Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.053245

ARTICLE

Image Captioning Using Multimodal Deep Learning Approach

Rihem Farkh[1,*], Ghislain Oudinet[1] and Yasser Foued[2]

[1]Institut Supérieur de l'Electronique et du Numérique Méditerranée, ISEN Méditerranée, Toulon, 83000, France

[2]College of Applied Engineering, Muzahimiyah Branch, King Saud University, Riyadh, 11421, Saudi Arabia

*Corresponding Author: Rihem Farkh. Email: rihem.farkh@yncrea.fr

Received: 28 April 2024 Accepted: 01 August 2024 Published: 19 December 2024

ABSTRACT

The process of generating descriptive captions for images has witnessed significant advancements in last years, owing to the progress in deep learning techniques. Despite significant advancements, the task of thoroughly grasping image content and producing coherent, contextually relevant captions continues to pose a substantial challenge. In this paper, we introduce a novel multimodal method for image captioning by integrating three powerful deep learning architectures: YOLOv8 (You Only Look Once) for robust object detection, EfficientNetB7 for efficient feature extraction, and Transformers for effective sequence modeling. Our proposed model combines the strengths of YOLOv8 in detecting objects, the superior feature representation capabilities of EfficientNetB7, and the contextual understanding and sequential generation abilities of Transformers. We conduct extensive experiments on standard benchmark datasets to evaluate the effectiveness of our approach, demonstrating its ability to generate informative and semantically rich captions for diverse images. The experimental results showcase the synergistic benefits of integrating YOLOv8, EfficientNetB7, and Transformers in advancing the state-of-the-art in image captioning tasks. The proposed multimodal approach has yielded impressive outcomes, generating informative and semantically rich captions for a diverse range of images. By combining the strengths of YOLOv8, EfficientNetB7, and Transformers, the model has achieved state-of-the-art results in image captioning tasks. The significance of this approach lies in its ability to address the challenging task of generating coherent and contextually relevant captions while achieving a comprehensive understanding of image content. The integration of three powerful deep learning architectures demonstrates the synergistic benefits of multimodal fusion in advancing the state-of-the-art in image captioning. Furthermore, this approach has a profound impact on the field, opening up new avenues for research in multimodal deep learning and paving the way for more sophisticated and context-aware image captioning systems. These systems have the potential to make significant contributions to various fields, encompassing human-computer interaction, computer vision and natural language processing.

KEYWORDS

Image caption; multimodel methods; YOLOv8; efficientNetB7; features extration; Transformers; encoder; decoder; Flickr8k

1 Introduction

Generating descriptive captions for images is a complex task that pushes the boundaries of both computer vision and natural language processing (NLP) that involves generating textual descriptions for images automatically. The objective is to build a system capable of interpreting images, bridging the gap between visual perception and comprehension, and generating a relevant and coherent description in natural language. The growing field of image captioning offers exciting possibilities for assistive technologies, enabling visually impaired individuals to access image information; powering advanced content-based image retrieval systems; and enhancing human-computer interaction [1].

The task of image captioning is multidisciplinary, combining techniques from computer vision, deep learning, and natural language processing. It requires models to extract meaningful features from images and then generates grammatically correct and semantically meaningful sentences to describe them. Key components of an image captioning system include an image encoder, a text decoder, and an attention mechanism [2]. By employing an attention mechanism, the model can selectively focus on specific parts of the image, leading to more accurate and contextually relevant captions [3].

2 Related Work

Traditionally, image feature extraction relied on handcrafted methods. These involved using mathematical tools like SIFT, LBPs, and HOGs [4,5] to capture spatial and textural information from images. Classifiers like SVMs, AdaBoost, random forests, and CRFs [6] were then trained on these features to make predictions pixel-by-pixel. However, this approach has limitations. Manually designing and implementing features is a time-consuming and subjective process, often leading to biases and poor performance on unseen data.

Early image captioning approaches can be categorized into two main types: template-based methods and composition-based methods [7]. Template-based methods involved creating caption templates with placeholders, which were then filled in using information extracted from object detection, attribute classification, and scene recognition. In contrast, composition-based methods leveraged existing image-caption databases to identify relevant caption components and combine them to generate new descriptions. The swift progress of computer technology has empowered deep learning methods to excel in image processing, particularly in tasks like classification and detection.

