Multi-layer Gated Recurrent Unit based Recurrent Neural Network for Image Captioning

Özkan Çaylı
Centre for Vision, Speech and Signal Processing
University of Surrey, Guildford, GU2 7XH, United Kingdom
o.cayli@surrey.ac.uk

Volkan Kılıç
Department of Electrical and Electronics Engineering
İzmir Kâtip Çelebi University, Çiğli, 35620, Türkiye
volkan.kilic@ikcu.edu.tr

Aytuğ Onan
Department of Computer Engineering
İzmir Kâtip Çelebi University, Çiğli, 35620, Türkiye
aytug.onan@ikcu.edu.tr

Wenwu Wang
Centre for Vision, Speech and Signal Processing
University of Surrey, Guildford, GU2 7XH, United Kingdom
w.wang@surrey.ac.uk

Generating natural language descriptions of an image, namely image captioning, has received much attention in computer vision and natural language processing. Recent image captioning models are mainly based on the encoder-decoder framework in which visual information is extracted by an encoder, e.g. using convolutional neural network (CNN), and captions are generated by a decoder, e.g. using recurrent neural network (RNN). Although this framework is promising for image captioning, there are still issues in the RNN decoder for exploiting the visual information to generate grammatically and semantically correct captions. More specifically, the RNN decoder has limited ability in dealing with long-term complex dependencies, leading to ineffective use of contextual information from the encoded data. To address this issue, in this paper, we introduce a multi-layer gated recurrent unit (ML-GRU) within the conventional RNN decoder, which enables the modulation of the relevant information flow inside the unit, and thus leads to the generation of semantically coherent captions. The proposed ML-GRU based RNN decoder has been extensively evaluated on the MSCOCO dataset, and experimental results demonstrate the advantage of our proposed approach over the state-of-the-art approaches across multiple performance metrics.

Keywords: Image captioning; multi-layer GRU; natural language processing.

1. Introduction

Image captioning aims to generate grammatically correct and human-readable descriptions of an image using techniques from computer vision (CV) and natural

language processing (NLP). This task leverages the connection between CV and NLP and has attracted increasing interest, due to its potential applications such as image indexing or retrieval and virtual assistants for visually-impaired people [2, 8, 19, 27, 28, 42]. Image captioning is a challenging task because it requires an advanced level of understanding of an image, including the recognition of the objects and actions in the image, in order to generate meaningful captions with proper linguistic properties. Therefore, it goes beyond the conventional CV tasks such as image classification and object detection. Early efforts to address this problem in the literature have considered the use of either retrieval-based or template-based models before using deep neural networks. Recently, the encoder-decoder [21] based neural structure has emerged, which is promising and has become a popular model for image captioning. This model is composed of two sub-networks, where the encoder aims to generate a feature representation of an image using methods such as CNN, while the decoder translates this representation into natural language descriptions using methods such as RNN.

For the encoders of the captioning systems, the CNN architectures like Inception-v3 [36], NASNet-Large [52] (neural architecture search network), Xception [4], and ResNet152 v2 [11] are popular choices. Inception-v3 [36] is a 42-layered deep CNN architecture that uses the asymmetric approach to decompose a kernel of large-scale convolution into a small-scale kernel of convolution. NASNet [52] is designed using reinforcement learning and contains two types of cells, namely, the normal cell, which keeps the width and height of the feature map, and the reduction cell, which reduces the width and height of the feature map by half. Xception [4] is a deep CNN consisting of 36 convolutional layers with 14 modules that have linear residual connections around them and a logistic regression layer for feature extraction. This architecture is obtained by modifying Inception-v3 with depth-wise separable convolutional layers. ResNet152 v2 [11] is a deep CNN, which is composed of residual nets with 152 layers. Unlike the ResNetV1, this architecture uses the normalization of the stack before each weight layer. The ResNet152 v2 architecture with the removed classification layer extracts the high-level image feature vector of the input image using convolution and pooling layers.

The visual information of images extracted by the encoders is then utilized in language decoders to convert this information word-by-word into natural language captions. The conventional RNN based decoders, however, have vanishing and exploding gradient problems. As a result, they are not effective in exploiting long-term temporal dependencies [14]. Long short-term memory (LSTM) [13] and gated recurrent unit (GRU) [5] networks are proposed to address these problems. LSTM uses memory cells to retain information for long periods, while GRU does not use additional memory cells to maintain the flow of information. When the RNN-based language decoders are used for caption generation, the visual information can be fed either directly into the RNN or in a layer preceding the RNN [19, 38].

Several RNN-based architectures have been proposed, which can be categorized

into the following four: init-inject [30], pre-inject [48], par-inject [17], and merge [29]. The visual information can be fed as a latent vector to the initial hidden state of the RNN in init-inject [6, 26], while the latent vector is used as the initial input of the RNN in the pre-inject architecture [44]. The latent vector is used with the word vectors of the caption prefix in parallel as an input to the RNN in the par-inject architecture [7]. Different from the above architectures, the latent vector is not fed to the RNN directly in the merge architecture as the image is presented to the language model after the caption prefix is generated by the RNN [3, 38].

Although the current encoder-decoder framework improves captioning accuracy compared to its counterparts, effectively extracting and employing contextual information from encoded data remains a challenge that results in insufficient performance in captioning. This paper introduces a novel image captioning model that utilizes NASNet-Large for image encoding and a multi-layer GRU based decoder under the init-inject architecture, thereby enhancing the use of visual information for accurate caption generation. Based on extensive experimental studies, NASNet-Large is found to be adequate for encoding visual information. The motivation behind using GRU is two-fold. First, GRU needs fewer parameters and is computationally cheaper than LSTM as GRU has one hidden state vector while LSTM has two state vectors, namely, hidden and cell states [45]. In addition, GRU has two gates, i.e. the update and reset gates, while LSTM has three gates, i.e. the input, forget, and output gates. Second, the GRU with one hidden state vector offers an excellent fit for the requirement of the init-inject architecture in terms of computational efficiency in practical implementation [38]. The number of layers in GRU is incremented to ensure the modulation of the most relevant information flow inside the unit. A higher number of upper layers deployed in the multi-layer GRU can provide detailed contextual information from the data, thereby providing an enhanced prediction model [18, 20]. As GRUs are operated on sequence data, adding layers will increase the level of abstraction over time for input observations. In turn, this can provide chunking of observations over time or represent the data at various time scales.

Integrating an ML-GRU into RNN enhances the ability of the decoder to retain important semantic image information, thereby improving caption generation. Furthermore, to achieve high-quality image features, we implemented NASNet-Large. This integration enriches the quality of the encoded data, thus elevating the coherence of the generated captions. Although ML-GRUs are widely used in various applications, our study presents an implementation within the field of image captioning for the first time. Our approach combines a structured integration of GRU layers under the init-inject architecture, with each GRU layer fine-tuned. Each layer in the ML-GRU is designed within the init-inject architecture to refine the decoder at every step. The init-inject architecture structure integrates initial image features directly into the GRU, allowing the first few layers to focus on encoding critical visual information that establishes the context for the caption. As the processing

advances through the layers, the architecture incrementally injects more detailed and complex semantic features, which the later layers decode to form a coherent and contextually rich caption. This approach ensures accurate recognition of visual elements and their meaningful linguistic translation. By adopting such a precise configuration, the approach aims to advance the state-of-the-art image captioning with enhanced accuracy and contextual relevance in the generated captions. Experimental results on the MSCOCO dataset show the advantage of our proposed approach over the state-of-the-art approaches for caption generation with a higher performance metric score.

The major contributions of this paper can be summarized as follows.

- We propose a new approach to the neural encoder-decoder framework of image captioning by introducing multi-layer GRU based RNN, which refines the decoder to evaluate the image attributes extracted in the encoder for enhanced image captioning. This approach was designed under the init-inject architecture, and to the best of our knowledge, this is the first time that the multi-layer GRU is exploited in the encoder-decoder based image captioning models.
- We perform comprehensive experiments on the MSCOCO dataset and show that the proposed approach significantly outperforms the state-of-the-art approaches consistently across different performance metrics. We also investigate the optimal number of GRU layers to be used for image captioning.

2. Theoretical background

In this section, we present background details about a conventional encoder-decoder system for image captioning, i.e., the main components utilized in the image encoder and the language decoder.

2.1. *Image encoder*

An image encoder converts image data into a feature vector, which represents the information of the image. CNNs are predominantly employed in the current image captioning frameworks as an encoder due to their capabilities for dealing with high-dimensional data and outstanding performance on feature extraction. The convolutional, pooling, and fully connected layers are the main building blocks of a conventional CNN. The convolutional layer uses a set of learnable filters to create the feature map of the image. The pooling layer reduces the spatial size of the feature map, whereas the fully connected layer produces the output based on all input from the previous layers [51]. Several well-known CNN architectures have been employed as the encoder, such as ResNet [11], Xception [4], NASNet-Large [52], and Inception-v3 [36]. The image encoder extracts a high-level feature representation of an input image. Then, the features of each input image will be fed into the language decoder to generate captions, as discussed next.

2.2. *Language decoder*

A language decoder utilizes the feature representation to describe the image with grammatically and semantically correct sentences which are generated word-by-word. The main components of the decoder (RNN, GRU, embedding layer, and dense layer) and init-inject architecture are explained next.

