RESEARCH ARTICLE

A Novel Image Captioning Technique Using Deep Learning Methodology

Abdullah Khan[1] **and Jaswinder Singh**[1,*]

[1] Department of the AIML-CSE Apex Institute of Technology, Chandigarh University, Mohali, India

Abstract

The capacity of robots to produce captions for images independently is a big step forward in the field of artificial intelligence and language understanding. This paper looks at an advanced picture captioning system that uses deep learning techniques, notably convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to provide contextually appropriate and meaningful descriptions of visual content. The suggested technique extracts features using the DenseNet201 model, which allows for a more thorough and hierarchical comprehension of picture components. These collected characteristics are subsequently processed by a long short-term memory (LSTM) network, a specific RNN variation designed to capture sequential dependencies in language, resulting in captions that are coherent and fluent.The model is trained and assessed on the well-known Flickr8k dataset, attaining competitive performance as judged by BLEU score metrics and proving its capacity to provide humanlike descriptions. This use of CNNs and RNNs demonstrates the value of merging computer vision and natural language processing for automated caption development. This approach has the potential to be applied in a range of industries, including assistive technology for the visually impaired, automated content production for digital media, enhanced indexing and retrieval of multimedia assets, and improved human-computer interaction. Furthermore, advances in attention processes and transformer-based models offer opportunities to improve the accuracy and contextual relevance of picture captioning models. The study emphasizes machine-generated captions' larger implications for increasing accessibility, boosting searchability in large-scale databases, and enabling seamless AI-human cooperation in content interpretation and storytelling.

Keywords: convolutional neural networks (CNN), recurrent neural networks (RNN), deep learning, image captioning, LSTM, DenseNet201, attention mechanism, BLEU score, natural language processing (NLP), multimodal learning.

1 Introduction

Picture captioning, which describes the content of a picture, has gained popularity in recent years. This technology can be used in a variety of contexts, including modifying app recommendations, virtual assistants, picture indexing, and disabled support [1]. Recent advances in artificial intelligence have driven computers to unparalleled levels of visual awareness, allowing them to evaluate photos while also producing

Submitted: 11 March 2025
Accepted: 23 May 2025
Published: 01 August 2025

Vol. 1, **No.** 2, 2025.
10.62762/TMI.2025.886122

***Corresponding author:**
Jaswinder Singh
jassi724@gmail.com

Citation
Khan, A., & Singh, J. (2025). A Novel Image Captioning Technique Using Deep Learning Methodology. *ICCK Transactions on Machine Intelligence*, 1(2), 52–68.

coherent and detailed written captions. Among these improvements, picture captioning has arisen as a challenging but critical endeavor that connects Natural language processing and computer vision. Image captioning is the process of converting visual information into comprehensible spoken explanations, a task that necessitates the seamless integration of feature extraction and language modeling. The capacity to produce contextually appropriate and semantically rich captions has important implications for a variety of real-world applications, spanning from better organizing and retrieving digital content to increasing accessibility for those who are blind or visually challenged. Image captioning is to provide a statement that is both linguistically reasonable and semantically accurate to the image's content [2].

Our creative, continuously updated visual representation serves as a long-term memory for previously expressed ideas during sentence construction. This makes it possible for the network to automatically choose important ideas to convey that haven't been said before [4].

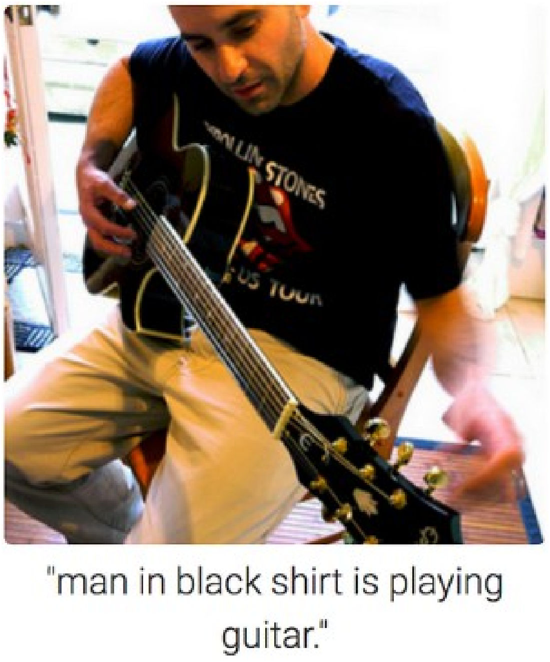

Figure 1. Image captioning examples.

Figure 1 illustrates representative examples of our model's captioning outputs, demonstrating its ability to generate syntactically correct and semantically relevant descriptions for images containing multiple objects and complex scenes.

Deep learning has transformed picture captioning, with convolutional neural networks (CNNs) acting as the foundation for visual feature extraction and recurrent neural networks (RNNs) enabling the sequential creation of descriptive text [5]. CNNs excel in learning visual hierarchies by capturing spatial patterns, object structures, and fine-grained features. DenseNet201, a cutting-edge CNN architecture is used in this work because of its capacity to optimize feature reuse via densely linked layers, increasing the richness of retrieved visual data. Unlike typical CNNs, which have diminishing gradients and redundant parameters, DenseNet201 enhances information flow and computational efficiency, making it ideal for complicated image processing workloads.

Following their retrieval, the traits need to be precisely converted into comprehensible language. RNNs, particularly long short-term memory (LSTM) networks, which are designed to replicate long-range associations in sequential data, are used to do this. In contrast to standard RNNs, which struggle with vanishing gradients in lengthy sequences, LSTMs preserve and update contextual information selectively over time using gated techniques. With this approach, the model can produce captions that accurately convey an image's information while maintaining linguistic coherence. The hybrid architecture of our model, as illustrated in Figure 2.

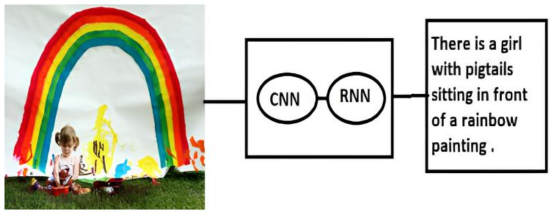
CNN
RNN
There is a girl with pigtails sitting in front of a rainbow painting .

Figure 2. The following diagram examples of CNN and RNN.

Flickr8k and Flickr30k datasets employ 1,000 photographs for validation and testing, with the remaining images used for training [2]. To assess the proposed model's performance, the Flickr8k dataset is used, which contains 8,000 photos associated with various human-annotated captions. We reserve MSCOCO captions dataset The remaining 98K photographs are utilized as training data, while 5K photos are used for validation and 5K images for further testing [3]. The dataset contains a diverse mix of products, situations, and activities, making it a great benchmark for assessing captioning algo-rithms. The model's performance is assessed using BLEU score metrics, which compare the linguistic correctness and fluency of output captions to human comments.

Aside from its technological contributions, automatic picture captioning has broad uses. It assists visually challenged people by providing verbal explanations of visual material. It allows for automated metadata development in media and content creation, which improves content classification and retrieval efficiency. Furthermore, captioning systems allow for more intuitive communication between users and AI-driven apps, opening the path for advances in multimodal AI.

Despite advances in this discipline, problems persist. Captions frequently need contextual knowledge, comprehension of complex concepts, and reasoning about the interconnections between things in a picture. Future initiatives include combining attention mechanisms with transformer-based designs, such as Vision Transformers (ViTs) and multimodal transformer models, to increase contextual understanding and captioning quality. This study advances the synergy between CNNs and NLP models, contributing to the continued development of AI systems capable of producing human-like descriptions with better precision and contextual awareness, as illustrated in Figure 3, which outlines the process of feature extraction.

