each another, they learn more complex patterns and captures long-term dependencies of data. The stacked GRU layers are depicted in Figure 14.

Here is the workflow of CNN with Stacked GRU, given in Figure 15.

The image is given to the CNN model, which extracts the feature vectors, and these features are passed to the GRU layer. An additional GRU layer helps to learn more complex patterns to generate captions.

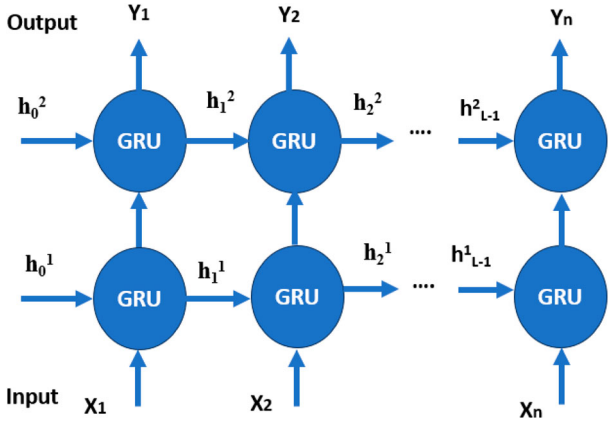

Figure 14. Stacked GRU decoder for caption generation.

3.2.6. Bidirectional GRU

Figure 16 illustrates the Bi-GRU structure.

A Bi-GRU is a language model that consists of two GRUs. One GRU layer is for forward processing, and another GRU layer is for backward processing. It takes both the past and future data [41].

Here is the workflow of CNN with Bidirectional GRU, given in Figure 17.

The image is passed to the CNN model, which extracts the feature vectors, and is passed to forward and backward GRU layers to preserve the future and past information and finally produce captions [42].

3.2.7. Greedy search

It is one of the most used NLP text decoding techniques for creating captions from output tokens. It always

Figure 15. Image captioning – ResNet50 for feature extraction with Stacked GRU for caption generation.

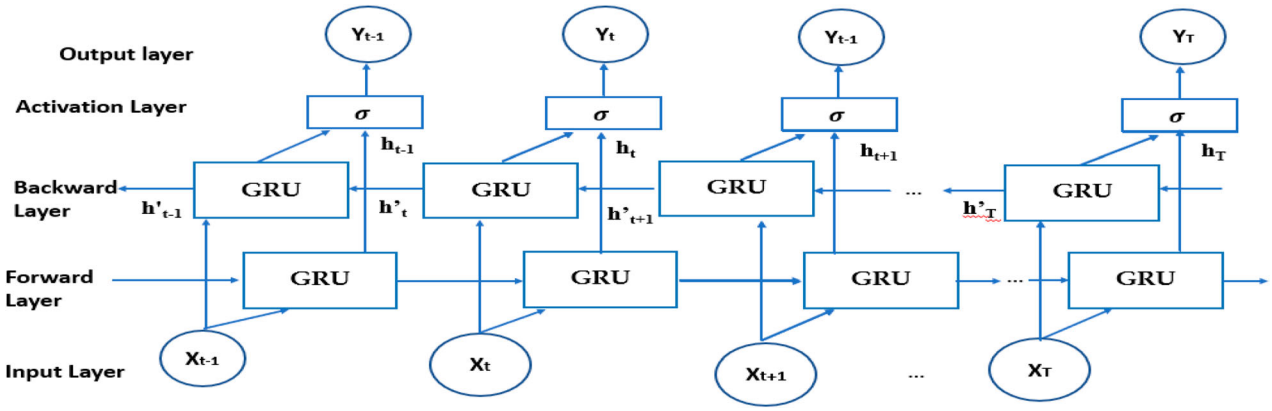

Figure 16. Bi-GRU decoder for caption generation.

Figure 17. Image captioning ResNet50 encoder for feature extraction with bidirectional GRU for caption generation.

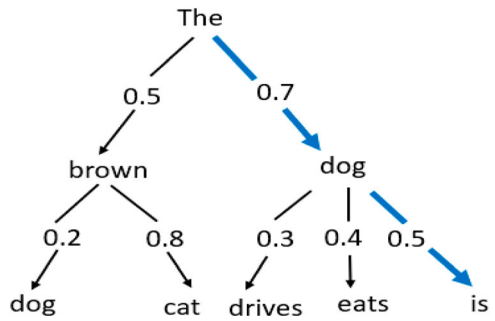

Figure 18. Greedy search method – example.

chooses the word with the highest likelihood [43]. The sample below demonstrates Greedy Search.

According to Figure 18

"The dog is," has a probability of $0.7 \times 0.5 = 0.35$

"The brown cat" has a probability of $0.5 \times 0.8 = 0.40$

As a result, Greedy search finds the line "The dog is" to have the highest likelihood. This decoding approach ignores high-probability words hidden underneath low probability ones. As a result, in the preceding example, Greedy searches for "The dog is" rather than "The brown cat". It is quick and consumes less resources.

The fundamental disadvantage of this Greedy search is that it does not provide optimal, high-probability claims. It leaves out high-probability words. Even if it begins with the most ideal word, it does not result in the most optimal phrase continuation.

The Greedy Search algorithm is given below with detailed steps:

1. Begin with an empty sequence, which represents the created caption
2. Change the current input to the special start token, < start >
3. Repeat steps until an end token (< end >) is generated or a maximum caption length is achieved
4. Use a language decoder model to anticipate the next word based on the present context and previously created words
5. Choose the word with the greatest probability for the prediction of the next word. At each phase, this is the best option available locally
6. Make the picked word the current input for the following iteration
7. Add the chosen word to the developing caption sequence
8. End the generating process when an end token is generated or the predefined maximum caption length is reached.

3.3. Beam search

Beam search is a popular approach for image captioning that produces more accurate and diversified captions. It addresses the Greedy Search problem by retaining the most likely hypotheses (beams) at each time step and selecting the hypothesis with the highest overall probability at the end. It is a search algorithm that analyses multiple paths in the search space and selects the most likely alternative using a score or probability.