Recurrent neural network RNN, a type of deep neural network, is able to model long-term dependencies in sequential data and suitable for NLP tasks such as speech recognition, machine translation, and image captioning [51]. Each output is calculated by repeatedly processing the same function over each instance of the sequence in RNN.

RNN computes the hidden vector sequence $\mathbf{h} = (h_1, ..., h_T)$ and output vector sequence $\mathbf{y} = (y_1, ..., y_T)$ using the input sequence $\mathbf{x} = (x_1, ..., x_T)$ with the variable-length for $t = 1, ..., T$. The hidden vector h_t at time step t is computed with the input vector x_t as $h_t = f(Wh_{t-1} + Ux_t)$ where W and U denote the weight matrices, and f denotes a nonlinear activation function such as *tanh*, *ReLU*, and *sigmoid*. The output vector is computed as $y_t = f(Vh_t)$, where V is a matrix that connects the current hidden layer with the current output layer [51]. RNNs employ the information in arbitrarily long sequences in theory but suffer from vanishing and exploding gradients in practice and cannot capture long-term dependencies. Despite the fact that a variety of RNN-based architectures could be used, such as LSTM, as a proof of concept, the GRU is used here, which is more feasible in handling vanishing and exploding gradients problems, employed for processing sequential data to generate captions in our experiments.

Gated recurrent unit GRU, which is a type of RNN with a gating mechanism, has been implemented to address the aforementioned issues. GRU consists of a hidden state and two gates: update and reset [5]. In GRU, the transition has been carried on based on the following equations [5]:

$$r_t = \sigma\left(W_r x_t + U_r h_{t-1}\right) \tag{1}$$

$$z_t = \sigma\left(W_z x_t + U_z h_{t-1}\right) \tag{2}$$

$$u_t = \tanh\left(Wx_t + U\left(r_t \bigodot h_{t-1}\right)\right) \tag{3}$$

$$h_t = \left(1 - z_t\right) h_{t-1} + z_t u_t \tag{4}$$

where r_t, z_t and u_t denote reset gate vector, update gate vector, and candidate hidden vector, respectively. The subscripts r and z in W_r and W_z denote the weights of the reset and update gates. σ and *tanh* are the sigmoid and tangent hyperbolic activation functions, respectively. $\bigodot$ denotes the element-wise multiplication operator. h_{t-1} is taken from the previous GRU as input, and the output of GRU, y_t is

calculated with the sigmoid function as

$$y_t = \sigma(W_o h_t + b) \tag{5}$$

where the subscript o denotes the weight of the output vector, and b is the bias. This makes it easier to configure stacked or multi-layer GRU architectures with two or more layers that outperform the conventional RNN-based architectures on many NLP tasks, including language modeling [33].

Embedding and dense layers In decoders, the embedding layer processes texts to get meaningful units or tokens, resulting in the extraction of linguistic features. The embedding layer produces embedding vectors of a specified size, representing the tokens (or words) with numeric components. The embedding vector includes the linguistic features of tokens that are fed into the GRU. A fully connected dense layer with an activation function is used to calculate the probabilities of each word in the vocabulary being selected for representing the image features, which leads to the next word in the caption to be predicted.

Init-inject architecture In current image captioning approaches, the image and linguistic features are fed into the RNN with several architectures, including init-inject, pre-inject, par-inject, and merge. Here, init-inject architecture is employed due to its superior performance in generation and retrieval measures compared to its counterparts [38]. The feature vector is utilized as an initial hidden state vector for the RNN under the assumption that the image feature vector has the same size as the hidden state vector. The linguistic vector of the token is then fed to the RNN as an input vector after initialization. An efficient way to implement the RNN decoder under the init-inject architecture is to use the GRU due to its simplicity in terms of state and gate vectors. In contrast, LSTM is more complex due to the use of two-state vectors with three gates, causing multiple versions of the init-inject architecture to be tested for captioning performance. In addition, GRU has fewer hyper-parameters to tune and allows its initial state to grow without bound from an activation function, resulting in the best compatibility with the init-inject architecture [38].

3. Proposed image captioning approach

This section presents a new approach to enhance the image captions by introducing multi-layer GRU to the text decoder.

3.1. *Multi-layer gated recurrent unit for image captioning*

The proposed image captioning approach consists of two steps: image encoder and text decoder. First, the image encoder is utilized to extract features from an image.

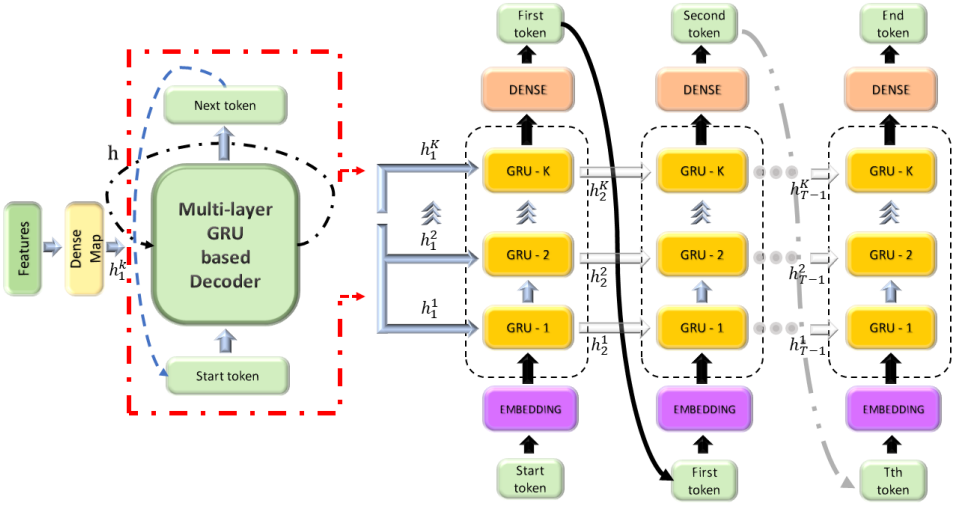


Fig. 1. The proposed multi-layer GRU based decoder (inside the red dashed line) is given on the left side while unfolded on the right side.

Then, these features are fed into the text decoder that processes the features to generate a caption word-by-word. CNN based encoder employed here is a recently emerged framework that has been found to be promising for feature extraction of an image. The NASNet-Large model is utilized as a CNN architecture where all image features are obtained after the average pooling layer, which returns a 4032-element vector.

The approach proposed in this study involves a multi-layer GRU based decoder, as depicted in Fig. 1. This decoder provides a novel solution to several limitations in the current literature, such as efficient visual attributes injection and modulation of the relevant information flow. The decoder architecture comprises an embedding layer, multiple GRU layers, and a dense layer, which are utilized under the init-inject architecture. This architecture facilitates the parallel processing of image features obtained from a dense map and linguistic features, derived from the embedding layer.

The multi-layer GRU is a combination of K-GRU for $k = 1, ..., K$, while $h_t^{(k)}$ and $x_t^{(k)}$ are defined as the hidden and input vector for the kth GRU layer. Each initial hidden vector $(h_1^{(k)})$ contains image features as a separate vector with reduced size from 4032 to the 512-element vector by the dense map to feed the multi-layer GRU at $t = 1$. For the subsequent iterations, multi-layer GRU is fed by the updated hidden vector from the previous iteration $(h_{t-1}^{(k)})$ rather than the dense map. The first GRU layer is located after the embedding layer, which generates the predefined size of a meaningful embedding vector, namely the linguistic features, using the start token. The embedding vector is processed at the first GRU layer, leading to the

first output vector ($y_1^{(1)}$), which is the input of the next GRU layer ($x_1^{(2)}$). The same procedure is repeated K-times until $y_1^{(K)}$ is generated as in Eq. (6), which is the input for the dense layer.

$$y_1^{(K)} = \sigma(W^{(K)} h_1^{(K)} + b^{(K)}) \tag{6}$$

To generate the first token, the *argmax* function has been employed on the output of the dense layer d_1, which is computed as:

$$d_1 = f(W y_1^{(K)} + b) \tag{7}$$

then the output and hidden state are carried to the RNN as input and hidden state, respectively, to generate the next token. This process is continued until an end-of-caption token is generated. In the end, the generated tokens are converted into their corresponding words employing a vocabulary that is created from the reference captions of the training set.

4. Experimental evaluations

This section evaluates the proposed captioning approach on the MSCOCO dataset [25], and a performance comparison with state-of-the-art approaches is presented.

4.1. *Dataset*

There are several well-known datasets for performance evaluations of image captioning systems, such as Flickr [34], VizWiz-Captions [10], and MSCOCO [25]. Flickr8k and Flickr30k are the sub-datasets of Flickr. Flickr8k contains 8000 images consisting of 6000 training, 1000 test, and 1000 validation images, whereas Flickr30k contains 29783 training, 1000 test, and 1000 validation images. The VizWiz-Captions dataset consists of 39181 images captured by blind people. The MSCOCO dataset contains 118287 training, 41000 test, and 5000 validation images [25] and each image is described with five reference captions. MSCOCO is the most suitable dataset for the evaluation of our proposed image captioning approach due to its various images with semantically rich reference captions.