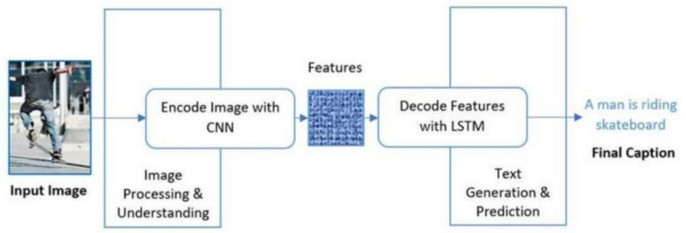
Input Image
Encode Image with CNN
Image Processing & Understanding
Features
Decode Features with LSTM
Text Generation & Prediction
A man is riding skateboard
Final Caption

Figure 3. Feature extraction diagram.

2 Existing System

Recent years have seen a significant shift in the development of photo captioning systems, with a range of methodologies and frameworks intended to close the gap between visual identification and language-creation [7]. Template-based and retrieval-based techniques were major components of early picture captioning models. Typically, template-based approaches produced strict, formulaic captions by using predetermined phrase structures that were filled with identified objects or characteristics. Conversely, retrieval-based algorithms would search a dataset for related photos and modify the captions for those images. Despite offering a basic comprehension, these early methods lacked the adaptability and contextawareness required to produce organic, evocative captions.

As deep learning progressed, attention turned to data-driven approaches that make use of neural networks to more efficiently complete picture captioning jobs [8]. The development of encoder-decoder architectures was a significant advancement that set the stage for the most advanced image captioning models available today. An encoder, usually a convolutional neural network (CNN), analyzes the input picture in this framework in order to extract useful information. A decoder, usually a recurrent neural network (RNN), receives these characteristics and uses the visual information from the encoder to produce descriptive text. Because the algorithms could now directly learn patterns from image-caption pairings in big datasets, this method enabled more fluid and contextually correct caption synthesis.

By adding attention techniques, which let the model to concentrate on particular areas of an image while it creates each word in the caption, contemporary image captioning models have improved the encoder-decoder structure even further [9]. By giving the network the ability to dynamically change its focus and mimic how people describe pictures by sequentially focusing on pertinent details, attention mechanisms overcome a major drawback in simple encoder-decoder models. Because models can now create thorough descriptions that capture both major and subtle components of a picture, this innovation has resulted in significant advances in the quality of generated captions.

The use of transfer learning has proven crucial in helping to enhance picture captioning systems [19]. Because they can capture high-level visual characteristics that have been learnt from large picture datasets like ImageNet, pretrained CNNs like VGG16, ResNet, and DenseNet are frequently used as encoders. The captioning model's performance is greatly enhanced by these pretrained networks, which provide it a strong base without requiring a lot of processing power to train from scratch. DenseNet201 has been a popular option in recent systems because of its dense connection design, which increases parameter efficiency and maximizes feature reuse.

As the decoder component has evolved, long short-term memory (LSTM) and gated recurrent unit (GRU) networks are increasingly popular choices for language synthesis tasks in picture captioning [18]. In particular, LSTMs have proven to be highly effective at preserving contextual coherence and managing long-term dependencies, which makes them ideal for producing accurate and linguistically fluid captions. Certain systems use generative adversarial networks (GANs) with reinforcement learning to improve captions and make sure they closely resemble human-like descriptions.

Various datasets, such Flickr8k, Flickr30k, and MS COCO, which offer a variety of image collections with annotated captions, are still used to assess existing systems [17]. These datasets provide a quantifiable indicator of caption quality and allow researchers to compare model performance using assessment criteria like as BLEU, METEOR, and CIDEr scores [20]. Despite significant advancements, there are still issues with current systems, such as creating captions for intricate scenes with several items, comprehending object connections, and creating captions with a variety of artistic elements.

A sequence-to-sequence model was proposed by Nag et al. [16], in which a succession of objects and their locations are encoded by an LSTM network, and the representation is decoded by an LSTM language model to provide captions.Their method improves caption accuracy by extracting item layouts (object categories and positions) from photos using the YOLO object detection paradigm. Additionally, they provide a version that uses the pre-trained VGG image classification model on ImageNet to obtain visual features. An object category (encoded as a one-hot vector) and a location configuration vector with the item's width, height, top-most position, and left-most position are sent to the encoder at each time step.It was all back to normal. The object detection model does not inherit the mistake, even when the model is trained via back-propagation. They demonstrated that the accuracy of their model was improved by using the CNN and YOLO modules. Not all of the information provided by YOLO's object properties, including object size and confidence, was used.

Using YOLO9000 and Faster R-CNN, Vo-Ho et al. [10] created an image captioning system that extracts object properties. An attention module handles each kind of feature, producing local features that reflect the model's current emphasis. The LSTM model calculates the probability of each word in the vocabulary set at each time step after concatenating the two local feature sets. The best possible caption is selected by analyzing the data using a beam search approach. They extracted features from photographs using the ResNet CNN. After extracting a list of tags from a picture using YOLO9000, they divided each tag into words and eliminated repeats to get a list of only unique words.They then insert each word in a ddimensional space using the word embedding approach. They generated language using LSTM units. Only the top twenty tags with the highest likelihood are retained.

In order to provide various descriptions, Cornia et al. [11] presented a novel framework for picture captioning that blends controllability with grounding. They employed a recurrent architecture to explicitly anticipate textual chunks based on regions while adhering to the limitations of the control when provided a control signal in the form of a series or a collection of picture regions. Flickr30k Entities and COCO Entities, a more sophisticated variant of COCO with semi-automated grounding annotation, are used for the experiments. According to their research, the method yields state-of-the-art results for customizable picture captioning in terms of caption quality and variety. In contrast to other attempts, we utilize every object characteristic that is accessible. The efficacy of this approach is demonstrated in the experiments section.

Verb-specific Semantic Roles (VSR) were presented by Chen et al. [12] as a novel CIC control signal. A verb and semantic roles that represent the actions and responsibilities of the entities involved make up VSR. In order to recognize and contextualize every aspect associated with every role in a VSR, they developed the Grounded Semantic Role Labeling (GSRL) paradigm. Then, in order to teach students descriptive semantics that is human-like, they introduced a Semantic Structure Planner. Lastly, they used a role-shifting paradigm to construct captions.

An inventive Anchor-Captioner method was presented by Wang et al. [15]. They started by determining which important tokens needed more focus and utilizing them as anchors. The associated anchor-centered graph (ACG) was created by joining the appropriate sentences for each anchor. Lastly, to increase the variety of generated captions, they employed multi-view caption production using several ACGs.

3 Methodology

Our approach aims to create descriptions for picture areas. During training, our model receives a set of photos and language descriptions [6].

This study uses a hybrid technique that combines convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to generate meaningful image descriptions.The model gathers information from input photos before producing captions using an LSTM-based sequence model. Data preparation, feature extraction, custom data production, model design, and assessment are all critical tasks.

We propose three unique strategies each technique focuses on a specific strategy used in computer vision and natural language processing (NLP) to interpret and describe visual input.

3.1 Nearest Neighbor-Based Approach

The leftmost image is the input image, which depicts a dog sitting at a table with a cake. The middle portion demonstrates a retrieval-based strategy in which the system compares the input image to a vast database of annotated images. This method employs an embedding space, in which pictures are represented as high-dimensional feature vectors. The system uses techniques like clustering or closest neighbor search to find the most comparable image.

The rightmost result is a caption that roughly fits the input image: "A puppy with a tie is sitting at a table with a cake." This approach obtains existing captions from related images rather than creating new ones. The description's correctness is determined by the dataset's quality and variety. It is computationally efficient but cannot generalize beyond previously classified pictures.

3.2 Template-Based Approach with Object Detection and Scene Graphs

The left picture focuses on identified items in the scene, such as the dog and cake. The center diagram depicts an organized approach that includes:

A Convolutional Neural Network (CNN) functions as an object detector, identifying essential things in the picture. A semantic relationship model examines the recognized objects, their features (such as colors and sizes), and their relationships [14]. This is sometimes arranged as a scene graph, with nodes representing items and edges representing connections (for example, "dog near cake").

The procedure removes components like these:

- **Objects (Obj)**: Identifying entities such as "dog" and "cake."
- **Attributes (Attr)**: Identifying qualities like "one dog" and "one cake."
- **Prepositions (Prep)**: Understanding spatial and contextual links (e.g., "on the table").