Assume we are performing a beam search using two beams. On the first timestep, beam search would take

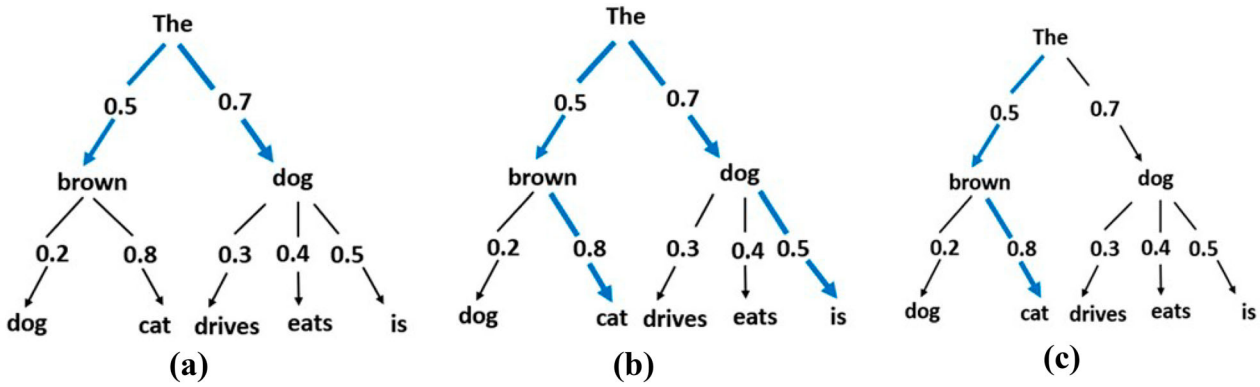

Figure 19. Beam Search – (a) Two Beams at first timestep, (b) Two Beams at second timestep, (c) Highlighting final prediction.

into account both the terms "brown" and "dog", as well as their probability.

The Beam search will examine all possible continuations of the two beams at the second timestep and select just the two with the highest likelihood, as illustrated in Figure 19(a). All possible continuations and their probabilities are presented.

The brown dog: $0.5 \times 0.2 = 0.1$

The brown cat: $0.5 \times 0.8 = 0.4$ (highest)

The dog drives: $0.7 \times 0.3 = 0.21$

The dog eats: $0.7 \times 0.4 = 0.28$

The dog is: $0.7 \times 0.5 = 0.35$ (second highest)

Thus the two beams at the next timestep will be "The brown cat" and "The dog is", which are shown in Figure 19(b). If we stop the beam search here, our anticipated outcome would be "The brown cat", which has the highest probability, shown in Figure 19(c). As the number of beams increases, it results in greater accuracy, but the computational complexity increases.

The algorithm for Beam Search is discussed in full below:

1. Begin the beam with a single hypothesis as the starting token
2. Create a set of next word predictions for each hypothesis in the current beam at time step "t". This can be performed by training the network using the current image and previous word embeddings as input.
3. Score each candidate's next word prediction for each hypothesis in the current beam. The score is determined as the sum of the candidate words' log probabilities up to time step "t".
4. Create "k" ideal hypotheses for each hypothesis in the current beam by picking the highest-scoring candidate words. The top "k" theories will combine to generate a new beam.
5. Repeat steps 2–4 until the required amount of time steps or an end token is generated.
6. At the end of the operation, choose the hypothesis with the highest score as the caption

4. Proposed image captioning model

The proposed work uses a ResNet50 cum hybrid LSTM–GRU approach to create captions from images. ResNet50 encoder extracts visual features, and a hybrid combination of the LSTM–GRU decoder creates image captions [44]. The novel model is used for solving the long-term memory problem of complex feature information. The proposed visual captioning model is depicted in Figure 20.

The proposed work is explained elaborately as follows:

The proposed work uses the CNN encoder to pass the input image to capture the features. The various transfer learning CNN models used are VGG16, InceptionV3, ResNet50 and DenseNet121. The Flickr8k dataset is trained using the aforementioned CNN models. The hyperparameters, such as loss and accuracy, are determined using epochs 40 and batch sizes 32 with Adam Optimizer. The last layer of these CNN models is skipped while training to extract the feature vectors from the given image. The embedding layer generates the text features [45]. The decoder LSTM model combines the output of the above two layers and uses a dense layer to make the caption prediction. The encoder–decoder approaches like VGG16 + LSTM, InceptionV3 + LSTM, ResNet50 + LSTM, and Dense Net121 + LSTM are implemented on the Flickr8K dataset. ResNet50 + LSTM produces good accuracy and low loss when compared to other models such as VGG16 + LSTM, InceptionV3 + LSTM and Dense Net121 + LSTM. Hence, ResNet50 is used as an encoder model because of its good accuracy and low loss, and variants of LSTM such as Stacked LSTM and Bi-LSTM are used as decoder models [46]. Furthermore, along with ResNet50 as an encoder, decoder models such as GRU and variants of GRU such as Stacked GRU and Bi-GRU are implemented to generate

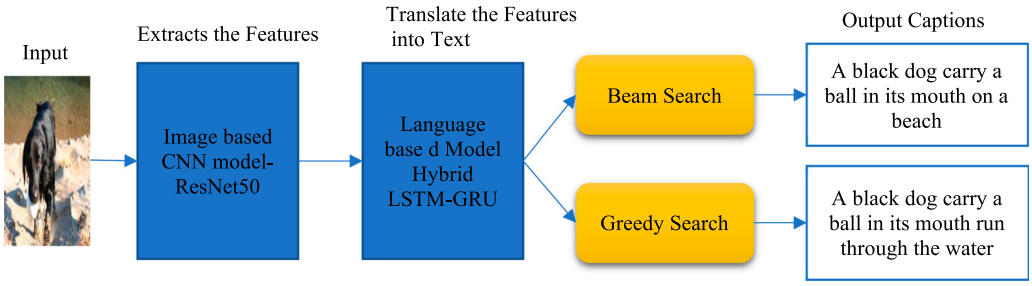

Figure 20. Image captioning ResNet50-Hybrid LSTM–GRU model with beam search.