Recent advances in image captioning have been driven by the development of deep learning models [8], particularly convolutional neural networks (CNNs) [9] and recurrent neural networks (RNNs) [10]. These models have been shown to be effective in extracting features from images and generating captions that are both accurate and fluent. Nevertheless, the RNN's challenges in gradient propagation prompted the integration of Long Short-Term Memory (LSTM) [11]. LSTMs were employed to interpret extracted image features, transforming them into captions by generating a cohesive string of words [12].

A significant milestone in the field of image captioning was the incorporation of attention mechanisms into the encoder-decoder architecture [13]. Inspired by their effectiveness in machine translation and object detection, attention models empowered captioning systems to selectively focus on pertinent aspects of input images during caption generation. This breakthrough not only substantially enhanced the quality of generated captions but also became a standard component in numerous state-of-the-art image captioning models [14].

The utility of attention mechanisms transcends various captioning models, as evidenced by their widespread adoption [15–17]. The advent of attention mechanisms laid the groundwork for the emergence of self-attention, which subsequently facilitated the development of multi-head attention within the Transformer model. The Transformer model revolutionized the abstraction of multi-head self-attention operations into a unified module, enabling the stacking of these modules to attain the requisite non-linearity and representational capacity for modeling intricate functions. Originally applied to machine translation, these advancements were later extended to the realm of image processing. Since its inception in 2017 [18], the Transformer model has emerged as a cornerstone framework underpinning many contemporary state-of-the-art models [19–21]. The Attention on Attention (AoA) mechanism has shown significant promise in the field of image captioning, contributing to advancements in how visual features are processed and used to generate textual descriptions of images [22].

In the following Table 1, we present the advantages and limitations of existing practices.

Table 1: Comparison of image captioning methods

Method	Description	Advantages	Limitations
Encoder-decoder architectures [23]	Use CNN as encoder to extract image features and RNN as decoder to generate captions.	Effective for generating coherent captions.	May struggle with long captions and context.
Attention mechanisms [24]	Focus on specific regions of the image while generating captions.	Improves caption quality by highlighting relevant objects.	Can be computationally expensive.
Multimodal fusion [25]	Combine different deep learning architectures to leverage their strengths.	Enhances caption quality by integrating multiple modalities.	Requires large datasets and computational resources.
Vision-language pre-training [26]	Pre-train models on large datasets to learn joint representations of images and text.	Improves caption quality by learning shared representations.	Requires large datasets and computational resources.
Soft attention [27]	Use weighted sum of image features to generate captions.	Effective for generating coherent captions.	May struggle with long captions and context.
Hard attention [28]	Select specific regions of the image to generate captions.	Improves caption quality by highlighting relevant objects.	Can be computationally expensive.
Dual attention mechanism [29]	Use two attention mechanisms to focus on different aspects of the image.	Improves caption quality by highlighting relevant objects.	Can be computationally expensive.

(Continued)

Table 1 (continued)

Method	Description	Advantages	Limitations
Auto-encoding scene graphs [30]	Use scene graphs to represent image content and generate captions.	Effective for generating coherent captions.	Requires large datasets and computational resources.

In light of the limitations associated with individual models, researchers have embraced ensemble learning as a means to harness the diverse strengths of multiple models. By combining several individual models and implementing specific strategies, multimodel learning aims to enhance the overall model's capacity for generalization. This approach has demonstrated effectiveness in mitigating the aforementioned challenges [31].

The research gap in image captioning lies in the limitations of existing methods to generate coherent, contextually relevant, and semantically rich captions. The proposed solution addresses these limitations by integrating three powerful deep learning architectures: YOLOv8 (You Only Look Once) for object detection, EfficientNetB7 for feature extraction, and Transformers for sequence modeling. YOLOv8 is a state-of-the-art object detection model that can accurately detect objects in images, addressing the limitation of incomplete object detection. EfficientNetB7 is a powerful feature extraction model that can extract robust and informative features from images, enhancing the model's ability to understand the context of the image. Transformers are well-suited for sequence modeling tasks like image captioning, enabling the model to generate coherent and contextually relevant captions. By combining these three architectures, the proposed solution can generate high-quality captions that accurately describe the content of images.