The rightmost output depicts the scenario in an organized way: "This snapshot depicts one dog and one cake. The dog is..."

Key insight: This strategy identifies items and their relationships before forming sentences. It provides more organized and factually precise captions but may lack fluidity and natural linguistic diversity. The description may be incomplete if an object is misidentified or relationships are improperly inferred.

3.3 A Deep Learning-Based Encoder-Decoder Approach

The left picture is unmodified and serves as the input image. The center graphic depicts an encoder-decoder system, a popular deep learning approach for picture captioning:

The Encoder (CNN) extracts high-level characteristics from images. These attributes are sent into the Decoder (LSTM/Transformer) [13], which produces a sequential caption word by word. The system learns to correlate visual elements with language representations in an end-to-end sequence.

The rightmost output is a dynamically created caption: "A dog is sitting on a couch with a toy." This technique does not require pre-existing captions or organized templates. Instead, it learns patterns and creates descriptions based on the training data.

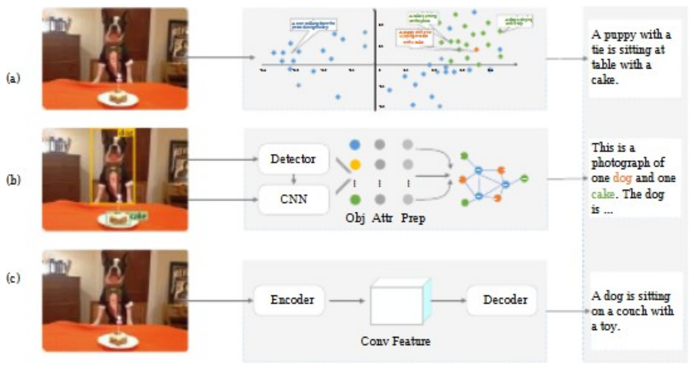
(a)
A puppy with a tie is sitting at table with a cake.
(b)
Detector
CNN
Obj
Attr
Prep
This is a photograph of one dog and one cake. The dog is ...
(c)
Encoder
Conv Feature
Decoder
A dog is sitting on a couch with a toy.

Figure 4. Three strategies for creating relevant textual descriptions.

It is capable of producing more natural, contextually rich, and diversified captions. However, the resulting caption may contain slight inconsistencies (for example, the model incorrectly recognizes the setting as a sofa rather than a table). Performance is determined by the quality of the dataset and the training method, as shown in Figure 4, which illustrates three strategies for creating relevant textual descriptions.

3.4 Deep Learning-Based Image Captioning Algorithm

- Step 1: Load and Preprocess the Dataset
  - Collect the image-caption dataset (such as Flickr8k or MS COCO).
  - Preprocess photos by resizing them to a certain form.
  - Convert photos to numerical arrays and normalize the pixel values.
  - Clean and preprocess text captions by eliminating special characters, lowercasing, and tokenizing.
  - Create a vocabulary with unique words from captions and assign integer indices.
  - Separate the dataset into training and validation sets.
- Step 2: Extract Image Features using CNN
  - Use a pre-trained CNN (such as VGG16 [1], InceptionV3, ResNet, or DenseNet201) as a feature extractor.
  - Remove the last classification layer to get feature embeddings.
  - Pass each image through the CNN and get the feature vectors from the final pooling layer.
  - Save extracted characteristics for efficient processing during training.
- Step 3: Encode Captions Using Word Embedding
  - Tokenize captions and transform words into integer sequences.
  - To guarantee that sequence lengths are uniform, use padding.
  - Use word embeddings (such as GloVe or trainable embeddings) to represent words in a dense vector space.
- Step 4: Design the Caption Generation Model
  - Define the encoder-decoder model:
    - The encoder (CNN) performs image processing and feature extraction.
    - The decoder (LSTM/GRU) creates captions consecutively.
  - Use a fully connected layer to map picture characteristics to the word embedding area.
  - To enhance sequential dependencies, use an LSTM network that includes attention techniques.
  - Use dropout and batch normalization to avoid overfitting.
- Step 5: Train the Model
  - Define the loss function (categorical cross-entropy in word prediction).
  - When updating gradients, use an optimizer like Adam.
  - Train the model on batches of image-caption pairings with instructor forcing.
  - Monitor validation loss and use early stopping to avoid overfitting.
  - Save the trained model and word-index mapping for further inference.
- Step 6: Create Captions for New Images
  - Load a test picture and process it similarly to the training photos.
  - Apply the learned CNN model to extract picture characteristics.
  - Use the learned LSTM model to create captions word for word.
  - Begin with a predetermined `<start>` token and anticipate the following word repeatedly.
  - Stop caption production whenever the `<end>` token is created or the maximum length is reached.
- Step 7: Evaluate Model Performance
  - Calculate the BLEU [2], METEOR [1], and ROUGE [6] scores for created captions.
  - Use n-gram matching to compare expected and ground truth captions.
  - Examine failure instances in which captions do not fit well with the image content.
- Step 8: Improve the Model with Fine-Tuning

– Implement Beam Search or Greedy Search to improve caption production.

– End-to-end training allows you to fine-tune both the CNN and the LSTM.

3.5 Image Processing

Table 1. Detailed of dataset.

Dataset name	Train	Validation	Test
Flickr8k	6000	1000	1000
Flickr30k	25700	3000	3000

As shown in Table 1, the dataset we are using here is the Flickr8k dataset, which is comprised of 8000 [16] images along with 5 captions for each image. This preparation stage includes scaling the photos to a common input size, standardizing pixel values, and enriching the dataset with techniques like rotation and flipping to increase its variety.The preprocessing steps include resizing, normalization, conversion to numerical format, and storage in a dictionary format.

To preserve uniformity across the dataset, all photos are downsized to a consistent dimension of 224 by 224 pixels [18]. After scaling, pixel values are min-max normalized to ensure they are inside the range [0,1]. The transformation follows this formula:

$$P_{\text{norm}} = \frac{PP_{\text{min}}}{P_{\text{max}}P_{\text{min}}} \tag{1}$$

where P denotes the original pixel value, and P_{min} and P_max signify the minimum and maximum pixel values, respectively. Normalization improves model training stability and efficiency by reducing numerical instability and speeding up convergence.

Non-alphabetical characters, such as punctuation marks and digits, are then removed to eliminate noise from the text data. Each caption is structured using predefined tokens: a start token (`<startseq>`) is added at the beginning of the phrase to signal the start of a caption, while an end token (`<endseq>`) is appended at the end to indicate the conclusion of the sequence. These tokens assist the model in recognizing sentence boundaries during training.

Tokenization plays a crucial role in converting textual data into a numerical format. Each unique word in the vocabulary is assigned a numerical identifier, translating captions into integer sequences. For example, if a caption contains ten words and the dataset vocabulary consists of 5,000 [6] unique words, the caption is encoded as a ten-element array, with each element representing the numerical index of the corresponding word in the vocabulary.

To ensure uniformity in sequence lengths, the longest caption in the dataset determines the maximum caption length (e.g., `max_length = 34 words`). Captions shorter than this limit are padded with special tokens to match the fixed sequence length, while longer captions are truncated accordingly. By completing these preprocessing steps, both image and caption data are structured in a manner that enhances model performance, allowing for efficient training and accurate caption generation. The use of dictionary-based storage ensures rapid retrieval, while text tokenization facilitates effective sequence modeling, thereby improving the overall efficiency of the image captioning system.

3.6 Feature Extraction with DenseNet201

We use DenseNet201, a pretrained Convolutional Neural Network (CNN), to improve computational efficiency and the model's understanding of visual material. This model was trained on a large-scale picture dataset, such as ImageNet, so it can recognize hierarchical patterns like edges, textures, objects, and high-level abstract representations. DenseNet201 uses its learnt representations to produce a meaningful summary of each image, collecting its key aspects in a condensed manner.