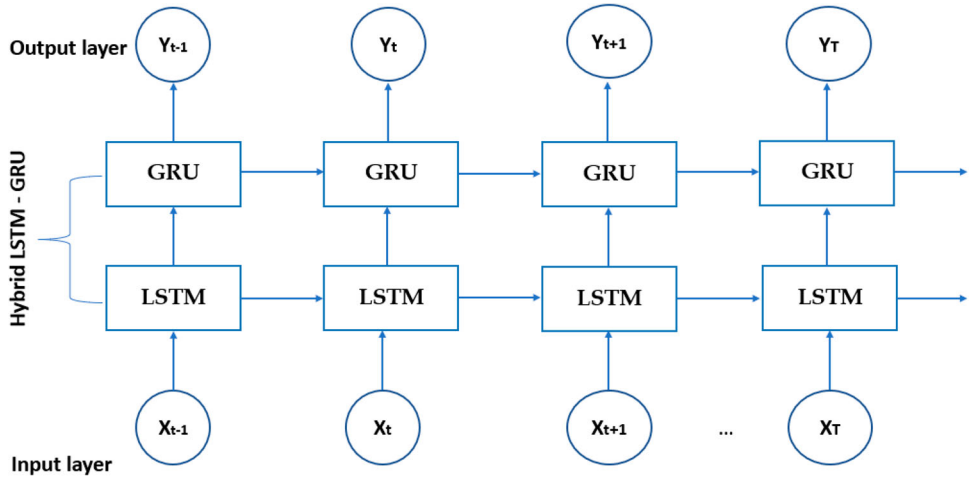

Figure 21. Proposed Hybrid LSTM–GRU model.

captions from the images. To further boost the quality of the caption, the ResNet50 is used as the encoder model, while the GRU model is stacked on top of the LSTM as the decoder model. Moreover, NLP decoding methods are used for the prediction of captions. The greedy search decoding algorithm generates the most likely word at each time step, and it relies on a fixed vocabulary of words. It does not capture a full range of vocabulary words, which may fail to capture important details. It does not consider the less likely word, which may be a better caption, and sometimes it generates repetitive words.

To overcome these limitations, Beam search is used for predicting the captions. It examines multiple paths in the search space and selects the most likely word based on a score or probability.

5. Hybrid LSTM–GRU model

A hybrid LSTM–GRU model combines both LSTM and GRU layers in a neural network architecture, which is depicted in Figure 21. The idea behind using such hybrid models is to benefit from the strengths of both LSTM and GRU while potentially mitigating their respective weaknesses.

Both LSTM and GRU are types of recurrent neural networks that tackle the vanishing gradient problem and capture long-term dependencies in the data. LSTM has three gating mechanisms (input, forget and output gates), while GRU has two gating mechanisms (reset and update gates) and is computationally less complex compared to LSTM [47].

In our suggested hybrid LSTM–GRU model, LSTM comprises three gates. They are Input (i_t), Forget (f_t) and Output (o_t). Gates store information in memory. Here, a sigmoid function is used, which takes the range of values from 0 to 1. If the gate value is one, data is stored in memory; otherwise, the data is deleted. Tanh function is used which covers the range of 1 to +1. The use of a second derivative protects information from fading.

GRU comprises two gates, they are Update gate (u_t) and Reset gate (r_g). The LSTM output is given to the GRU. x_t^i is the input feature set that contains the hidden state (h_t) of LSTM. In hybrid LSTM–GRU model [48], LSTM gates are described as using Equations (6)–(8)

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{6}$$

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{7}$$

$$O_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{8}$$

LSTM gate values are obtained by multiplying x_t and h_{t-1} by W_i (weight), and adding them to the bias b_i which is then passed to sigmoid function [49]. The input state of the cell C_t is defined using Equation (9)

$$C_t = tanh(W_c[h_{t-1}, x_t] + b_c) \tag{9}$$

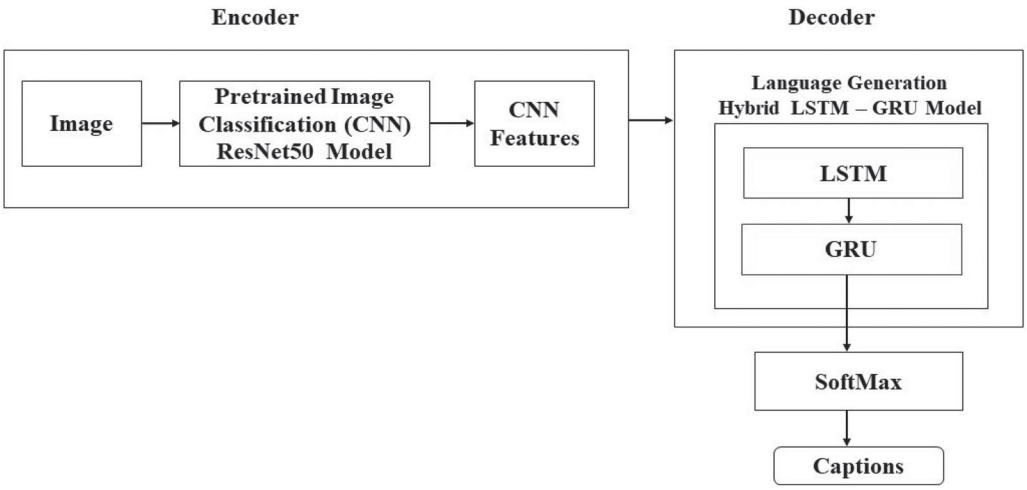

Figure 22. ResNet50 with Hybrid LSTM–GRU detailed view.

The output state of the cell C'_t is defined using Equation (10):

$$C'_t = f_t \times C'_{t-1} + i_t \times C_t \tag{10}$$

The equations above indicate that C'_t is sent into the GRU (z_t) layer, while z_t and h_{t-1} are multiplied by weight and sent to the reset gate (r_t) is computed using Equations (11) and (12).

$$z_t = \sigma\left(W_z[C'_t] + W_z[h_{t-1}]\right) \tag{11}$$

$$r_t = \sigma\left(W_r[C'_t] + W_r[h_{t-1}]\right) \tag{12}$$

The output layer attaches h'_t, which uses the tanh function to generate captions. It decides what information is to be kept.

$$h'_t = tanh(W_{C_t} + r_t \odot W_{C_t}[h_{t-1}]) \tag{13}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h'_t \tag{14}$$

$$h_t = O_t \times tanh(h_t) \tag{15}$$

In Equations (13)–(15), h'_t and h_t – Hidden units of the LSTM-GRU model W_i, W_f, W_o, and W_c – Weights of LSTM–GRU model b_i, b_f, b_o, and b_c – Biases of the LSTM-GRU model.

Here is the workflow of CNN (Res50) with LSTM–GRU, given in Figure 22.

The given image is fed to the CNN, which extracts the feature vectors, and these features are given to the hybrid combination. The output of LSTM is given to the GRU layer to generate captions [50].