4.2. *Performance metrics*

To analyze the performance of the compared captioning approaches, several metrics are employed, including bilingual evaluation understudy (BLEU) [32], consensus-based image description evaluation (CIDEr) [43], metric for evaluation of translation with explicit ordering (METEOR) [22], recall-oriented understudy for gisting evaluation-longest common subsequence (ROUGE-L) [24] and semantic propositional image caption evaluation (SPICE) [1]. The key points of those metrics are summarized as follows:

- BLEU-n (with n = 1, ..., 4) is a machine translation metric that uses n-gram (e.g. BLEU-2 for 2-grams) pairs to compare a machine-generated caption with the human-generated ground truth captions [32]. For each n-gram level, BLEU calculates the precision, which is the ratio of the number of n-grams in the generated caption that match those in any reference caption to the total number of n-grams in the candidate text. This method ensures that the generated caption not only uses correct words but also forms them into coherent phrases and sentences as found in the reference. One issue with the BLEU-n is that a higher score can be measured when searching pairs on short captions even though the result is incorrect. A brevity penalty is used to overcome this issue, which chooses the closest reference length if more than one reference is used for a candidate sentence.
- CIDEr is designed especially for image captioning tasks to ensure the consistency of a generated caption, calculating the different weights of n-gram words with term-frequency-inverse document frequency [43]. It begins by computing the Term Frequency-Inverse Document Frequency (TF-IDF) weights for n-grams in the generated captions to emphasize more informative n-grams while reducing the impact of common ones. Each caption is then represented as a vector of these TF-IDF scores. CIDEr evaluates the cosine similarity between the TF-IDF vector of the generated caption and that of each reference caption, averaging these similarities to derive a consensus score. This process is repeated across multiple n-gram lengths to capture both detailed and broader semantic content. The final CIDEr score is normalized by the average cosine similarity of ideal captions to balance the evaluation and prevent bias towards longer captions.
- METEOR is an automatic machine translation metric that generalizes unigram matches between a machine-generated caption and a human-generated ground truth captions [22]. This metric is developed to address the weakness of the BLEU-n incorporating semantic and morphological similarities, enhancing accuracy. METEOR calculates its score using the harmonic mean of precision (the proportion of correctly predicted words in the generated caption) and recall (the proportion of reference words captured in the generated caption), and integrates a penalty for word order discrepancies. This penalty is based on the number of contiguous word chunks in the hypothesis that appear out of order compared to the reference captions, ensuring generated captions are not only accurate in word choice but also in structural integrity, aligning more closely with human judgment.
- ROUGE-L measures sentence-to-sentence similarity based on the longest common subsequence (LCS) between the generated caption and a set of reference captions [24]. It computes recall, precision, and F-measure scores, reflecting how much of the LCS is captured in both the generated and

Table 1. Comparison of different CNN encoders with single-layer GRU.

CNN	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR	SPICE	CIDEr
ResNet152 v2	0.686	0.503	0.359	0.258	0.497	0.221	0.148	0.801
Inception-v3	0.693	0.513	0.368	0.264	0.506	0.230	0.161	0.851
Xception	0.702	0.520	0.373	0.265	0.508	0.230	**0.162**	0.859
NASNet-Large	**0.707**	**0.524**	**0.376**	**0.270**	**0.510**	**0.231**	0.161	**0.876**

reference captions. This metric is especially useful in tasks such as summarization and machine translation, where the sequence of words impacts the coherence and fluency of the output. The ability of ROUGE-L to handle multiple references and its sensitivity to word order makes it a useful tool for assessing the linguistic accuracy and relevance of generated captions.

- SPICE is also specially designed to evaluate image captioning tasks. It measures the semantic correctness of the caption using scene graphs that contain objects, attributes, and relationships between them [1]. These graphs represent objects, attributes, and their interrelationships, allowing SPICE to evaluate captions based on their semantic content rather than syntactic similarity. It calculates an F-Score to balance precision and recall, focusing on the correctness and completeness of semantic elements.

For all the metrics, a higher score indicates better performance. Our results are sorted based on CIDEr and SPICE metrics due to their better correlation with human assessment compared to BLEU-n, METEOR, and ROUGE-L.

4.3. *Results and discussion*

To construct an image captioning system with high performance, we have analyzed four different CNN architectures as an encoder in conjunction with a multi-layer GRU based decoder. In this regard, the Inception-v3, Xception, ResNet152 v2, and NASNet-Large with five different layer-sized GRU were evaluated in terms of BLEU-n, CIDEr, METEOR, SPICE, and ROUGE-L metrics. All these configurations have been evaluated based on hyper-parameter optimization.

Our proposed multi-layer GRU based decoder takes linguistic features from the embedding layer. Two critical parameters based on linguistic features for the performance of image caption generation are the embedding vector size and the vocabulary size. The size of the embedding vector is typically set between 50 and 300 [31]. The embedding vector with a small size does not capture the word relations completely, whereas the large embedding vectors cause overfitting. The size of the embedding vector affects the training time, computational costs, and the performance of embedding. The vocabulary size, which has a critical role in the image captioning tasks, is determined based on the number of common words in all reference captions and usually varies from 10000 to 40000 words [35]. To optimize the embedding vector and vocabulary values, our proposed multi-layer GRU based

Table 2. Comparison of different CNN encoders with multi-layer GRU.

CNN	# of Layers	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR	SPICE	CIDEr
ResNet152 v2	3	0.679	0.494	0.347	0.244	0.488	0.219	**0.150**	0.782
	6	0.675	0.492	0.349	0.248	0.490	**0.221**	0.150	**0.786**
	9	**0.683**	**0.498**	**0.352**	**0.249**	**0.493**	0.219	0.148	0.778
	12	0.546	0.326	0.184	0.105	0.414	0.152	0.090	0.420
	15	0.544	0.325	0.182	0.104	0.411	0.151	0.093	0.421
Inception-v3	3	0.678	0.499	0.356	0.254	0.496	**0.229**	**0.158**	**0.821**
	6	0.680	0.500	0.357	0.254	0.496	0.227	0.157	0.818
	9	**0.689**	**0.506**	**0.362**	**0.258**	**0.497**	0.225	0.154	**0.821**
	12	0.547	0.334	0.192	0.112	0.420	0.154	0.091	0.452
	15	0.555	0.335	0.191	0.110	0.417	0.157	0.093	0.453
Xception	3	0.698	0.513	0.366	0.261	0.501	0.228	0.160	0.846
	6	0.694	0.509	0.363	0.259	0.499	0.227	0.158	0.844
	9	**0.702**	**0.519**	**0.371**	**0.263**	**0.505**	**0.229**	**0.162**	**0.850**
	12	0.692	0.507	0.358	0.251	0.496	0.221	0.155	0.792
	15	0.559	0.343	0.195	0.115	0.420	0.158	0.093	0.467
NASNet-Large	3	0.690	0.513	0.369	0.266	0.506	**0.237**	**0.169**	**0.884**
	6	0.695	0.514	0.370	0.265	0.505	0.236	0.168	0.884
	9	**0.705**	**0.522**	**0.374**	**0.268**	**0.507**	0.235	0.168	0.878
	12	0.559	0.343	0.197	0.115	0.427	0.161	0.097	0.488
	15	0.561	0.343	0.195	0.113	0.422	0.161	0.100	0.484

decoder is tested under ten different vocabulary sizes, including 250, 500, 750, 1000, 2000, 3000, 5000, 10000, 20000, and 40000, and eight different embedding vector sizes (namely, vector sizes with 25, 50, 75, 100, 125, 150, 200, and 250). The optimization tests were carried out by keeping one of two parameters fixed due to the high training time and computational cost. In the encoder side, the Inception-v3 is employed as a reference CNN architecture, while a single-layer GRU based decoder is used in the decoder. First, this reference system was evaluated under different performance metrics with ten different vocabularies and the embedding vector of fixed-size, as 100. The best CIDEr metric was observed when the vocabulary size was 750. Then, the reference system was evaluated under the same performance metrics with eight different embedding vectors and the vocabulary of fixed-size, as 750. The best CIDEr metric was observed when the embedding vector size was 100. Hence, the embedding vector size and vocabulary size have been determined based on the empirical analysis of the aforementioned configurations.

Three different CNN based encoders (i.e., Xception, ResNet152 v2, and NASNet-Large) are employed to observe the best CNN architecture compatible with these embedding vector and vocabulary sizes, 100 and 750, respectively. The evaluation results are given in Table 1. The NASNet-Large based encoder outperforms the other three CNNs. The experiments were employed on NASNet-Large CNN architecture as a reference due to its promising results. To find the optimum parameters according to layer size, NASNet-Large with three-layer GRU was evaluated under the same performance metrics with ten different vocabularies and the embedding vector of fixed-size, as 100. The best CIDEr metric was observed when the vocabulary size was 10000. Then, the best CIDEr metric was observed when the embedding vector size was 125. Using these parameters (10000 for vocabulary and 125 for embedding vector), three CNN (Inception-v3, ResNet152 v2, and Xception) based encoder was employed to observe the best result for 3-layer GRU. Applying

the same strategy to the 6, 9, 12, and 15 layer GRU, the optimum parameters were determined as 150, 75, 250, and 200 for the embedding vector size; 20000, 2000, 2000, and 40000 for the vocabulary size, respectively.