DenseNet201 processes each picture and derives high-level feature representations using its convolutional layers. This procedure can be mathematically stated as follows:

$$F_i = CNN(I_i). \tag{2}$$

where:

- F_i denotes the extracted feature vector of image I_i,
- CNN denotes the DenseNet201 model,
- I_i is the input image.

This feature extraction technique produces a *one-dimensional feature vector* of size (1,1920), where 1920 is the number of retrieved features. These extracted characteristics work as condensed representations of the image, storing visual patterns that help generate relevant captions.

By keeping the extracted feature vectors with the relevant photos, the model removes the need to reprocess images during each training cycle. This drastically lowers computing complexity and improves training efficiency. Instead of recalculating picture characteristics at each epoch, the model returns previously extracted features, allowing it to focus on learning the mapping between visual patterns and textual descriptions. This feature extraction method guarantees that the model efficiently captures crucial picture representations while minimizing training time, thereby enhancing the overall performance of the image captioning system.

3.7 Batch Data Handling with a Custom Data Generator

Handling huge datasets effectively is critical in deep learning, as loading all data at once is sometimes impossible owing to memory limitations. To solve this, a custom data generator is created, which allows for bulk processing of image-caption pairings. This method maximizes training efficiency by guaranteeing that data is loaded dynamically, lowering the risk of memory overflow while preserving model performance.

The custom data generator uses an organized process:

- **Batch Processing**: Rather of loading the complete dataset, the generator fetches smaller batches of image-caption pairings to ensure memory efficiency.
- **Randomized Data Shuffling**: To remove biases caused by fixed sequential learning, the generator randomly shuffles the dataset before each training period. This improves the model's generalization capabilities.
- **Mini-Batch Gradient Descent**: The model changes its parameters iteratively with mini-batches, resulting in more steady convergence and lower processing overhead.

Mathematical Representation of Mini-Batch Gradient Descent.

The weight update equation for each batch b of size m is as follows:

$$W = W\alpha \cdot \frac{1}{m}\sum_{i=1}^{m} \nabla L(W, x_i, y_i) \tag{3}$$

where:

- W represents the model.
- α is the learning rate that determines the step size during optimization.
- The gradient of the loss function in relation to the model parameters is represented by $\nabla L(W, x_i, y_i)$.
- The mini-batch has input-output pairs (x_i, y_i).
- The batch size m determines how many samples are utilized in each iteration.

This technique enables the model to gradually alter its parameters, exploiting several tiny updates rather than computing gradients over the whole dataset at once. As a result, training is more computationally viable and less likely to overfit.

By structuring data into batches and dynamically loading it during training, the custom data generator provides efficient memory management while increasing learning efficiency. This method dramatically improves model performance by eliminating memory overflow, ensuring computational stability, and allowing for smooth parameter updates during the training process, as illustrated in Figure 5, which shows the process of image captioning using a CNN-LSTM model.

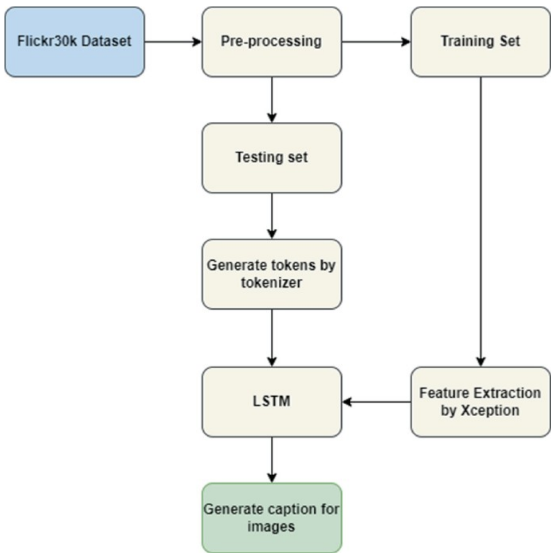
Flickr30k Dataset
Pre-processing
Training Set
Testing set
Generate tokens by tokenizer
LSTM
Feature Extraction by Xception
Generate caption for images

Figure 5. Image captioning using CNN-LSTM.

3.8 Model Architecture

The model has two essential components:

1. *Image Feature Extraction Module:*
   - DenseNet201 retrieves advanced visual characteristics.
   - These characteristics are compressed into a 256-dimensional vector using a fully

connected layer.

2. *Text Processing Module:*
   - A word embedding layer transforms tokenized words into 300-dimensional word vectors.
   - These vectors are fed into an LSTM network to detect long-term relationships in caption sequences.

3. *Mathematical Representation of LSTM:* At each time step t, an LSTM cell changes its hidden state by:

$$g_t = \sigma(U_g \cdot [s_{t-1}, r_t] + v_g) \tag{4}$$

$$j_t = \sigma(U_i \cdot [s_{t-1}, r_t] + v_i) \tag{5}$$

$$\tilde{D}_t = \tanh(U_D \cdot [s_{t-1}, r_t] + v_D) \tag{6}$$

$$D_t = g_t \cdot D_{t-1} + j_t \cdot \tilde{D}_t \tag{7}$$

$$k_t = \sigma(U_k \cdot [s_{t-1}, r_t] + v_k) \tag{8}$$

$$s_t = k_t \cdot \tanh(D_t) \tag{9}$$

where:

- g_t, j_t, k_t are the forget, input, and output gates, respectively.
- U_g, U_i, U_D, U_k are the weight matrices.
- v_g, v_i, v_D, v_k are the bias terms.
- σ is the sigmoid activation function.
- $\tanh$ is the hyperbolic tangent activation function.

Figure 6 shows a two-branch neural network, with one branch processing picture characteristics (CNN output) and the other processing text (embedded word sequences). These branches are eventually fused to provide meaningful captions.

The recovered picture characteristics are represented by input2 (InputLayer), which may be obtained from a pretrained CNN like DenseNet201. These characteristics are processed by a Dense layer (dense) to provide a lower-dimensional representation. The output is subsequently reshaped using a Reshape layer (reshape) to meet the LSTM's sequence processing needs.

Input3 (InputLayer) represents text input (partial caption sequences). These word indices are converted into continuous vector representations using an Embedding layer (embedding), which maps words into high-dimensional dense sets.

The outputs of the image feature extraction and text embedding branches are mixed using a Concatenate layer (concatenate). The fused representation is then sent into an LSTM layer (lstm) to capture sequential relationships within the caption. A dropout layer (dropout) is used to prevent overfitting. The Add layer (add) combines modified image features with LSTM output to enforce more complex feature interactions.

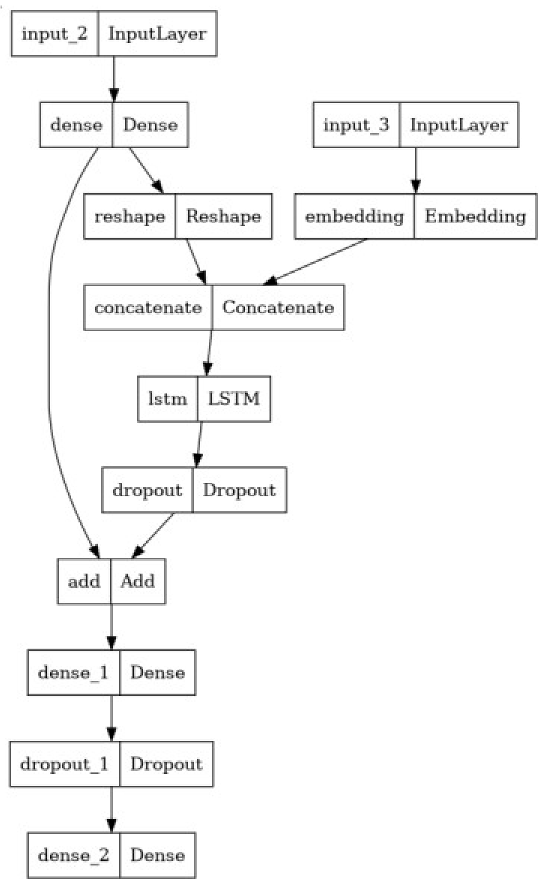

Figure 6. Image captioning model flowchat.