6. Experimental results and discussion

6.1. Experimental setup

Our image captioning model uses Flickr8k dataset from Kaggle, which has 8092 images, each with five captions. For verification of the experimental work, 1000 images are taken; for testing, 1000 images are taken; and the balanced images are used for training. Images are pre-processed using VGG16 (224 × 224 pixels), InceptionV3 (299 × 299 pixels), Densenet121 (224 × 224 pixels) and Resnet50 (224 × 224 pixels). Using that encoder architecture, we extract features from images. Text is pre-processed by converting it to lowercase, removing punctuation and extracting unique words. The Adam optimizer is used for VGG16, InceptionV3, ResNet50 and DenseNet121. Decoders included language models such as LSTM, Stacked LSTM, Bi-LSTM, GRU, Stacked GRU and Bi-GRU, as well as a hybrid LSTM–GRU combination. In Google Colab, the encoder–decoder model is constructed with the Tensorflow framework and Python programming. The model training was conducted on an NVIDIA GPU to accelerate computations using 40 epochs with 32 batch sizes with a learning rate of 0.001 using Adam optimization. Our experiments' beam width was chosen by weighing model performance over the computational efficiency. The experiments with different beam widths (such as 3, 5 and 10) found accuracy minimized after the width of 5. The ideal beam width for our model's performance is evaluated to be 5 after weighing the trade-off among the execution time and output quality. The performance metric BLEU is employed to assess our suggested captioning model.

Table 1. Performance (accuracy and loss) of various CNN with LSTM.

Encoder–decoder model	Epochs	Loss	Accuracy
ResNet50 + LSTM	40	0.5959	0.8459
DenseNet121 + LSTM	40	0.6889	0.8329
InceptionV3 + LSTM	40	0.7212	0.8284
VGG16 + LSTM	40	0.7484	0.8222

6.2. Transfer learning-based CNN encoder with LSTM decoder

The results of various CNN encoder models with LSTM are listed in Table 1.

Table 1 shows the loss and accuracy for CNN encoder models with an LSTM of 40 epochs. As per

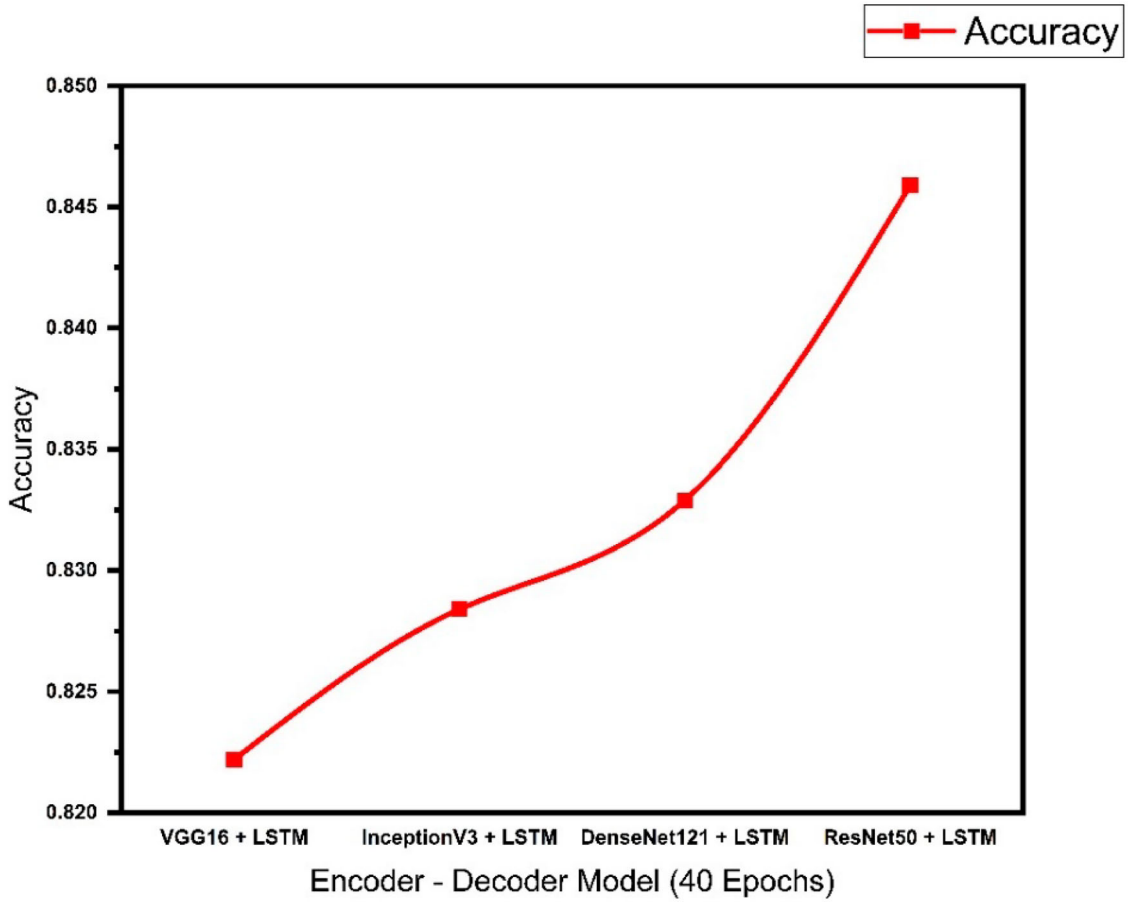

Figure 23. Accuracy graph for VGG16, InceptionV3, DenseNet121, ResNet50.

the results of Table 1, the ResNet50 + LSTM model produces a higher accuracy of 0.8459 than other encoder–decoder models. Also, the ResNet50 + LSTM model produces a low loss value of 0.5959 compared to other encoder–decoder models.

Figure 23 depicts the accuracy of various encoder–decoder models, such as VGG16 + LSTM, InceptionV3 + LSTM, DenseNet121 + LSTM, and ResNet 50 + LSTM. The graph clearly shows that ResNet50 + LSTM yields a higher accuracy of 0.8459.

Figure 24 depicts the loss of various encoder–decoder models, such as VGG16 + LSTM, InceptionV3 + LSTM, DenseNet121 + LSTM and ResNet 50 + LSTM. It is observed from Figure 24 that ResNet50 + LSTM yields a low loss value of 0.5959.

From Figures 23 and 24, ResNet50 with LSTM yields better accuracy and loss value than the other encoder–decoder models mentioned. Hence, ResNet50 is used as an encoder model, and with that, decoder models like Stacked LSTM, Bi-LSTM, GRU, Stacked GRU, Bi-GRU, and the proposed model, hybrid LSTM–GRU, are applied to the Flickr8K dataset. The performance of these models with respect to loss and accuracy at 40 epochs is given in Table 2.

Figure 25 depicts the performance metric of ResNet 50 with hybrid LSTM–GRU, with other encoder–decoder models. It is observed from the given below figure that ResNet50 + Hybrid LSTM–GRU yields a low loss value of 0.4013 and an accuracy of 0.8932.