The empirical analysis with different vocabulary sizes with a fixed-size embedding vector indicates that the CIDEr metric gradually increases until the 9-layer GRU, where the maximum level has been reached. The performances of multi-layer GRU with four different CNN configurations have been listed in Table 2. The empirical results listed in Table 2 indicate that increasing the number of layers until 12-layer can enhance the predictive performance in the proposed image captioning approach. Among all the configurations, 9-layer GRU architecture outperforms the other compared schemes in terms of BLEU-n and ROUGE-L metrics, and 3-layer GRU architecture outperforms the other schemes in terms of METEOR, SPICE, and CIDEr.

Table 3 presents a comprehensive performance evaluation of the proposed 9-layer GRU against recent state-of-the-art image captioning architectures utilizing the MSCOCO dataset. We compare our approach with the following frameworks. The comparative frameworks include StyleNet [9], which employs a factored LSTM for extracting stylistic elements in captions. A customizable captioning model tailored for specific application requirements is described in [41]. SemStyle [30] aims to generate semantically consistent styled captions. An encoder-decoder architecture that utilizes the Inception-v3 model and LSTM for caption generation is presented in [44]. The gLSTM [15] extends LSTM by injecting semantic information from images into each unit, aiming to align captions with image content. Phi-LSTM [37] uses phrases to generate image captions rather than the conventional sequential word-by-word approach. A CNN+CNN based approach designed for handling natural language attributes is outlined in [46]. The Mixture of Recurrent Experts system, which focuses on generating diverse styled captions, is detailed in [12]. Lastly, [47] presents an attention-based method that applies both soft and hard attention techniques. Additionally, recent advancements in zero-shot and controlled captioning are also evaluated, including MeaCap [49], DeCap [23], ConZIC [50], a zero-shot video captioning strategy using GPT-2 and CLIP models [39], and the ZeroCap approach [40] which merges a visual-semantic framework with a large language model. Evaluation employs metrics such as BLEU-1 to BLEU-4, ROUGE-L, METEOR, SPICE, and CIDEr. The proposed approach consistently outperforms others, especially in BLEU-1, BLEU-2, BLEU-3, SPICE, and CIDEr, highlighting advanced context interpretation and description generation capabilities. Moreover, this is further supported by competitive performance in BLEU-4, ROUGE-L, and METEOR. The integration of a 9-layer GRU facilitates the handling of complex temporal dynamics and ensures the maintenance of comprehensive context during caption generation. The results underscore the potential of the proposed approach in progressing image captioning research.

The approaches are sorted based on CIDEr metrics, and the highest score is indicated with bold fonts in each column. The proposed approach outperforms

Table 3. Comparison of our proposed approach with some state-of-the-art architectures on MSCOCO dataset.

	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR	SPICE	CIDEr
[9]	0.625	-	-	0.212	-	0.218	0.135	0.664
[41]	-	-	-	**0.270**	0.500	**0.240**	0.009	0.680
[30]	0.653	0.478	0.337	0.238	0.482	0.219	0.157	0.769
[44]	0.667	-	-	0.238	-	0.224	0.154	0.772
[15]	0.663	0.485	0.354	0.262	-	0.230	-	0.813
[37]	0.666	0.489	0.355	0.258	0.497	0.231	0.165	0.821
[46]	0.688	0.513	0.370	0.265	0.507	0.234	-	0.839
[12]	0.679	0.501	0.356	0.252	0.501	0.226	0.166	0.844
[47]	-	-	-	0.250	**0.516**	0.230	-	0.865
MeaCap ToT [49]	-	-	-	0.090	-	0.178	0.127	0.483
MeaCap TF [49]	-	-	-	0.071	-	0.166	0.118	0.425
[40]	-	-	-	0.026	-	0.115	0.055	0.146
[39]	-	-	-	0.022	-	0.127	0.073	0.172
[23]	-	-	-	0.088	-	0.160	0.109	0.421
[50]	-	-	-	0.013	-	0.112	0.050	0.133
Our proposed 9-layer GRU	**0.705**	**0.522**	**0.374**	0.268	0.507	0.235	**0.168**	**0.878**

other methods in BLEU-1, BLEU-2, SPICE, and CIDEr metrics. Fig. 2 shows the ground truth and generated captions by the proposed approach for four images. From those results, we observe that our proposed approach is capable of capturing image information with correct and descriptive captions. For instance, in the first image (Fig. 2 (a)), the generated caption can successfully describe the *chair* and *umbrella* with its color in the image. In the second image (Fig. 2 (b)), the proposed approach identifies a *branch* and the action of *sitting*; in the third image (Fig. 2 (c)), it identifies *cattle* and the action of *grazing*. In the fourth image (Fig. 2 (d)), the proposed approach generates the words *surfboard* and *row*, which accurately describe the content of the image. These examples show that our proposed approach can generate natural sentences related to the image.

Deep learning models, including CNNs and RNNs, often require enhanced interpretability. Our approach, as shown in Fig. 2, sometimes generates captions that diverge from ground truth captions due to a focus on more salient visual elements. In Fig. 2 (a), the generated caption "a chair with a blue umbrella sitting on the sand" omits the ground truth detail "A woman walks out of the ocean towards a beach chair and umbrella". This suggests the approach prioritizes distinct objects like the chair and umbrella over less prominent elements. Similarly, in Fig. 2 (b), the ground truth "The bird is sitting on a small branch of the tree" is simplified to "a bird is sitting on a branch of a tree", missing finer details. In Fig. 2 (c), the contextual elements of the ground truth are reduced in the generated "a herd of cattle grazing on a lush green field". Finally, in Fig. 2 (d), the specifics about positioning of surfboards of the ground truth are absent in "a bunch of surfboards lined up in a row".

The performance of the generated captions in Fig. 2 is evaluated using performance metrics. The proposed approach scores well on BLEU-1 (0.825) indicating good word-level accuracy, but scores lower on higher n-grams (BLEU-4: 0.392), suggesting difficulties with longer phrase accuracies. The METEOR score (0.318) shows moderate semantic alignment, while the ROUGE-L score (0.607) presents

(a)

Ground Truth Captions

A woman walks out of the ocean towards a beach chair and umbrella

This is a chair and umbrella that is sitting near an ocean

A beach chair and umbrella in the sand on the beach

A chair and umbrella sitting on a beach near a person

a chair and a umbrella that is on a beach

Generated Caption

a chair with a blue umbrella sitting on the sand

(b)

Ground Truth Captions

There is a bird sitting on a tree branch

The bird is sitting on the small branch of the tree

A bird is perched on a twig in the trees

There is a bird perched on the tree branch

A gray bird is standing on small brown branch

Generated Caption

a bird is sitting on a branch of a tree

(c)

Ground Truth Captions

A group of cows grazing in a field near a body of water

Several animals standing in the grass near a lake

Several cows grazing on grass near water with trees in the background

a herd of cows graze lazily by the pond

A herd of cattle grazing on top of a grass covered field

Generated Caption

a herd of cattle grazing on a lush green field

(d)

Ground Truth Captions

A row of surfboards sticking out of the sand sitting next to each other

a row of surf boards placed in the sand

Several surfboards standing in a row on the beach

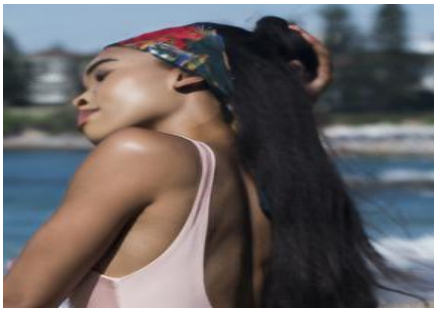
A row of surfboards leaned up against a wood rail in the sand

Many surfboards are propped against a rail on the beach

Generated Caption

a bunch of surf boards lined up in a row

Fig. 2. The generated captions by our proposed approach for four different images from the MSCOCO dataset.



Generated Caption: a beautiful woman in a bathing suit on a beach

(a)

Generated Caption: a close up of a person wearing a blue hat

(b)

Generated Caption: a small white house sitting on a lush green hillside

(c)

Generated Caption: a train traveling down train tracks next to a forest

(d)

Fig. 3. The generated captions by our proposed approach for four different images from the PIPAL dataset.

decent structural understanding. The high CIDEr score (1.669) suggests effective capture of salient information, but the lower SPICE score (0.305) indicates a need for better capturing of finer semantic details of scenes. These results suggest that while the proposed approach effectively identifies prominent objects, it often overlooks contextual and subtler details.

Fig. 3 demonstrates the image captioning capabilities of our proposed approach on four different images from the PIPAL dataset [16]. This dataset includes images generated by GANs, which present unique challenges such as diverse visual artifacts and variations. The captions generated by our approach for these images are as follows:

- (a) “a beautiful woman in a bathing suit on a beach”
- (b) “a close up of a person wearing a blue hat”
- (c) “a small white house sitting on a lush green hillside”

Table 4. Performance metrics scores across supercategories on the MSCOCO dataset, displaying count and percentage of images in the training and validation sets relative to the total images and their respective ratios.