The proposed solution has been validated through the use of the BLEU score on the Flickr8k dataset.

The Flickr8k dataset was selected for image captioning due to its realistic and diverse images, multiple captions per image, small and manageable size, well-established benchmark, availability, and accessibility. The dataset contains images with five captions each, providing a comprehensive evaluation of the model's performance. Its small size makes it easy to work with on a workstation, and its widespread use as a benchmark allows for easy comparison of different models. Additionally, the dataset requires pre-processing and feature extraction, making it suitable for various techniques and models.

The BLEU score is a widely used metric for evaluating the quality of machine-generated text, including image captions. It measures the similarity between the generated caption and the reference caption, with higher scores indicating better quality captions. The experimental results demonstrate the effectiveness of the proposed solution, achieving a BLEU score of 0.72, which indicates a high level of similarity between the generated captions and the reference captions. This validates the proposed solution as a reliable and accurate approach for image captioning.

3 Proposed Approach

Multimodal fusion for image captioning involves integrating information from various modalities, such as visual and textual data, to generate coherent and contextually accurate descriptions of images. This approach leverages the complementary strengths of different models to enhance the performance of image captioning system.

Image captioning models can be built by combining the strengths of CNNs and Transformers (Fig. 1). This figure illustrates a multimodal image captioning model that integrates features from EfficientNetB7 and YOLOv8 with a Transformer-based encoder-decoder architecture to generate descriptive captions for images. First, images are preprocessed and fed into a pre-trained EfficientNetB7 [32] and YOLOv8 [33,34], which extract visual features. Then, a Transformer decoder takes over. It receives the encoded image features and starts generating captions word by word. The Transformer attends to relevant parts of the image features based on the caption being built, ensuring coherence between the image content and the emerging sentence. This back-and-forth process continues until the decoder predicts an "end" token, finalizing the caption. This approach leverages CNNs' ability to capture visual details and the Transformer's talent for handling sequential data like text, resulting in accurate and descriptive captions.

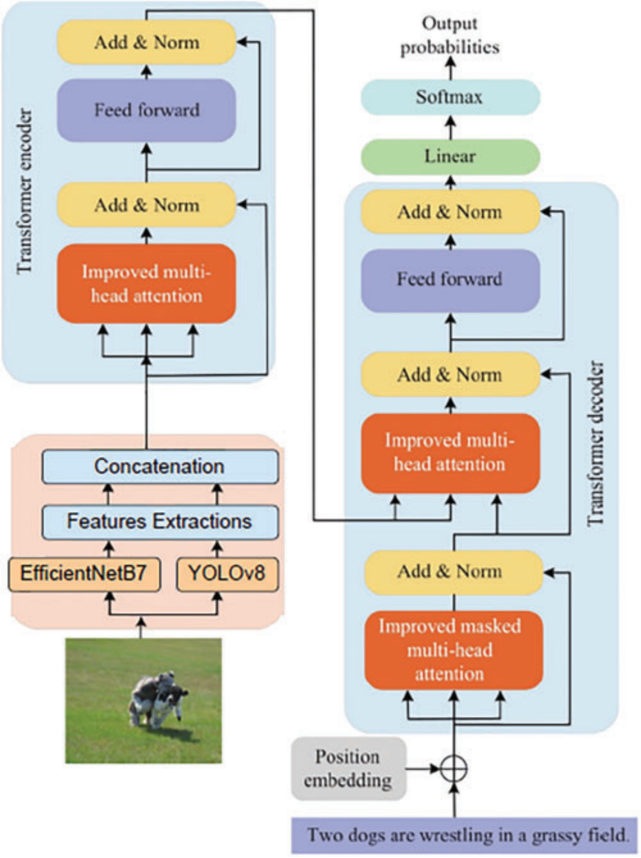


Figure 1: Overall architecture of the system

The encoder and decoder are built upon a series of Transformer layers, enabling them to effectively process both image regions and words. The decoder layer consists of three key components: a self-attention layer, a cross-attention layer, and a feed-forward layer. Each of these layers is followed by an "add-norm" operation, which combines the layer's output with the input, allowing for efficient processing. Finally, the decoder's output is linearly transformed and passed through a softmax function to produce probabilities for the next word in the generated caption.