Table 2. Performance (accuracy and loss) of ResNet50 encoder model with variants of decoder model.

Encoder–decoder model	Epochs	Loss	Accuracy
ResNet50 + LSTM	40	0.5959	0.8459
ResNet50 + Stacked LSTM	40	0.4185	0.8832
ResNet50 + Bi-LSTM	40	0.5483	0.8554
ResNet50 + GRU	40	0.5255	0.8594
ResNet50 + Stacked GRU	40	0.4555	0.8722
ResNet50 + Bi-GRU	40	0.5076	0.8632
ResNet50 + Hybrid LSTM–GRU (Our model)	40	0.4013	0.8932

6.3. *Comparative experimental analysis of different image captioning methods*

The table below compares the maximum BLEU Score values of ResNet50 + Hybrid LSTM–GRU to different encoder–decoder models employing greedy search and beam search, beam size k = 3, on 40 example images.

Figure 26 depicts the caption results of the proposed model, ResNet50 + Hybrid LSTM–GRU. The following are the reference captions and the anticipated caption for the image.

Semantic Propositional Image Caption Evaluation (SPICE) used to compare the scene graphs that depict objects, relationships and properties to evaluate

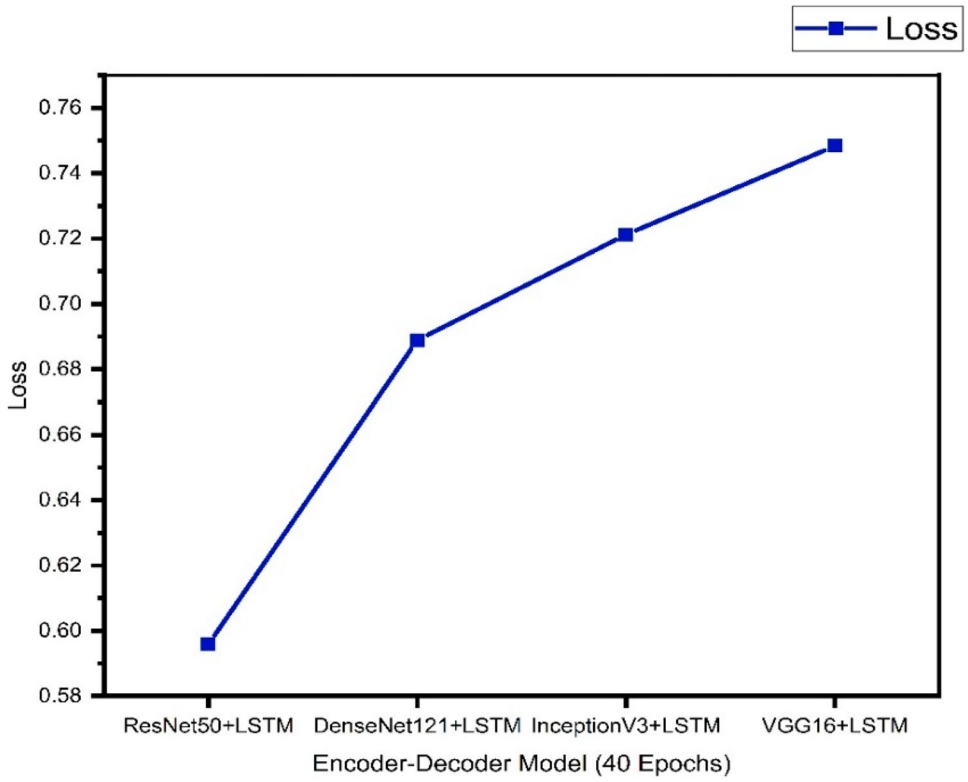

Figure 24. Loss graph for VGG16, InceptionV3, DenseNet121, ResNet50.

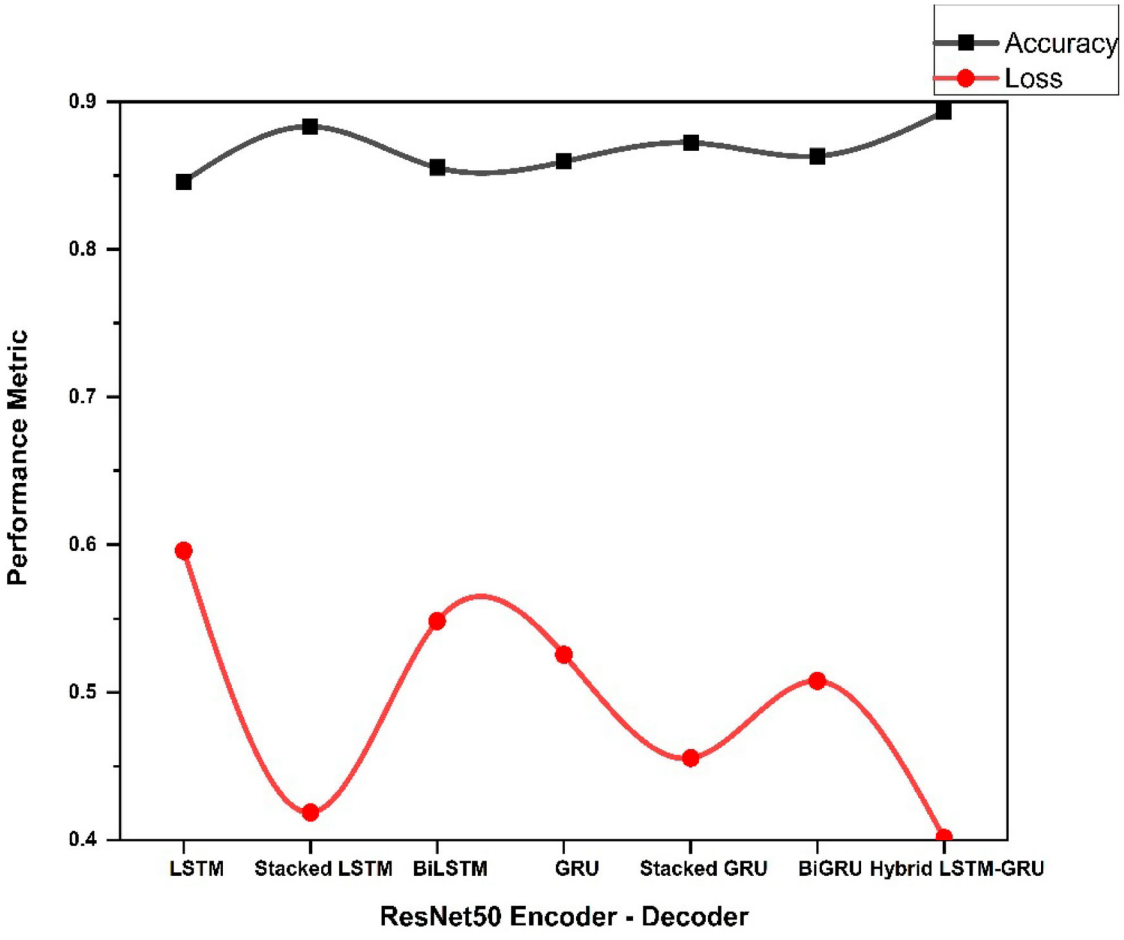

Figure 25. Performance of proposed (ResNet50 + Hybrid LSTM–GRU) model.