Supercategory	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR	SPICE	CIDEr	Train Images	Val Images	Val/Train (%)	Train/Total (%)	Val/Total (%)
Person	0.7093	0.5350	0.3952	0.2937	0.5251	0.2558	0.1845	0.8963	**64115**	**2693**	**4.20%**	**54.23%**	**53.86%**
Vehicle	0.6906	0.5067	0.3614	0.2589	0.5085	0.2422	0.1739	0.8044	27358	1160	4.24%	23.14%	23.20%
Food	0.7214	0.5432	0.3979	0.2902	0.5188	0.2413	0.1864	0.7967	16255	708	4.36%	13.75%	14.16%
Kitchen	0.7346	0.5686	0.4233	0.3134	0.5354	0.2492	0.1843	0.8757	20792	909	4.37%	17.58%	18.18%
Appliance	**0.7517**	**0.5952**	**0.4365**	0.3208	0.5527	0.2507	0.1705	0.7441	7880	320	4.06%	6.66%	6.40%
Animal	0.7079	0.5441	0.4026	0.2935	0.5316	0.2565	0.1921	**0.9779**	23989	1016	4.23%	20.28%	20.32%
Accessory	0.6876	0.5087	0.3750	0.2778	0.5067	0.2404	0.1724	0.7898	17691	726	4.10%	14.95%	14.52%
Electronic	0.7336	0.5728	0.4352	0.3321	0.5420	0.2573	0.1834	0.8389	12944	597	4.61%	10.94%	11.94%
Indoor	0.7241	0.5595	0.4149	0.3080	0.5307	0.2461	0.1727	0.8639	15847	652	4.12%	13.40%	13.04%
Furniture	0.7362	0.5715	0.4261	0.3179	0.5411	0.2519	0.1825	0.9121	29481	1257	4.26%	24.93%	25.14%
Outdoor	0.6960	0.5171	0.3748	0.2684	0.5158	0.2447	0.1768	0.7875	12880	560	4.35%	10.89%	11.20%
Sports	0.7443	0.5771	**0.4365**	**0.3330**	**0.5547**	**0.2786**	**0.2040**	0.8750	23218	938	4.04%	19.63%	18.76%

- (d) “a train traveling down train tracks next to a forest”

These examples illustrate the robustness of our approach in generating contextually relevant and accurate descriptions even for GAN-generated images, which often contain complex and varied content.

Table 4 presents the evaluation of our image captioning approach on supercategories from the MSCOCO image captioning dataset. In this context, supercategories are classifications within the dataset, such as “Person”, “Vehicle”, “Food”, “Kitchen”, “Appliance”, “Animal”, “Accessory”, “Electronic”, “Indoor”, “Furniture”, “Outdoor”, and “Sports”, which group together related objects and scenes. The purpose of this evaluation is to identify the best and worst-performing categories, thereby highlighting the strengths and weaknesses of our approach. Each row in the table represents a different supercategory, with corresponding performance metrics. The highest-performing supercategory is “Appliance”, which achieves the top scores in several metrics, including BLEU-1 (0.7517), BLEU-2 (0.5952), and BLEU-3 (0.4365). This indicates that our proposed approach is effective in generating accurate and contextually relevant captions for images related to appliances. The high scores across multiple BLEU-n metrics, as well as high scores in ROUGE-L and CIDEr, show that the approach is able to consistently capture details within this supercategory. On the other hand, the “Vehicle” supercategory has some of the lowest scores, particularly in BLEU-4 (0.2589), ROUGE-L (0.5085), and CIDEr (0.8044). This suggests that our approach struggles with accurately captioning images within this category, potentially due to the complex and varied nature of vehicles and their surrounding environments. The lower scores across these metrics indicate a need for further refinement and inclusion of additional contextual data to improve performance in this category. Additionally, the “Animal” supercategory has the highest CIDEr score (0.9779), which reflects the strength of the approach in generating captions that are both relevant and diverse for images featuring animals. This high CIDEr score is supported by strong performances in BLEU-1, BLEU-2, and METEOR, underscoring the effectiveness of the approach furthermore. The number of images in each supercategory also appears to influence the performance metrics. Categories with a larger number of training images, such as “Person” (64115 images) and “Furniture” (29481 images), present high metric scores. In con-

Table 5. ANOVA results for various metrics

Factor	Sum of Squares	Degrees of Freedom	F-statistic	P-value	Performance Metrics
Layer	0.059963	4	174.8955	2.104×10^{-26}	BLEU-1
Supercategory	0.024904	11	26.413	8.034×10^{-16}	BLEU-1
Residual	0.003771	44			BLEU-1
Layer	0.1186	4	274.288	1.737×10^{-30}	BLEU-2
Supercategory	0.049226	11	41.413	1.462×10^{-19}	BLEU-2
Residual	0.004755	44			BLEU-2
Layer	0.167	4	410.037	3.360×10^{-34}	BLEU-3
Supercategory	0.045194	11	40.342	2.448×10^{-19}	BLEU-3
Residual	0.004481	44			BLEU-3
Layer	0.1644	4	421.821	1.832×10^{-34}	BLEU-4
Supercategory	0.034463	11	32.153	1.983×10^{-17}	BLEU-4
Residual	0.004287	44			BLEU-4
Layer	0.032873	4	590.720	1.312×10^{-37}	METEOR
Supercategory	0.007765	11	50.742	2.572×10^{-21}	METEOR
Residual	0.000612	44			METEOR
Layer	0.027437	4	236.179	4.048×10^{-29}	ROUGE-L
Supercategory	0.019794	11	61.960	4.509×10^{-23}	ROUGE-L
Residual	0.001278	44			ROUGE-L
Layer	1.207	4	485.277	9.060×10^{-36}	CIDEr
Supercategory	0.2139	11	31.260	3.392×10^{-17}	CIDEr
Residual	0.027369	44			CIDEr
Layer	0.021805	4	191.152	3.344×10^{-27}	SPICE
Supercategory	0.003872	11	12.343	3.809×10^{-10}	SPICE
Residual	0.001255	44			SPICE

trast, categories with fewer images, like "Appliance" (7880 images) and "Electronic" (12944 images), still perform well, particularly "Appliance", despite having a lower number of images. This shows that while the quantity of training data is important, the ability of the approach to generalize in certain categories can compensate for smaller datasets. However, categories like "Vehicle" with a sufficient number of images (27358 images) still underperform, indicating that both data quantity and the complexity of the supercategory significantly affect performance. In summary, while the approach performs well in categories such as "Appliance" and "Animal", the performance in categories like "Vehicle" indicates a need for improvement. Although categories with a higher number of training images, like "Person" and "Furniture", generally perform well, the results in "Appliance" despite its smaller dataset show that data quantity is not the only factor. Results for the 3, 6, 12, and 15 layers are presented in Tables A1, A2, A3, A4, respectively.

Table 5 presents the results of an Analysis of Variance (ANOVA) test applied to assess the impact of different factors—specifically, approach type and supercategory—on performance metrics. The factors under consideration include the different number of layers being compared (Layer) and the various data groupings or categories (Supercategory). The "Residual" in the table represents the unexplained variance after considering the influence of the factors under study. The Sum of Squares quantifies the variance in the performance metrics that can be attributed to each factor, with higher values indicating a greater contribution to the observed

Table 6. Tukey significance test results for different approach layers (Metric: CIDEr)

Group 1	Group 2	Mean Difference	P-Adj	Lower	Upper	Reject
12-layer	15-layer	-0.3000	0.0000	-0.3763	-0.2238	True
12-layer	3-layer	0.0129	0.9890	-0.0633	0.0892	False
12-layer	6-layer	0.0086	0.9977	-0.0677	0.0848	False
12-layer	9-layer	0.1232	0.0003	0.0469	0.1995	True
15-layer	3-layer	0.3130	0.0000	0.2367	0.3892	True
15-layer	6-layer	0.3086	0.0000	0.2324	0.3849	True
15-layer	9-layer	0.4232	0.0000	0.3470	0.4995	True
3-layer	6-layer	-0.0044	0.9998	-0.0806	0.0719	False
3-layer	9-layer	0.1103	0.0013	0.0340	0.1865	True
6-layer	9-layer	0.1146	0.0008	0.0384	0.1909	True

variance. The Degrees of Freedom shows the number of levels within each factor, while the F-statistic measures the effect of the factor on the performance metrics, with higher values indicating an important effect. The P-value assesses the statistical significance of these effects, with lower values (typically less than 0.05) indicating that the factor has a significant impact. For most metrics, the "Layer" factor consistently shows significant P-values, proving that changes in the number of layers significantly influence approach performance. This is particularly obvious with METEOR and CIDEr scores, where very low P-values means high statistical significance. The "Supercategory" factor varies in its impact across different metrics but generally shows lower significance compared to the "Layer" factor according to P-values. The "Residual" Sum of Squares values are significantly lower compared to those of "Layer" and "Supercategory" which indicates that most of the variance in the performance metrics is successfully explained by these two factors.