To create the final word predictions, the combined representation is sent via two Dense layers (dense1 and dense2) followed by an intermediate Dropout layer (dropout1). At each time step, the model builds a probability distribution over the vocabulary and predicts the following word in the sequence.

3.9 Image Feature Extraction using CNN

The image captioning model employs a Convolutional Neural Network (CNN) to extract rich feature representations from images, which are then utilized for generating meaningful captions. The input image X is represented as a three-dimensional tensor:

$$X \in \mathbb{R}^{M \times N \times D}$$

where M and N denote the height and width of the image, respectively, and D represents the number of color channels. A pretrained CNN model, such as DenseNet201, processes the image through multiple convolutional layers, extracting meaningful spatial and semantic features. The extracted feature map F is given by:

$$F = \text{CNN}(I), \quad F \in \mathbb{R}^d$$

where d represents the dimensionality of the feature vector. To reduce the dimensionality and adapt the features for sequential processing in an LSTM network, a fully connected Dense layer is applied:

$$F' = W_f F + b_f, \quad F' \in \mathbb{R}^{d'}$$

W_f and b_f are trainable weight and bias parameters, respectively, and d' represents the decreased feature dimension. To integrate image features into the sequential caption generation process, the reshaped representation is defined as:

$$F'' = \text{Reshape}(F'), \quad F'' \in \mathbb{R}^{T \times d'}$$

where T represents the number of time steps used in the sequence model.

The extracted and transformed feature representation is then fused with the textual input branch to generate meaningful captions.

Instead of using a single CNN model, multi-scale feature extraction may be accomplished by combining intermediate feature maps from multiple layers. This method allows the model to collect both fine-grained local features and high-level contextual information. Feature pyramids and hierarchical representations can increase caption quality by preserving low-level textures as well as high-level semantic notions.

Pre-trained CNNs like DenseNet201 and ResNet work well on generic pictures, however they may not generalize well to specialized domains. Fine-tuning these models using domain-specific datasets, such as medical pictures or satellite imagery, increases the relevance of the retrieved characteristics. Transfer learning and contrastive pretraining are two domain adaption strategies that can help enhance caption accuracy even more. Integrating attention processes into CNN-based feature extraction increases the model's capacity to focus on key visual areas. Spatial Attention allows the model to dynamically focus on key portions of a picture while generating captions. This guarantees that words in the caption relate to important items or aspects in the image. Channel Attention (SE-Net) assigns varying weights to feature channels, allowing the model to highlight important visual characteristics. This approach refines CNN-extracted characteristics and enhances contextual comprehension. Memory-Augmented Attention keeps contextual signals consistent across several pictures in a sequence, maintaining coherence when creating captions for video frames or related images.

Pooling approaches minimize computing complexity while keeping key visual features. Global Average Pooling (GAP) rather than employing completely linked layers, minimizes overfitting while retaining spatial information. Adaptive Pooling, unlike fixed-size max pooling, dynamically modifies feature dimensions before sending them to the LSTM. Dilated Convolutions enable wider receptive fields without increasing the number of parameters, resulting in better feature extraction for larger objects in pictures.

CNN-extracted feature maps can be improved by using scene graphs, which clearly depict objects and their interactions. This strategy enhances the model's capacity to provide structured, context-aware captions. By creating object interaction graphs, the captioning model can better grasp how items in an image interact, resulting in more detailed and meaningful captions.

Traditional CNN-LSTM designs use LSTMs for sequence modeling. Transformer-based encoders, on the other hand, may be used to process extracted feature maps, allowing for better long-range dependency modeling. Vision Transformers (ViTs), unlike CNNs, divide an image into patches and use self-attention to model global relationships, which improves contextual comprehension. Hybrid CNN/ViT Models combine the feature extraction power of CNNs and the sequence modeling strength of transformers.

Occlusions, bad lighting, and clutter in the backdrop can all have an impact on feature extraction. Robust CNN models can address these concerns by self-supervised learning, teaching the model to recreate missing portions of an image to improve feature extraction for occluded objects. Contrastive Learning trains the model to discriminate between multiple object representations, making it more resistant to noise and variance in visual input.

For real-time captioning in edge AI applications, CNN models must be optimized. Model compression reduces model size while maintaining accuracy. Quantization reduces computing precision (for example, utilizing 8-bit rather than 32-bit floating point) to allow for speedier inference. Knowledge distillation is the process of training a smaller model using knowledge from a bigger, more complicated model. Pruning eliminates superfluous parameters to speed up processing while maintaining model performance.

Let us assume:

The input picture has dimensions of $224 \times 224 \times 3$ (height, width, and channels). The CNN (DenseNet201) generates a feature map of $7 \times 7 \times 1024$. The feature map is converted to a 1D vector (1024×1024) and processed by a fully linked layer.

A convolutional layer uses a series of filters to extract patterns. Assume: Filter dimensions: $3 \times 3 \times 3$.

Stride: 2.

Padding: 1 and the number of filters is 1024. Using the output size formula for convolution:

$$Q = \frac{(MN + 2R)}{T} + 1 \tag{10}$$

where:

- Q = output size
- M = input size (224)
- N = kernel size (3)
- R = padding (1)
- T = stride (2)

$$Q = \frac{(2243 + 2(1))}{2} + 1 = \frac{224}{2} + 1 = 112 \tag{11}$$

The first convolutional layer generates a 112×112 feature map.

The final CNN feature map is $7 \times 7 \times 1024$ after many downsampling layers (pooling).

The retrieved features $F \in \mathbb{R}^{7\times7\times1024}$ are flattened:

$$F_{\text{flattened}} = 7 \times 7 \times 1024 = 50176 \tag{12}$$

A dense layer simplifies this to $d' = 1024$:

$$F' = W_f F + b_f, \quad F' \in \mathbb{R}^{1024} \tag{13}$$

Given a weight matrix W_f of size 1024×50176 and a bias vector b_f of size 1024, we may compute a single output neuron value:

$$F'_i = \sum_{j=1}^{50176} W_{f_{ij}} \cdot F_j + b_{f_i} \tag{14}$$

Learned parameters are represented by W_f and b_f. To fit the sequential model, we restructure the feature vector:

$$F'' = \text{Reshape}(F'), \quad F'' \in \mathbb{R}^{T\times d'} \tag{15}$$

where $T = 10$ (time steps for LSTM input), $d' = \frac{1024}{10} = 102.4 \approx 103$ (rounded).

The reshaped input F'' transforms into a 10×103 feature matrix, ready for sequence modeling.

4 Model Training

4.1 Loss Function

The categorical cross-entropy loss function is utilized for model training. It quantifies the difference between expected and actual word probabilities during training and is defined as:

$$J = \sum_{j=1}^{M} p_j \log(\hat{p}_j) \tag{4}$$

where

- M represents the vocabulary size.
- p_j represents the real probability of word j.
- $\hat{p}_j$ represents the anticipated probability of word j.

The loss function penalizes inaccurate predictions by assigning a higher loss when the predicted probability of the correct word is low.

4.2 Optimization Strategy

a) Adam Optimizer: Gradient-based optimization is performed using the Adaptive Moment Estimation (Adam) optimizer. The parameters' updating rules are as follows:

$$\boldsymbol{v}_t = \lambda_1 \boldsymbol{v}_{t-1} + (1\lambda_1)\boldsymbol{q}_t \tag{17}$$

$$\boldsymbol{s}_t = \lambda_2 \boldsymbol{s}_{t-1} + (1\lambda_2)\boldsymbol{q}_t^2 \tag{18}$$

$$\hat{\boldsymbol{v}}_t = \frac{\boldsymbol{v}_t}{1\lambda_1^t}, \quad \hat{\boldsymbol{s}}_t = \frac{\boldsymbol{s}_t}{1\lambda_2^t} \tag{19}$$

$$\boldsymbol{\vartheta}_t = \boldsymbol{\vartheta}_{t-1} \frac{\alpha \hat{\boldsymbol{v}}_t}{\sqrt{\hat{\boldsymbol{s}}_t} + \epsilon} \tag{20}$$

where:

- $\boldsymbol{q}_t$ is the gradient of the loss function at time t.
- $\boldsymbol{v}$ and $\boldsymbol{s}$ represent moving averages of the gradients.
- λ_1 and λ_2 are decay rates (default: 0.9, 0.999).
- ϵ is a small constant to prevent division by zero.
- α is the learning rate.

b) *Early Stopping:* To avoid overfitting, training is halted automatically if the validation loss does not improve after a certain number of epochs.

c) *Learning Rate Reduction on Plateau:* If validation accuracy plateaus, the learning rate is lowered by a factor (e.g., multiplying by 0.1) to fine-tune performance.