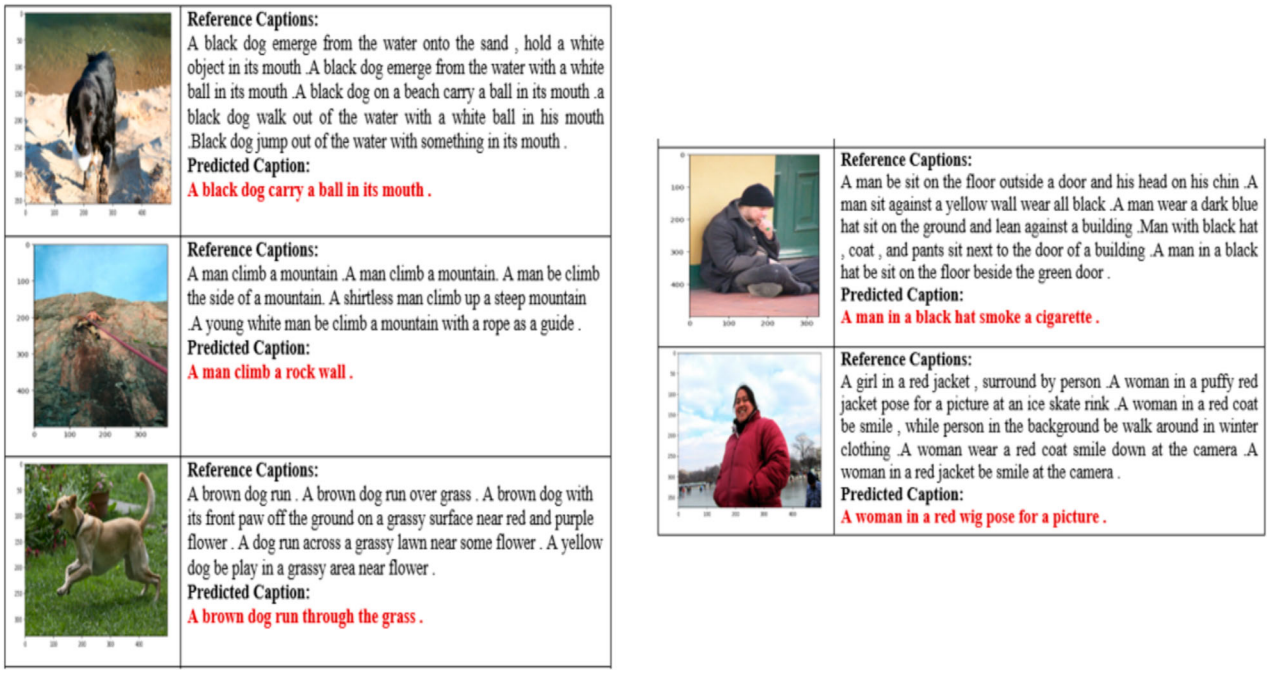

Figure 26. Image captioning results with reference captions and predicted caption.

Table 3. Comparison of maximum BLEU score on different image captioning methods with Greedy and Beam search.

	BLEU-1 score	
Encoder–decoder model	Greedy search	Beam search
ResNet50 + LSTM	0.2698	0.4286
ResNet50 + Stacked LSTM	0.4347	0.5623
ResNet50 + Bi-LSTM	0.3327	0.4785
ResNet50 + GRU	0.3986	0.4935
ResNet50 + Stacked GRU	0.4347	0.5633
ResNet50 + Bi-GRU	0.4286	0.5820
ResNet50 + Hybrid LSTM-GRU (Our Model)	0.5142	0.6034

Table 4. Performance analysis of various methods.

Model	Meteor	Spice
ResNet50 + LSTM	0.1573	0.1059
DenseNet121 + LSTM	0.2684	0.2089
M^2-Transf	0.2941	0.2212
ResNet50 + Hybrid LSTM–GRU (Our model)	0.3124	0.2404

the image descriptions. The precision (P) and recall (R) are considered to evaluate the SPICE using Equation (16):

$$SPICE(c, S) = \frac{2.P(c, S).R(c, S)}{P(c, S) + R(c, S)} \tag{16}$$

METEOR maps words between candidate and reference captions, calculates precision (P) and recall (R), and applies a weighted harmonic mean. It is computed using Equation (17)

$$F_{mean} = \frac{P.R}{\alpha.P + (1 - \alpha)R} \tag{17}$$

A penalty $Pen = \gamma \cdot frag\beta$ adjusts for fragmentation, it is defined using Equation (18)

$$METEOR = (1 - Pen) \cdot Fmean \tag{18}$$

Parameters α, β, γ are tuned for human judgment correlation.

Table 4 shows the performance analysis study, the M2-Transformer [51] has better capturing syntactic and semantic links in the image captions, with the highest scores for METEOR 0.2941 and SPICE 0.2212. Additionally, the DenseNet121 + LSTM has significant performance in the feature extraction and sequence modelling, with METEOR 0.2684 and SPICE value of 0.2089. Despite its advancements, the ResNet50 + LSTM illustrates the moderate performance with the lessscores of METEOR: 0.1573, SPICE: 0.1059. The proposed ResNet50 + Hybrid LSTM–GRU model has highest score of METEOR: 0.3124, SPICE: 0.2404, showcases the additional tuning or architectural changes are required to enhance its captioning performance.

The existing method has several drawbacks, such as difficulties with scalability while using the larger datasets needs a processing power, as MSCOCO and Flickr30k. More complicated image attributes are challenged for the hybrid ResNet50 + LSTM–GRU model to maintain well that limits generalizability. Furthermore, real-time applications become less feasible because of the computational difficulty of integrating the LSTM and GRU units that might increase the training times. To address the limitations, future research should optimize the architecture and investigate the lightweight models or feature extraction methods.