Table 6 presents the results of a Tukey significance test, conducted to compare the performance of different layer configurations using the CIDEr metric. The parameters included in the table are as follows: "Group 1" and "Group 2" refer to the pairs of approach layers being compared, with each group representing a different number of layers (3-layer, 6-layer, 9-layer, 12-layer, and 15-layer). The "Mean Difference" column shows the difference in mean CIDEr scores between two groups. A negative value indicates that Group 1 outperforms Group 2, while a positive value indicates that Group 2 outperforms Group 1. This is calculated by subtracting the mean CIDEr score of Group 1 from Group 2. The "P-Adj" column contains the P-values adjusted for multiple comparisons using the Tukey method; a low P-Adj value (typically less than 0.05) indicates that the observed difference in mean CIDEr scores is statistically significant. The "Lower" and "Upper" columns represent the lower and upper bounds of the confidence interval for the mean difference, providing a range within which the true difference in means is likely to fall. The "Reject" column shows whether the null hypothesis—that there is no difference in mean scores

Table 7. Complexity analysis of our approach on training and test, with GPU memory usage, training time, and inference time per image.

		Training					Test	
# of Layers	Parameters (Million)	GPU Memory Allocated (GB)	Max GPU Memory Allocated (GB)	GPU Memory Reserved (GB)	Max GPU Memory Reserved (GB)	Training Time (s)	Inference Time (s) for 100 Images	Average Seconds Per Image
3	187.44	1.82	2.79	3.30	3.30	929.71	41.91	0.42
6	262.97	2.73	4.17	4.70	4.70	1162.84	63.62	0.64
9	338.50	3.63	5.54	6.19	6.19	1397.20	88.26	0.88
12	414.04	4.54	6.92	7.66	7.66	1735.95	110.79	1.11
15	489.57	5.45	8.30	9.24	9.24	2088.02	133.96	1.34

between the two groups—can be rejected. A "True" value indicates a statistically significant difference, while a "False" value means there is no significant difference.

In the comparison between the 12-layer and 15-layer approaches, the mean difference of -0.3 indicates that the 12-layer approach performs better. The significant P-value (P-Adj < 0.001) confirms that this difference is statistically reliable. Although the 3-layer and 6-layer approaches show a slightly better performance than the 12-layer approach, with mean differences of 0.0129 and 0.0086 respectively, these differences are not statistically significant, as indicated by the high P-values (P-Adj $= 0.989$ and 0.9977). The 9-layer approach significantly outperforms the 12-layer approach, with a mean difference of 0.1232 and a highly significant P-value of 0.0003, indicating an improvement in performance. The 15-layer approach performs significantly worse than the 3-layer, 6-layer, and 9-layer approaches, with mean differences of 0.313, 0.3086, and 0.4232 respectively, all of which have highly significant P-values of 0.000. The 3-layer approach performs slightly better than the 6-layer approach with a mean difference of -0.0044, but this difference is not statistically significant, as indicated by the high P-value of 0.9998. The 9-layer approach significantly outperforms both the 3-layer and 6-layer approaches, with mean differences of 0.1103 and 0.1146 respectively, and very low P-values of 0.0013 and 0.0008, confirming substantial improvements in performance. In the supplementary information, additional comparisons using the BLEU-n, ROUGE-L, METEOR, and SPICE metrics are presented in Tables B1, B2, B3, B4, B5, B6, B7, respectively.

Table 7 presents a complexity analysis of our approach, detailing GPU memory usage, training time, and inference time for approaches with 3 to 15 layers. As the number of layers increases, the number of parameters ranges from 187.44 million (3 layers) to 489.57 million (15 layers), causing higher GPU memory usage and longer training and inference times. GPU memory allocation rises from 1.82 GB to 5.45 GB, and training time from 929.71 seconds to 2088.02 seconds. Inference time for 100 images also increases from 41.91 seconds (3 layers) to 133.96 seconds (15 layers), with the average time per image ranging from 0.42 to 1.34 seconds in blind test. This shows the trade-off between approach depth and computational resources, highlighting the need for balance based on available resources and performance requirements. While deeper approaches capture more complex patterns, they demand significantly more resources. The 9-layer approach is the best option for image captioning based on its superior performance metrics and balanced resource requirements. It achieves

the highest scores in BLEU-1 (0.705), BLEU-2 (0.522), BLEU-3 (0.374), BLEU-4 (0.268), and ROUGE-L (0.507) with a reasonable sacrifice for the user in terms of processing time as it increases from 0.42 (3 layers) to 0.88 seconds (9 layers) during the blind test. Additionally, the 9-layer approach maintains a reasonable balance in GPU memory usage and training/inference times compared to deeper approaches. The optimal balance between performance and computational efficiency makes the 9-layer approach the ideal choice for high-quality image captioning.

5. Conclusion

Encoder-decoder frameworks often encounter difficulties in efficiently extracting and utilizing contextual information from encoded data, causing inadequate performance in caption generation. To address these issues, in this paper, we have introduced a novel image captioning approach utilizing the NASNet-Large CNN encoder and a multi-layer GRU based decoder under the init-inject architecture. This modification substantially enhances the ability of the decoder to modulate the relevant information flow within the unit, thereby addressing the long-standing issue of RNN decoders challenged by managing long-term complex dependencies. The outcome is an improved decoder capable of producing semantically consistent and contextually accurate captions. Experimental results obtained from comprehensive evaluations in the MSCOCO dataset validate the effectiveness of our approach. Regarding the different CNN-based encoders considered in the image captioning system, NASNet-Large architecture outperforms the other compared architectures in terms of seven out of eight performance metrics (i.e., BLEU-1, BLEU-2, BLEU-3, BLEU-4, ROUGE-L, METEOR, and CIDEr). The empirical analysis indicates that multi-layer GRU based decoders can yield higher performance compared to single-layer. The performance improvements can be achieved as the number of layers increases up to 9 layers. However, there is a subtle trend of a decrease with 12 and 15 layers. This system was developed to respond to significant challenges in the image captioning field, particularly in generating semantically consistent and grammatically accurate captions. Our future work will focus on the implementation of attention mechanisms to enhance caption generation by prioritizing key parts of the input image.

References

1. P. Anderson, B. Fernando, M. Johnson and S. Gould, Spice: Semantic propositional image caption evaluation, in *European Conference on Computer Vision* (2016) pp. 382–398.
2. S. Aydın, Ö. Çaylı, V. Kılıç and A. Onan, Sequence-to-sequence video captioning with residual connected gated recurrent units, *European Journal of Science and Technology* (35) (2022) 380–386.
3. M. Baran, Ö. T. Moral and V. Kılıç, Akıllı telefonlar için birleştirme modeli tabanlı görüntü altyazılama, *European Journal of Science and Technology* (26) (2021) 191–196.

4. F. Chollet, Xception: Deep learning with depthwise separable convolutions, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017) pp. 1251–1258.
5. J. Chung, C. Gulcehre, K. Cho and Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, *arXiv preprint arXiv:1412.3555* (2014).
6. J. Devlin, H. Cheng, H. Fang, S. Gupta, L. Deng, X. He, G. Zweig and M. Mitchell, Language models for image captioning: The quirks and what works, *arXiv preprint arXiv:1505.01809* (2015).
7. J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko and T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015) pp. 2625–2634.
8. B. Fetiler, Ö. Çaylı, Ö. T. Moral, V. Kılıç and A. Onan, Video captioning based on multi-layer gated recurrent unit for smartphones, *European Journal of Science and Technology* (32) (2021) 221–226.
9. C. Gan, Z. Gan, X. He, J. Gao and L. Deng, Stylenet: Generating attractive visual captions with styles, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017)
10. D. Gurari, Y. Zhao, M. Zhang and N. Bhattacharya, Captioning images taken by people who are blind, in *European Conference on Computer Vision* (2020) pp. 417–434.
11. K. He, X. Zhang, S. Ren and J. Sun, Deep residual learning for image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016) pp. 770–778.
12. M. Heidari, M. Ghatee, A. Nickabadi and A. Pourhasan Nezhad, Diverse and styled image captioning using singular value decomposition-based mixture of recurrent experts, *Concurrency and Computation: Practice and Experience* **34**(22) (2022) p. e6866.
13. S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Computation* **9**(8) (1997) 1735–1780.
14. M. Z. Hossain, F. Sohel, M. F. Shiratuddin and H. Laga, A comprehensive survey of deep learning for image captioning, *ACM Computing Surveys* **51**(6) (2019) 1–36.
15. X. Jia, E. Gavves, B. Fernando and T. Tuytelaars, Guiding the long-short term memory model for image caption generation, in *Proceedings of the IEEE International Conference on Computer Vision* (2015) pp. 2407–2415.
16. G. Jinjin, C. Haoming, C. Haoyu, Y. Xiaoxing, J. S. Ren and D. Chao, Pipal: a large-scale image quality assessment dataset for perceptual image restoration, in *European Conference on Computer Vision* (2020) pp. 633–651.
17. A. Karpathy and L. Fei-Fei, Deep visual-semantic alignments for generating image descriptions, in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (2015) pp. 3128–3137.
18. R. Keskin, Ö. T. Moral, V. Kılıç and A. Onan, Multi-gru based automated image captioning for smartphones, in *29th Signal Processing and Communications Applications Conference* (2021) pp. 1–4.
19. R. Keskin, Ö. Çaylı, Ö. T. Moral, V. Kılıç and A. Onan, A benchmark for feature-injection architectures in image captioning, *European Journal of Science and Technology* (31) (2021) 461–468.
20. V. Kılıç, Deep gated recurrent unit for smartphone-based image captioning, *Sakarya University Journal of Computer and Information Sciences* **4**(2) (2021) 181–191.
21. R. Kiros, R. Salakhutdinov and R. S. Zemel, Unifying visual-semantic embeddings with multimodal neural language models, *arXiv preprint arXiv:1411.2539* (2014).