4.3 Model Architecture

a) *Image Feature Extraction:* As shown in Figure 7, the DenseNet201 convolutional neural network extracts image features, which are processed through a dense layer to obtain a 256-dimensional feature vector:

$$F_{\text{image}} = W_f \cdot \text{DenseNet201}(I) + b_f \tag{21}$$

where I is the input image, and W_f, b_f are the dense layer weights and biases.

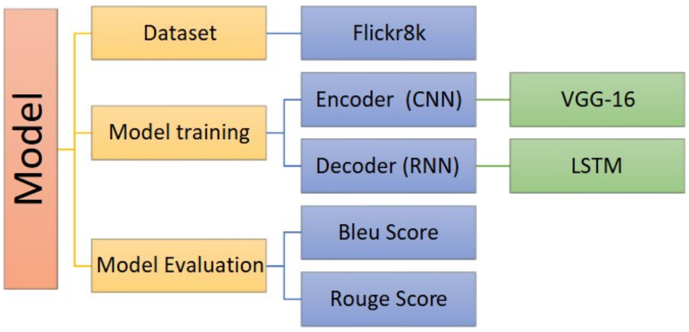

Figure 7. Image captioning model.

b) *Text Feature Extraction:* Captions are tokenized and converted to embeddings before being passed to an LSTM network:

$$G_{\text{seq}} = \text{LSTM}(D(v_t)) \tag{22}$$

where $D(v_t)$ represents the embedding of word v_t.

LSTM equations:

$$q_t = \sigma(U_q[z_{t-1}, y_t] + d_q) \tag{23}$$
$$r_t = \sigma(U_r[z_{t-1}, y_t] + d_r) \tag{24}$$
$$\tilde{S}_t = \tanh(U_s[z_{t-1}, y_t] + d_s) \tag{25}$$
$$S_t = q_t \cdot S_{t-1} + r_t \cdot \tilde{S}_t \tag{26}$$
$$s_t = \sigma(U_s[z_{t-1}, y_t] + d_s) \tag{27}$$
$$z_t = s_t \cdot \tanh(S_t) \tag{28}$$

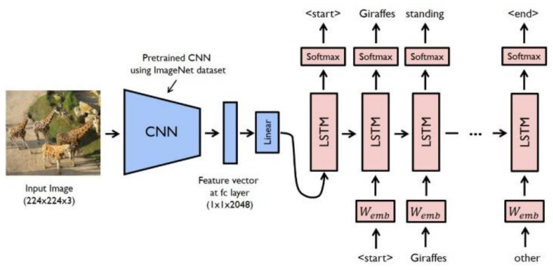

Figure 8. CNN-LSTM architecture.

As shown in Figure 8, Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) [2] specifically designed to process sequential data while addressing the vanishing gradient problem. In the context of image captioning, LSTMs analyze extracted text features, enabling the model to generate coherent and contextually appropriate captions. These networks employ a gating mechanism to control the flow of information, ensuring the retention of important long-term dependencies necessary for language modeling tasks. The main components of an LSTM cell include the forget gate, input gate, cell state update, and output gate.

The forget gate applies a sigmoid activation function to select whether bits of the prior cell state should be maintained or discarded.

$$q_t = \sigma(U_q[z_{t-1}, y_t] + d_q) \tag{29}$$

values close to zero in the forget gate's output indicate that components of the previous cell state are discarded, while values close to one indicate that components are retained.
The input gate consists of a sigmoid activation and a tanh activation, deciding which new information should be added to the cell state:

$$r_t = \sigma(U_r[z_{t-1}, y_t] + d_r) \tag{30}$$

$$\tilde{S}_t = \tanh(U_s[z_{t-1}, y_t] + d_s) \tag{31}$$

The cell state is updated based on the previous state and the candidate values:

$$S_t = q_t \cdot S_{t-1} + r_t \cdot \tilde{S}_t \tag{32}$$

allowing LSTMs to maintain long-term dependencies effectively.
The output gate regulates the next hidden state, which is passed to the subsequent LSTM cell and used for final predictions:

$$q_t = \sigma(V_q[r_{t-1}, x_t] + e_q) \tag{33}$$

$$r_t = q_t \cdot \tanh(T_t) \tag{34}$$

The hidden state r_t serves as the main output at each time step, influencing future predictions.

In image captioning, LSTMs play a crucial role in processing text sequences. The extracted image features serve as the initial input to the LSTM, followed by the word embeddings of the caption sequence. Each word in the caption is converted into a dense vector representation through an embedding layer, ensuring that words with similar meanings have closely related representations:

$$G_{\text{seq}} = \text{LSTM}(D(v_t)) \tag{35}$$

where $D(v_t)$ represents the embedding of word v_t.

The LSTM predicts the next word progressively using the context supplied by the previously produced words and the retrieved picture attributes.This process allows the model to generate fluent and contextually meaningful captions. By preserving long-term dependencies, LSTMs help maintain coherence in generated sentences, ensuring that each predicted word aligns appropriately with the image content and the prior words in the sequence. This ability to retain and process sequential information makes LSTMs highly effective in language modeling and caption generation tasks.

c) *Integrating Image and Text Features:* The image features F_{image} and text features F_{text} are concatenated:

$$F_{\text{combined}} = \text{Concatenate}(F_{\text{image}}, F_{\text{text}}) \tag{36}$$

d) *Final Caption Generation:* The final layer is a softmax classifier that predicts the next word:

$$P(w_t \mid w_{1:t-1}, I) = \frac{e^{W_o h_t}}{\sum_{j=1}^{N_w} e^{W_o h_j}} \tag{37}$$

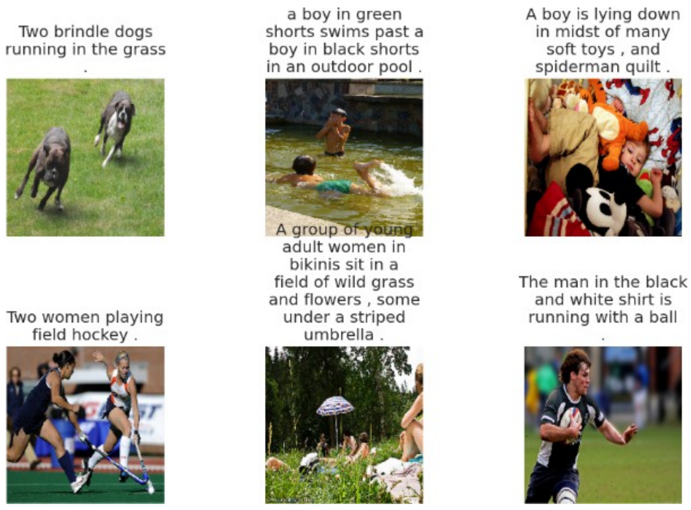
Two brindle dogs running in the grass .
a boy in green shorts swims past a boy in black shorts in an outdoor pool .
A boy is lying down in midst of many soft toys , and spiderman quilt .
Two women playing field hockey .
A group of young adult women in bikinis sit in a field of wild grass and flowers , some under a striped umbrella .
The man in the black and white shirt is running with a ball .