7. Conclusion and future scope

The goal of this work is to create an image captioning model using deep learning techniques. It uses encoders for the extraction of image features and decoders for the generation of captions. The encoder uses a transfer learning-based CNN model for image feature extraction. The decoder uses LSTM for text generation. The performance of various CNN models, such as VGG16, InceptionV3, ResNet50 and DenseNet121, with LSTM is discussed in this study. It is observed that while training these models for 40 epochs at 32 batch sizes, there is an extensive variation in terms of hyperparameters such as loss and accuracy. The results depict that ResNet50 + LSTM performs better with an accuracy of 0.8459 and a low loss of 0.5959 than the other models on the Flickr8k dataset. In this ResNet50 model, variants of LSTM, such as Stacked LSTM, Bi-LSTM and variants of GRU such as Stacked GRU, Bi-GRU are used as decoders. ResNet50 + Stacked LSTM provides an accuracy of 0.8832 and a loss of 0.4185. ResNet50+ Bi-LSTM provides an accuracy of 0.8554 and a loss of 0.5483. ResNet50 + Stacked GRU provides an accuracy of 0.8722 and a loss of 0.4555. ResNet50 + Bi-GRU provides an accuracy of 0.8632 and a loss of 0.5076. Furthermore, to enhance the caption quality, ResNet50, an encoder model, and a hybrid combination of LSTM and GRU, a decoder model, are used. This model produced an accuracy of 0.8932 and a loss of 0.4013.

Finally, two popular decodings, such as greedy search and beam search, are used along with the ResNet50 LSTM–GRU model to generate meaningful descriptions. The results of the proposed hybrid model yield accurate, meaningful captions and provide a better BLEU score of 0.6034 than compared to the results of a greedy search. The accuracy of captions impacted by the complex image elements that makes a harder to capture the subtle representations. Furthermore, the biases introduced by the Beam Search algorithm that ignores diverse or contextually richer outputs in favour of higher-probability sequences. It is intended to apply transformer-based models for image captioning to increase caption quality, which should be done in the future depending on the findings obtained. Moreover, transformer-based encoder models may be used for extracting features from images, and transformer-based decoder models may be used for text generation.

Acknowledgement

The authors wish to thank the Management, Sri Ramakrishna Engineering College, Coimbatore for providing the support to carry out the research activity.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

[1] Hossain MZ, Sohel F, Shiratuddin MF, et al. A comprehensive survey of deep learning for image captioning. ACM Comput Surv (CsUR). 2019;51(6): 1–36.

[2] Farhadi A, Hejrati M, Sadeghi MA, et al. Every picture tells a story: generating sentences from images. Lect Notes Comput Sci. 2010;6314 LNCS(PART 4):15–29. doi:10.1007/978-3-642-15561-1_2

[3] Hodosh M, Young P, Hockenmaier J. Framing image description as a ranking task: data, models and evaluation metrics, 2013.

[4] Vinyals O, Toshev A, Bengio S, et al. Show and tell: lessons learned from the 2015 MSCOCO image captioning challenge. IEEE Trans Pattern Anal Mach Intell. 2017;39(4):652–663. doi:10.1109/TPAMI.2016.2587640

[5] Mao J, Xu W, Yang Y, et al. Explain images with multimodal recurrent neural networks [Online]. 2014 Oct. Available from: http://arxiv.org/abs/1410.1090.

[6] Vinyals O, Toshev A, Bengio S, et al. Show and tell: a neural image caption generator [Online]. 2014 Nov. Available from: http://arxiv.org/abs/1411.4555.

[7] Khaing PP, Yu MT. Two-tier LSTM model for image caption generation. Int J Intell Eng Syst. 2021;14(4):22–34. doi:10.22266/ijies2021.0831.03

[8] Jia X, Gavves E, Fernando B, et al. Guiding long-short term memory for image caption generation [Online]. 2015 Sep. Available from: http://arxiv.org/abs/1509.04942.

[9] Wang C, Yang H, Bartz C, et al. Image captioning with deep bidirectional LSTMs. MM 2016 – Proceedings of the 2016 ACM Multimedia Conference, Association for Computing Machinery, Inc; 2016. p. 988–997. doi:10.1145/2964284.2964299

[10] Yao T, Pan Y, Li Y, et al. Incorporating copying mechanism in image captioning for learning novel objects [Online]. 2017 Aug. Available from: http://arxiv.org/abs/1708.05271.

[11] Alam S, Rahman S, Mubin KA, et al. Comparison of different CNN model used as encoders for image captioning.

[12] Rahman R, Murad H, Rahman NN, et al. CapNet: an encoder–decoder based neural network model for automatic bangla image caption generation [Online]. Available from: https://data.mendeley.com/datasets/rxxch9vw59/2.

[13] Yucong Q, Li M. Image caption based on bigru and attention hybrid model. ACM International Conference Proceeding Series, Association for Computing Machinery; 2021. p. 128–136. doi:10.1145/3488933.3488978

[14] Verma A, Yadav AK, Kumar M, et al. Automatic Image caption generation using deep learning. 2022. doi:10.21203/rs.3.rs-1282936/v1.

[15] Chu Y, Yue X, Yu L, et al. Automatic image captioning based on ResNet50 and LSTM with soft attention. Wirel Commun Mob Comput. 2020;2020(1):8909458. doi:10.1155/2020/8909458

[16] Zhang X, He S, Song X, et al. Image captioning via semantic element embedding. Neurocomputing. 2020;395:212–221. doi:10.1016/j.neucom.2018.02.112

[17] Chang YH, Chen YJ, Huang RH, et al. Enhanced image captioning with color recognition using deep learning methods. Appl Sci. 2022;12(1):209. doi:10.3390/app12010209