22. A. Lavie and A. Agarwal, Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments, in *Proceedings of the Second Workshop on Statistical Machine Translation* (2007) pp. 228–231.
23. W. Li, L. Zhu, L. Wen and Y. Yang, Decap: Decoding clip latents for zero-shot captioning via text-only training, *arXiv preprint arXiv:2303.03032* (2023).
24. C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, in *Proceedings of the ACL-04 Workshop, volume 8* (2004) pp. 1–8.
25. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, Microsoft coco: Common objects in context, in *European Conference on Computer Vision* (2014) pp. 740–755.
26. S. Liu, Z. Zhu, N. Ye, S. Guadarrama and K. Murphy, Optimization of image description metrics using policy gradient methods, *arXiv preprint arXiv:1612.00370* **5** (2016).
27. B. Makav and V. Kılıç, A new image captioning approach for visually impaired people, in *11th International Conference on Electrical and Electronics Engineering* (2019) pp. 945–949.
28. B. Makav and V. Kılıç, Smartphone-based image captioning for visually and hearing impaired, in *11th International Conference on Electrical and Electronics Engineering* (2019) pp. 950–953.
29. J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang and A. Yuille, Deep captioning with multimodal recurrent neural networks (m-rnn), *arXiv preprint arXiv:1412.6632* (2014).
30. A. Mathews, L. Xie and X. He, Semstyle: Learning to generate stylised image captions using unaligned text, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018) pp. 8591–8600.
31. D. W. Otter, J. R. Medina and J. K. Kalita, A survey of the usages of deep learning for natural language processing, *IEEE Transactions on Neural Networks and Learning Systems* **32**(2) (2020) 604–624.
32. K. Papineni, S. Roukos, T. Ward and W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (2002) pp. 311–318.
33. R. Pascanu, T. Mikolov and Y. Bengio, On the difficulty of training recurrent neural networks, in *International Conference on Machine Learning* (2013) pp. 1310–1318.
34. B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier and S. Lazebnik, Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models, in *Proceedings of the IEEE International Conference on Computer Vision* (2015) pp. 2641–2649.
35. R. Staniūtė and D. Šešok, A systematic literature review on image captioning, *Applied Sciences* **9**(10) (2019) p. 2024.
36. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, Rethinking the inception architecture for computer vision, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016) pp. 2818–2826.
37. Y. H. Tan and C. S. Chan, Phrase-based image caption generator with hierarchical lstm network, *Neurocomputing* **333** (2019) 86–100.
38. M. Tanti, A. Gatt and K. P. Camilleri, Where to put the image in an image caption generator, *Natural Language Engineering* **24**(3) (2018) 467–489.
39. Y. Tewel, Y. Shalev, R. Nadler, I. Schwartz and L. Wolf, Zero-shot video captioning with evolving pseudo-tokens, *arXiv preprint arXiv:2207.11100* (2022).
40. Y. Tewel, Y. Shalev, I. Schwartz and L. Wolf, Zerocap: Zero-shot image-to-text generation for visual-semantic arithmetic, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022) pp. 17918–17928.

41. K. Umemura, M. A. Kastner, I. Ide, Y. Kawanishi, T. Hirayama, K. Doman, D. Deguchi and H. Murase, Tell as you imagine: Sentence imageability-aware image captioning, in *International Conference on Multimedia Modeling* (2021) pp. 62–73.
42. B. Uslu, Ö. Çaylı, V. Kılıç and A. Onan, Resnet based deep gated recurrent unit for image captioning on smartphone, *European Journal of Science and Technology* (35) (2022) 610–615.
43. R. Vedantam, C. Lawrence Zitnick and D. Parikh, Cider: Consensus-based image description evaluation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015) pp. 4566–4575.
44. O. Vinyals, A. Toshev, S. Bengio and D. Erhan, Show and tell: A neural image caption generator, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015) pp. 3156–3164.
45. H. Wang, H. Wang and K. Xu, Evolutionary recurrent neural network for image captioning, *Neurocomputing* (2020).
46. Q. Wang and A. B. Chan, Cnn+ cnn: Convolutional decoders for image captioning, *arXiv preprint arXiv:1805.09019* (2018).
47. K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel and Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, in *International Conference on Machine Learning* (2015) pp. 2048–2057.
48. Q. You, H. Jin and J. Luo, Image captioning at will: A versatile scheme for effectively injecting sentiments into image descriptions, *arXiv preprint arXiv:1801.10121* (2018).
49. Z. Zeng, Y. Xie, H. Zhang, C. Chen, Z. Wang and B. Chen, Meacap: Memory-augmented zero-shot image captioning, *arXiv preprint arXiv:2403.03715* (2024).
50. Z. Zeng, H. Zhang, R. Lu, D. Wang, B. Chen and Z. Wang, Conzic: Controllable zero-shot image captioning by sampling-based polishing, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023) pp. 23465–23476.
51. L. Zhang, S. Wang and B. Liu, Deep learning for sentiment analysis: a survey, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **8**(4) (2018) 1–25.
52. B. Zoph, V. Vasudevan, J. Shlens and Q. V. Le, Learning transferable architectures for scalable image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018) pp. 8697–8710.

Appendix A: Supercategory Results

Table A1. Performance metrics across various supercategories for a 3-layered GRU model on a new dataset, showcasing BLEU, METEOR, ROUGE-L, CIDEr, and SPICE scores.

Supercategory	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE
electronic	0.7087	0.5430	0.4010	0.2945	0.2480	0.5257	0.7578	0.1709
vehicle	0.6539	0.4645	0.3211	0.2252	0.2205	0.4793	0.7122	0.1582
outdoor	0.6753	0.4938	0.3531	0.2507	0.2317	0.4955	0.7424	0.1626
food	0.6679	0.4817	0.3426	0.2434	0.2211	0.4858	0.6295	0.1623
sports	0.7106	0.5321	0.3815	0.2743	0.2552	0.5264	0.7439	0.1769
indoor	0.6791	0.5100	0.3723	0.2680	0.2274	0.5064	0.7338	0.1567
person	0.6751	0.4930	0.3513	0.2516	0.2348	0.4977	0.7844	0.1638
animal	0.6701	0.4967	0.3606	0.2586	0.2360	0.5067	0.8358	0.1706
appliance	0.7205	0.5613	0.4098	0.2964	0.2362	0.5338	0.6642	0.1572
kitchen	0.6956	0.5217	0.3814	0.2778	0.2340	0.5109	0.7635	0.1688
accessory	0.6401	0.4573	0.3235	0.2327	0.2164	0.4715	0.6618	0.1489
furniture	0.7036	0.5348	0.3951	0.2904	0.2379	0.5226	0.8099	0.1659

Table A2. Performance metrics across various supercategories for a 6-layered GRU model on a new dataset, showcasing BLEU, METEOR, ROUGE-L, CIDEr, and SPICE scores.

Supercategory	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE
electronic	0.7191	0.5599	0.4216	0.3173	0.2477	0.5315	0.7792	0.1706
vehicle	0.6628	0.4741	0.3228	0.2200	0.2192	0.4774	0.6874	0.1522
outdoor	0.6789	0.4906	0.3461	0.2393	0.2269	0.4948	0.7153	0.1594
food	0.6762	0.4905	0.3463	0.2452	0.2143	0.4849	0.6093	0.1584
sports	0.7264	0.5484	0.4029	0.2941	0.2613	0.5318	0.7677	0.1738
indoor	0.7047	0.5351	0.3952	0.2867	0.2296	0.5159	0.7639	0.1574
person	0.6824	0.5011	0.3605	0.2609	0.2342	0.4972	0.7785	0.1595
animal	0.6826	0.5130	0.3741	0.2692	0.2391	0.5098	0.8522	0.1699
appliance	0.7276	0.5689	0.4114	0.2930	0.2315	0.5416	0.6439	0.1597
kitchen	0.7025	0.5295	0.3896	0.2831	0.2289	0.5122	0.7437	0.1665
accessory	0.6525	0.4703	0.3344	0.2383	0.2143	0.4737	0.6534	0.1444
furniture	0.7102	0.5406	0.3994	0.2928	0.2337	0.5207	0.7922	0.1631

Table A3. Performance metrics across various supercategories for a 12-layered GRU model on a new dataset, showcasing BLEU, METEOR, ROUGE-L, CIDEr, and SPICE scores.