Figure 9. Outputs of our model.

where $P(w_t)$ represents the probability of the next word, h_t is the LSTM hidden state, and W_o is the weight matrix of the softmax layer.
The process continues until an "endseq" token is generated, marking the sentence completion. Figure 9 Shown the outputs of our model.

5 Results and Discussion

A comparison of the loss curves for two models is presented in Figures 10 and 11, which illustrate the training (loss/train) and validation (valloss/val) losses across multiple epochs. These curves provide insights into the models' learning effectiveness and generalization to new data.

Figure 10 covers 5 epochs, reflecting a shorter training duration, while Figure 11 spans 12 epochs, offering a more detailed view of the learning trajectory. The layout of Figure 10 is compact with a conventional aspect ratio, whereas Figure 11 adopts a larger format, enhancing readability and facilitating the analysis of long-term trends.

Both models exhibit overfitting, though with distinct patterns. In the first model, the validation loss starts modestly but rises after approximately 3 epochs, indicating early overfitting as the model struggles to generalize despite performing well on training data. In the second model, the validation loss stabilizes around epoch 5 and increases only marginally thereafter, suggesting a more consistent training process, though overfitting becomes evident after 6 epochs. Techniques such as dropout or L2 regularization could help mitigate these issues.

The visualization styles also differ in interpretability. The first model's plot uses a basic white background with standard tick marks, resulting in a straightforward but less visually engaging presentation. In contrast, the second model's plot employs a seaborn-style light blue/gray background, offering a cleaner and more professional appearance. Additionally, the first model labels its curves as "loss" (training) and "valloss" (validation), while the second uses "train" and "val," which are shorter and more intuitive.

Regarding loss trends, the first model begins with a high initial loss (5.25), which decreases rapidly but shows early overfitting as validation loss rises after epoch 3. The second model starts with a lower initial loss (4.25) and declines steadily over 12 epochs, with validation loss diverging slightly after epoch 6 but remaining relatively stable. The extended training period in the second model provides a clearer picture of convergence, though both models suggest the need for regularization, early stopping, or additional training data to address overfitting. The second model's broader format and seaborn design further enhance visual clarity, making loss patterns easier to discern.

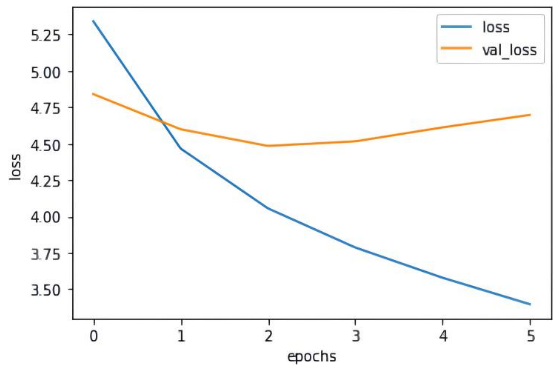

Figure 10. Existing Train-Val graph.

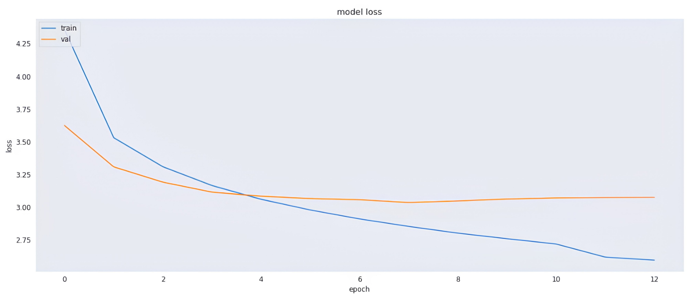

Figure 11. Our model Train-Val graph.

As shown in Table 2, the evaluation of the image captioning model employing DenseNet201 + LSTM on the Flickr8k dataset demonstrates its capability to generate relevant, human-like captions. The model's performance was assessed using BLEU scores, qualitative observations, and training performance analysis. The BLEU scores, as shown in Table 2, indicate that the model effectively captures key objects and actions in images. The higher BLEU-1 and BLEU-2 scores suggest that the model identifies relevant keywords and objects, whereas the relatively lower BLEU-3 and BLEU-4 scores imply a need for improvement in generating more complex sentence structures.

Table 2. BLEU score comparison on Flickr8k dataset.

Model	B-1	B-2	B-3	B-4
VGG16 + LSTM	0.47	0.28	0.19	0.08
InceptionV3 + LSTM	0.48	0.27	0.18	0.08
Multi-feature	0.50	0.30	0.20	0.09
Attention Model	0.51	0.31	0.22	0.10
DenseNet201 + LSTM	**0.52**	**0.32**	**0.23**	**0.11**

The DenseNet201-based model outperformed conventional CNN-based architectures such as VGG16 and InceptionV3, benefiting from the closely coupled layers improve feature reuse and gradient flow. The attention mechanism further contributed to generating contextually accurate captions. Qualitative analysis revealed that the model excels at recognizing objects and simple actions, generating fluent and coherent captions. However, challenges remain in handling complex scenes, fine-grained details, and contextual inference. While it accurately describes visible objects, it sometimes fails to infer abstract relationships or emotions, leading to more literal and less expressive captions.

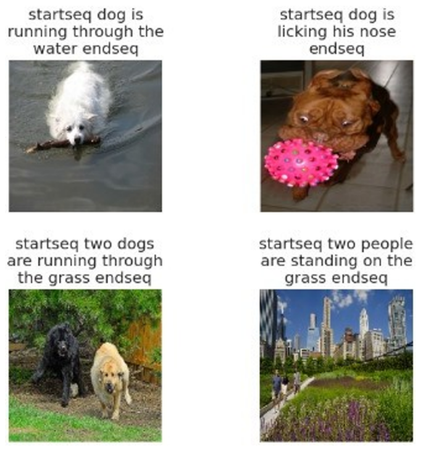
startseq dog is running through the water endseq
startseq dog is licking his nose endseq
startseq two dogs are running through the grass endseq
startseq two people are standing on the grass endseq

Figure 12. Final outputs Of image captioning model.

Training performance analysis showed stable convergence with decreasing training and validation loss, though minor overfitting was observed.

Techniques such as data augmentation and regularization can be further explored to enhance generalization. The model's ability to produce detailed descriptions reinforces its potential, yet incorporating external knowledge bases or advanced attention mechanisms could further refine its interpretative skills.

The DenseNet201 + LSTM-based image captioning model produces useful, human-like captions for pictures, outperforming typical CNN + LSTM designs. The BLEU score evaluation shows high performance in capturing keywords, objects, and actions, with BLEU-1 (0.52) and BLEU-2 (0.32) scores indicating the model's capacity to detect essential aspects inside a picture. However, poorer BLEU-3 (0.23) and BLEU-4 (0.11) results indicate the difficulty of creating complicated, multiword sequences that closely match human-annotated captions. Despite these problems, the model outperforms traditional techniques like as VGG16 + LSTM and InceptionV3 + LSTM, owing mostly to DenseNet201's fast feature propagation, gradient flow, and dense connections, which improve representation learning. Furthermore, the self-attention process increases contextual awareness, ensuring that generated captions are grammatically correct, fluid, and cohesive, as shown in Figure 12, which illustrates the final outputs of the image captioning model.

Qualitative examination reveals the model's capabilities in accurate object detection, as it excels at describing single-object images with distinct foregrounds, such as "a dog running in the park" or "a child playing with a toy." However, constraints emerge in complicated and abstract environments, where several things and activities coexist. The model struggles with fine-grained features, frequently forgetting minor characteristics such as facial expressions, emotions, or complex connections between objects. Furthermore, while it may accurately identify literal components of a picture, it falls short of capturing deeper semantic links, implicit meanings, and emotional context. A comparison with human-annotated captions demonstrates that human descriptions include purpose, subjectivity, and emotional depth, all of which pose challenges for the model.

Training and convergence analyses show that the model has robust learning patterns, with consistent reductions in training and validation loss over epochs. While the difference between training and validation loss is still small, mild overfitting tendencies indicate the need for data augmentation, dropout, and other regularization strategies. Learning rate scheduling has been shown to be useful in improving convergence while avoiding problems like vanishing gradients and excessive overfitting.