[18] Zhang J, Li K, Wang Z, et al. Visual enhanced gLSTM for image captioning. Expert Syst Appl. 2021;184:115462. doi:10.1016/j.eswa.2021.115462
[19] Ding S, Qu S, Xi Y, et al. Image caption generation with high-level image features. Pattern Recognit Lett. 2019;123:89–95. doi:10.1016/j.patrec.2019.03.021
[20] Al-Malla MA, Jafar A, Ghneim N. Image captioning model using attention and object features to mimic human image understanding. J Big Data. 2022;9(1):20. doi:10.1186/s40537-022-00571-w
[21] Tiwary T, Mahapatra RP. An accurate generation of image captions for blind people using extended convolutional atom neural network. Multimed Tools Appl. 2023;82(3):3801–3830. doi:10.1007/s11042-022-13443-5
[22] Azhar Y, Anugerah MR, Fahlopy MAR, et al. Image captioning using hybrid of VGG16 and bidirectional LSTM model. Kinetik: Game Technology. Information System, Computer Network, Computing, Electronics, and Control; 2022. doi:10.22219/kinetik.v7i4.1568
[23] Pal A, Kar S, Taneja A, et al. Image captioning and comparison of different encoders. J Phys: Conf Ser, Inst Phys Publ. 2020;1478(1):012004. doi:10.1088/1742-6596/1478/1/012004
[24] Veena S, Ashwin KS, Gupta P. Comparison of various CNN encoders for image captioning. J Phys: Conf Ser, Inst Phys. 2022;2335(1):012029. doi:10.1088/1742-6596/2335/1/012029
[25] Pradeep J, Raja Ratna S, Dhal PK, et al. Deepfore: a deep reinforcement learning approach for power forecasting in renewable energy systems. Electr Power Compon Syst. 2024:1–17. doi:10.1080/15325008.2024.2332391
[26] Singh S, Subburaj V, Sivakumar K, et al. Optimum power forecasting technique for hybrid renewable energy systems using deep learning. Electr Power Compon Syst. 2024:1–18.
[27] Chandrika VS, Kumar NMG, Kamesh VV, et al. Advanced LSTM-based time series forecasting for enhanced energy consumption management in electric power systems. Int J Renew Energy Res. 2024;14(1):127–139.
[28] Albelwi SA. Deep architecture based on DenseNet-121 model for weather image recognition [Online]. Available from: www.ijacsa.thesai.org.
[29] Soh M. Learning CNN-LSTM architectures for image caption generation.
[30] Rajaram A, Padmavathi K, Ch SK, et al. Enhancing energy forecasting in combined cycle power plants using a hybrid ConvLSTM and FC neural network model. Int J Renew Energy Res. 2024;14(1):111–126.
[31] Saravanan A, Farook S, Kathir I, et al. Adaptive solar power generation forecasting using enhanced neural network with weather modulation. Int J Renew Energy Res (IJRER). 2024;14(2):275–292.
[32] Gu J, Cai J, Wang G, et al. Stack-captioning: coarse-to-fine learning for image captioning [Online]. 2017 Sep. Available from: http://arxiv.org/abs/1709.03376.
[33] Shinde SK, Tirlangi S, Devaraj V, et al. Enhancing wind power generation forecasting with advanced deep learning technique using wavelet-enhanced recurrent neural network and gated linear units. Int J Renew Energy Res. 2024;14(2):324–338.
[34] Sujeeth T, Ramesh C, Palwe S, et al. Adaptive solar power generation forecasting using enhanced neural network with weather modulation. J Intell Fuzzy Syst. 2024;46(4):10955–10968.
[35] Karthikeyan M, Colak I, Sagar Imambi S, et al. Advancing electric demand forecasting through the temporal fusion transformer model. J Intell Fuzzy Syst: 1–18. (Preprint).
[36] Xie T, Ding W, Zhang J, et al. Bi-LS-AttM: a bidirectional LSTM and attention mechanism model for improving image captioning. Appl Sci. 2023;13(13):7916. doi:10.3390/app13137916
[37] Pushpavalli M, Dhanya D, Kulkarni M, et al. Enhancing electrical power demand prediction using LSTM-based deep learning models for local energy communities. Electr Power Compon Syst. 2024:1–18. doi:10.1080/15325008.2024.2316246
[38] Geetha G, Kirthigadevi T, Ponsam GG, et al. Image captioning using deep convolutional neural networks (CNNs). J Phys: Conf Ser. 2020;1712(1):012015. doi:10.1088/1742-6596/1712/1/012015
[39] Karthik A, Patthi S, Maheswari BU, et al. Advancing idiopathic pulmonary fibrosis prognosis through integrated CNN-LSTM predictive modeling and uncertainty quantification. Biomed Signal Process Control. 2025;100:106811. doi:10.1016/j.bspc.2024.106811
[40] Uslu B, Çayli Ö, Kiliç V, et al. Resnet based deep gated recurrent unit for image captioning on smartphone. Eur J Sci Technol. 2022;(35):610–615. doi:10.31590/ejosat.1107035
[41] Soares LD, Franco EMC. BiGRU-CNN neural network applied to short-term electric load forecasting. Production. 2022;32:e20210087. doi:10.1590/0103-6513.20210087
[42] Yan C, Gong B, Wei Y, et al. Deep multi-view enhancement hashing for image retrieval. IEEE Trans Pattern Anal Mach Intell. 2021;43(4):1445–1451. doi:10.1109/TPAMI.2020.2975798
[43] Bhatnagar P. Enhancing image captioning with neural models.
[44] Yan C, Li Z, Zhang Y, et al. Depth image denoising using nuclear norm and learning graph model. ACM Trans Multim Comput Commun Appl. 2020;16(4):1–17. doi:10.1145/3404374
[45] Yan C, Hao Y, Li L, et al. Task-adaptive attention for image captioning. IEEE Trans Circuits Syst Video Technol. 2022;32(1):43–51. doi:10.1109/TCSVT.2021.3067449
[46] Yan C, Teng T, Liu Y, et al. Precise no-reference image quality evaluation based on distortion identification. ACM Trans Multim Comput Commun Appl. 2021;17(3s):1–21. doi:10.1145/3468872
[47] Yan C, Meng L, Li L, et al. Age-invariant face recognition by multi-feature fusionand decomposition with self-attention. ACM Trans Multim Comput Commun Appl. 2022;18(1s):1–18. doi:10.1145/3472810
[48] Zafar N, Haq IU, Chughtai JUR, et al. Applying Hybrid Lstm-Gru model based on heterogeneous data sources for traffic speed prediction in urban areas. Sensors. 2022;22(9): 3348. doi:10.3390/s22093348
[49] Chenggang Y, Yaoqi S, Hao Z, et al. Review of omnimedia content quality evaluation. J Signal Process 2022;38(6):1111–1143.
[50] Zhang Z, Li L, Cong G, et al. From speaker to dubber: movie dubbing with prosody and duration consistency learning. Proceedings of the 32nd ACM International Conference on Multimedia. 2024. p. 7523–7532.
[51] Cornia M, Stefanini M, Baraldi L, et al. Meshed-memory transformer for image captioning. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020. p. 10578–10587.