Supercategory	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**	**METEOR**	**ROUGE-L**	**CIDEr**	**SPICE**
electronic	0.7094	0.5487	0.4075	0.3001	0.2393	0.5256	0.7452	0.1665
vehicle	0.6724	0.4806	0.3286	0.2216	0.2214	0.4863	0.7026	0.1583
outdoor	0.6771	0.4976	0.3582	0.2562	0.2290	0.4984	0.7350	0.1598
food	0.6795	0.4925	0.3420	0.2339	0.2102	0.4843	0.5861	0.1545
sports	0.7325	0.5632	0.4178	0.3085	0.2642	0.5393	0.8008	0.1845
indoor	0.7081	0.5423	0.3990	0.2860	0.2289	0.5161	0.7667	0.1602
person	0.6887	0.5089	0.3640	0.2604	0.2337	0.5020	0.7919	0.1622
animal	0.6817	0.5079	0.3645	0.2570	0.2325	0.5047	0.8330	0.1730
appliance	0.7120	0.5574	0.4014	0.2838	0.2247	0.5366	0.5835	0.1527
kitchen	0.7023	0.5292	0.3859	0.2765	0.2257	0.5109	0.7194	0.1641
accessory	0.6606	0.4767	0.3330	0.2321	0.2130	0.4776	0.6515	0.1479
furniture	0.7119	0.5435	0.3988	0.2873	0.2305	0.5221	0.7682	0.1601

Table A4. Performance metrics across various supercategories for a 15-layered GRU model on a new dataset, showcasing BLEU, METEOR, ROUGE-L, CIDEr, and SPICE scores.

Supercategory	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**	**METEOR**	**ROUGE-L**	**CIDEr**	**SPICE**
electronic	0.6328	0.4365	0.2739	0.1698	0.1873	0.4709	0.4297	0.1250
vehicle	0.6106	0.3924	0.2169	0.1168	0.1759	0.4442	0.4063	0.1234
outdoor	0.6179	0.4043	0.2395	0.1305	0.1825	0.4562	0.4160	0.1307
food	0.5881	0.3584	0.1972	0.1031	0.1573	0.4347	0.3183	0.1008
sports	0.6753	0.4666	0.2844	0.1628	0.2088	0.5007	0.4787	0.1486
indoor	0.6110	0.4155	0.2576	0.1570	0.1762	0.4578	0.4300	0.1164
person	0.6278	0.4155	0.2465	0.1417	0.1862	0.4663	0.4543	0.1299
animal	0.6268	0.4198	0.2467	0.1362	0.1875	0.4701	0.4944	0.1372
appliance	0.6458	0.4726	0.3138	0.2033	0.1861	0.4910	0.3962	0.1156
kitchen	0.6146	0.4060	0.2474	0.1467	0.1707	0.4580	0.4049	0.1116
accessory	0.6089	0.3971	0.2391	0.1432	0.1714	0.4464	0.3882	0.1215
furniture	0.6376	0.4401	0.2782	0.1732	0.1844	0.4772	0.4664	0.1219

Appendix B: Tukey Results

Table B1. Tukey significance test results for different model layers (Metric: BLEU-1)

Group 1	Group 2	Mean Diff.	P-Adj	Lower	Upper	Reject
12-layer	15-layer	-0.0699	0.0000	-0.0962	-0.0436	True
12-layer	3-layer	-0.0113	0.7438	-0.0376	0.0150	False
12-layer	6-layer	-0.0009	1.0000	-0.0271	0.0254	False
12-layer	9-layer	0.0251	0.0682	-0.0012	0.0514	False
15-layer	3-layer	0.0586	0.0000	0.0323	0.0849	True
15-layer	6-layer	0.0691	0.0000	0.0428	0.0953	True
15-layer	9-layer	0.0950	0.0000	0.0687	0.1213	True
3-layer	6-layer	0.0104	0.7948	-0.0158	0.0367	False
3-layer	9-layer	0.0364	0.0023	0.0101	0.0627	True
6-layer	9-layer	0.0260	0.0547	-0.0003	0.0522	False

Table B2. Tukey significance test results for different model layers (Metric: BLEU-2)

Group 1	Group 2	Mean Diff.	P-Adj	Lower	Upper	Reject
12-layer	15-layer	-0.1020	0.0000	-0.1380	-0.0659	True
12-layer	3-layer	-0.0132	0.8388	-0.0493	0.0229	False
12-layer	6-layer	-0.0022	0.9998	-0.0383	0.0339	False
12-layer	9-layer	0.0292	0.1647	-0.0068	0.0653	False
15-layer	3-layer	0.0888	0.0000	0.0527	0.1248	True
15-layer	6-layer	0.0998	0.0000	0.0637	0.1358	True
15-layer	9-layer	0.1312	0.0000	0.0952	0.1673	True
3-layer	6-layer	0.0110	0.9099	-0.0251	0.0471	False
3-layer	9-layer	0.0425	0.0133	0.0064	0.0785	True
6-layer	9-layer	0.0315	0.1151	-0.0046	0.0675	False

Table B3. Tukey significance test results for different model layers (Metric: BLEU-3)

Group 1	Group 2	Mean Diff.	P-Adj	Lower	Upper	Reject
12-layer	15-layer	-0.1216	0.0000	-0.1562	-0.087	True
12-layer	3-layer	-0.0089	0.9487	-0.0436	0.0257	False
12-layer	6-layer	0.0003	1.0000	-0.0343	0.0349	False
12-layer	9-layer	0.0316	0.0898	-0.0030	0.0662	False
15-layer	3-layer	0.1127	0.0000	0.0781	0.1473	True
15-layer	6-layer	0.1219	0.0000	0.0873	0.1565	True
15-layer	9-layer	0.1532	0.0000	0.1186	0.1878	True
3-layer	6-layer	0.0092	0.9425	-0.0254	0.0439	False
3-layer	9-layer	0.0405	0.0140	0.0059	0.0751	True
6-layer	9-layer	0.0313	0.0949	-0.0033	0.0659	False

Table B4. Tukey significance test results for different model layers (Metric: BLEU-4)

Group 1	Group 2	Mean Diff.	P-Adj	Lower	Upper	Reject
12-layer	15-layer	-0.1183	0.0000	-0.1488	-0.0877	True
12-layer	3-layer	-0.0033	0.9980	-0.0339	0.0272	False
12-layer	6-layer	0.0030	0.9986	-0.0275	0.0336	False
12-layer	9-layer	0.0337	0.0238	0.0031	0.0643	True
15-layer	3-layer	0.1149	0.0000	0.0844	0.1455	True
15-layer	6-layer	0.1213	0.0000	0.0907	0.1519	True
15-layer	9-layer	0.1519	0.0000	0.1214	0.1825	True
3-layer	6-layer	0.0064	0.9765	-0.0242	0.0369	False
3-layer	9-layer	0.0370	0.0102	0.0064	0.0676	True
6-layer	9-layer	0.0306	0.0049	0.0001	0.0612	True

Table B5. Tukey significance test results for different model layers (Metric: METEOR)

Group 1	Group 2	Mean Diff.	P-Adj	Lower	Upper	Reject
12-layer	15-layer	-0.0482	0.0000	-0.0624	-0.034	True
12-layer	3-layer	0.0038	0.9402	-0.0104	0.0181	False
12-layer	6-layer	0.0023	0.9908	-0.0119	0.0165	False
12-layer	9-layer	0.0218	0.0006	0.0076	0.036	True
15-layer	3-layer	0.0521	0.0000	0.0379	0.0663	True
15-layer	6-layer	0.0505	0.0000	0.0363	0.0647	True
15-layer	9-layer	0.0700	0.0000	0.0558	0.0842	True
3-layer	6-layer	-0.0015	0.9990	-0.0158	0.0127	False
3-layer	9-layer	0.0180	0.0066	0.0037	0.0322	True
6-layer	9-layer	0.0195	0.0026	0.0053	0.0337	True

Table B6. Tukey significance test results for different model layers (Metric: ROUGE-L)

Group 1	Group 2	Mean Diff.	P-Adj	Lower	Upper	Reject
12-layer	15-layer	-0.0442	0.0000	-0.0667	-0.0217	True
12-layer	3-layer	-0.0035	0.9924	-0.0260	0.0191	False
12-layer	6-layer	-0.0010	0.9999	-0.0236	0.0215	False
12-layer	9-layer	0.0216	0.0664	-0.0009	0.0441	False
15-layer	3-layer	0.0407	0.0000	0.0182	0.0633	True
15-layer	6-layer	0.0432	0.0000	0.0206	0.0657	True
15-layer	9-layer	0.0658	0.0000	0.0433	0.0883	True
3-layer	6-layer	0.0024	0.9981	-0.0201	0.025	False
3-layer	9-layer	0.0251	0.0221	0.0025	0.0476	True
6-layer	9-layer	0.0226	0.0485	0.0001	0.0452	True

Table B7. Tukey significance test results for different model layers (Metric: SPICE)

Group 1	Group 2	Mean Diff.	P-Adj	Lower	Upper	Reject
12-layer	15-layer	-0.0384	0.0000	-0.0495	-0.0273	True
12-layer	3-layer	0.0016	0.9944	-0.0095	0.0127	False
12-layer	6-layer	-0.0007	0.9997	-0.0119	0.0104	False
12-layer	9-layer	0.0200	0.0000	0.0089	0.0311	True
15-layer	3-layer	0.0400	0.0000	0.0289	0.0511	True
15-layer	6-layer	0.0377	0.0000	0.0266	0.0488	True
15-layer	9-layer	0.0584	0.0000	0.0473	0.0695	True
3-layer	6-layer	-0.0023	0.9976	-0.0134	0.0088	False
3-layer	9-layer	0.0184	0.0002	0.0073	0.0295	True
6-layer	9-layer	0.0207	0.0000	0.0096	0.0318	True