Overall, the DenseNet201 + LSTM model demonstrates a strong foundation for automated image captioning, with competitive performance across a wide range of picture categories. While it achieves high word-level accuracy and phrase coherence, further work is needed to improve its capacity to catch abstract linkages, contextual subtleties, and complicated scene dynamics.

6 Future Work

Several avenues for development in picture captioning models may be investigated, including architectural innovations, integration with external knowledge sources, multimodal learning, dataset extension, and real-time deployment optimization. The combination of transformer-based architectures, such as Vision Transformers (ViTs) with GPT-based models, can considerably improve semantic reasoning, long-range dependency modeling, and sentence coherence in produced captions. Selfsupervised learning approaches can increase generalization by training models on big datasets without the need for human annotations. Hybrid techniques that combine CNNs and transformers can preserve robust feature extraction capabilities while enhancing contextual understanding and sentence comprehension. Incorporating commonsense knowledge graphs (e.g., ConceptNet, ATOMIC) can aid models in inferring implicit links, such as object affordances ("A dog can eat cake" rather than "A dog is reading a book"). Fine-tuning models with external datasets that give cultural, social, and situational context can aid in context-aware caption production. Semantic role labeling (SRL) approaches can improve contextual inference by ensuring that captions include not just objects but also their functions, interactions, and intended behaviors.

Using adaptive attention processes, the model may dynamically focus on different areas of the image based on the produced words. Memory-augmented attention can assist maintain contextual signals across several pictures in a sequence, resulting in more cohesive captions for image series, video frames, or related images. Graph-based attention improves scene understanding by explicitly modeling item connections, spatial location, and interactions. In video captioning jobs, using audio cues, scene metadata, and temporal signals can increase understanding by providing more context to narrative descriptions. Large-scale vision-language pretraining models such as CLIP, ALIGN, Flamingo, and BLIP can improve caption quality by better aligning visual and linguistic embeddings. Zero-shot and few-shot learning strategies can help captioning algorithms generalize to previously unknown objects and settings with minimum labeled data.

One of the most difficult components of picture captioning is conveying an image's abstract, emotive, and creative elements. Future enhancements may include incorporating sentiment analysis and emotion recognition into captioning algorithms, allowing them to provide descriptions that reflect mood, tone, and subjective impression. Improving scene graph representations through improved relational reasoning can help models grasp deeper object interactions, allowing for more descriptive and meaningful captions. Models may also be

refined to provide innovative and stylish captions for use in digital art, narrative, and content creation. Extending datasets beyond established benchmarks like MS COCO by including datasets like Conceptual Captions, Open Images, and LAION5B can dramatically enhance generalizability. Addressing bias reduction through dataset balance and fairness-aware training algorithms can result in inclusive and impartial caption creation across diverse demographics and cultural situations. Active learning algorithms may be used to constantly enhance datasets by taking into account user comments and enhancing model flexibility over time.

To allow realtime captioning in edge AI applications, models should be optimized using approaches like model compression, which reduces model size without affecting speed. Quantization (a reduction in computing accuracy for efficiency), information distillation (teaching smaller models with information from bigger ones), and pruning (removing superfluous weights to accelerate inference) can further enhance performance. These improvements will make AIpowered captioning more accessible to real-time assistive devices, autonomous systems, augmented reality (AR), and human-computer interface (HCI) applications. Furthermore, using lightweight models on edge devices can improve applications like smart glasses for visually impaired people, robotic vision, and real-time video captioning in mobile apps.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Aneja, J., Deshpande, A., & Schwing, A. G. (2018). Convolutional image captioning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5561–5570. [CrossRef]

[2] Bai, S., & An, S. (2018). A survey on automatic image caption generation. *Neurocomputing, 311*, 291–304. [CrossRef]

[3] Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., & Zhang, L. (2017). Bottom-up and top-down attention for image captioning and vqa. *arXiv preprint arXiv:1707.07998, 2*(*4*), *8*.

[4] Chen, X., & Zitnick, C. L. (2015). Mind's eye: A recurrent visual representation for image caption generation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 2422–2431. [CrossRef]

[5] Ghandi, T., Pourreza, H., & Mahyar, H. (2023). Deep learning approaches on image captioning: A review. *ACM Computing Surveys, 56*(3), 1–39. [CrossRef]

[6] Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3128–3137. [CrossRef]

[7] Chen, L., Wei, X., Li, J., Dong, X., Zhang, P., Zang, Y., ... & Others. (2024). ShareGPT4Video: Improving video understanding and generation with better captions. *Advances in Neural Information Processing Systems, 37*, 19472–19495.

[8] Rastogi, R., Rawat, V., & Kaushal, S. (2024). Demonstration and analysing the performance of image caption generator: Efforts for visually impaired candidates for Smart Cities 5.0. *International Journal of Advanced Mechatronic Systems, 11*(3), 161–178. [CrossRef]

[9] Jamil, A., Mahmood, K., Villar, M. G., Prola, T., Diez, I. D. L. T., Samad, M. A., & Ashraf, I. (2024).

Deep learning approaches for image captioning: Opportunities, challenges and future potential. *IEEE Access, 12*, 12345–12367. [CrossRef]

[10] Vo-Ho, V. K., Luong, Q. A., Nguyen, D. T., Tran, M. K., & Tran, M. T. (2019). A smart system for text-lifelog generation from wearable cameras in smart environment using concept-augmented image captioning with modified beam search strategy. *Applied Sciences, 9*(9), 1886. [CrossRef]

[11] Cornia, M., Baraldi, L., & Cucchiara, R. (2019). Show, control and tell: A framework for generating controllable and grounded captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8307–8316. [CrossRef]

[12] Chen, L., Jiang, Z., Xiao, J., & Liu, W. (2021). Human-like controllable image captioning with verb-specific semantic roles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16846–16856. [CrossRef]

[13] Yang, X., Yang, Y., Ma, S., Li, Z., Dong, W., & Woz´niak, M. (2024). SAMT-generator: A second-attention for image captioning based on multi-stage transformer network. Neurocomputing, 593, 127823. [CrossRef]

[14] Zhao, W., Du, S., & Emery, W. J. (2017). Object-based convolutional neural network for high-resolution imagery classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 10*(7), 3386–3396. [CrossRef]

[15] Wang, Q., Deng, H., Wu, X., Yang, Z., Liu, Y., Wang, Y., & Hao, G. (2023). LCM-Captioner: A lightweight text-based image captioning method with collaborative mechanism between vision and text. *Neural Networks, 162*, 318–329. [CrossRef]

[16] Nag, I. (2021). Systematic literature review of deep visual and audio captioning (Technical Report No. 133909). Tampere University.

[17] Li, X., Xu, C., Wang, X., Lan, W., Jia, Z., Yang, G., & Xu, J. (2019). COCO-CN for cross-lingual image tagging, captioning, and retrieval. IEEE Transactions on Multimedia, 21(9), 2347–2360. [CrossRef]

[18] Hoxha, G. (2022). *Image captioning for remote sensing image analysis* [Doctoral dissertation, University of Trento].

[19] Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2016). Show and tell: Lessons learned from the 2015 MSCOCO image captioning challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(4), 652–663. [CrossRef]

[20] Jaiswal, S., Pallthadka, H., Chinchewadi, P., & Jaiswal, T. (2023). An extensive analysis of image captioning models, evaluation measures, and datasets. *International Journal of Multidisciplinary Science Research Review, 1*(1), 21–37.

Mr. Abdullah Khan is a student scholar in Chandigarh University, Mohali, India. His area of interest is Artificial Intelligence, Image Processing and Deep Learning. (Email: khanabdul4t4@gmail.com)

Mr. Jaswinder Singh is a highly accomplished academician and researcher with an impressive track record of excellence in computer science. Currently, he is serving as an Assistant Professor in Chandigarh University, Mohali, India. He has authored over 10 peer-reviewed research papers. (Email: jassi724@gmail.com)