The label (caption) is used as an input to the model during the training phase to help it learn and establish the relationship between image features and textual descriptions. During inference, the decoder will use the previously generated words as input to predict the next word in the sequence.

4 Dataset for Image Caption Generator

The Flickr8k dataset [35] comprises 8091 images, each accompanied by five English captions. Available on the Kaggle website, this dataset has a size of 1 GB. It includes over 31,000 images, with each image featuring five reference sentences, meticulously provided by human annotators. The core of the dataset is 'Flickr8k.token'. This file links each image name in the 'Flickr8k_Dataset' folder (containing 8091 images) to its corresponding captions stored in the separate 'Flickr_8k_text' folder.

The distribution of training, validation, and testing datasets is shown in Table 2.

Table 2: Dataset distribution

Flickr8k dataset	Total
Training dataset	6000
Validation dataset	1000
Test dataset	1000

4.1 Parameters

The following parameters are essential for configuring the model architecture, and training process in an image-captioning pipeline. Adjusting these values can affect the performance and behavior of the image-captioning model. We specify the vocabulary size VOCAB_SIZE, the sequence length SEQ_LENGTH, the embedding dimension EMBED_DIM, and the number of units in the feed-forward network FF_DIM. These parameters will influence the model's performance and computational efficiency.

- VOCAB_SIZE: Size of the vocabulary, indicating the number of unique words in the captions: 3000–10,000, which is a reasonable range to capture the majority of words in the dataset.

For example, a simpler model like a CNN-LSTM might require a smaller vocabulary size, while a more complex model like a Transformer-based architecture might benefit from a larger vocabulary size. Here are some possible VOCAB_SIZE values for the Flickr8k dataset: Small vocabulary: 3000 to 4000 [36]; Medium vocabulary: 5000 to 6000 [37] and Large vocabulary: 8000 to 10,000 [38].

- SEQ_LENGTH: Maximum length allowed for any sequence, typically representing the maximum number of tokens in a caption: 10–30, which is the average length of the captions.

Here are some possible SEQ_LENGTH values for the Flickr8k dataset: Short sequence: 10–15 [36]; Medium sequence: 15–20 [37]; Long sequence: 25–30 [38].

- EMBED_DIM: Dimensionality of the image embeddings and token embeddings: 128–512, which is a common range for embedding dimensions in natural language processing tasks.

Here are some possible EMBED_DIM values for the Flickr8k dataset: Small embedding: 128 [36]; Medium embedding: 256 [37]; Large embedding: 512 [38].

- FF_DIM: Number of units in the feed-forward network used in the Transformer layers: 2–4 times the EMBED_DIM, which is a common practice to set the feedforward dimension.

Here are some possible FF_DIM values for the Flickr8k dataset: Small feedforward: 512 [36]; Medium feedforward: 1024 [37]; Large feedforward: 2048 [38].

The choice of FF_DIM in a model should be carefully considered based on several factors. Firstly, larger FF_DIM values can potentially improve model performance by capturing more complex patterns in the data, but they also increase the risk of overfitting. Secondly, larger FF_DIM values require more computational resources and may result in longer training times, which can impact the model's scalability. Finally, smaller FF_DIM values may make the model more interpretable but could limit its ability to capture intricate relationships within the data.

- BATCH_SIZE: 64–128, which is a reasonable range to utilize GPU resources efficiently while keeping memory usage manageable.

4.2 Text Vectorization

Vectorizing text data involves converting textual input into numerical representations that machine-learning models can process. In the context we provide, the **Text Vectorization** layer is used for this purpose, available in the Keras library.

5 Components of Image Captioning

5.1 Image Encoder

The image encoder is responsible for extracting visual features from the input image. YOLOv8 and EfficientNetB7 are used as image encoders due to their ability to capture hierarchical features of images. An extra-large YOLOv8 backbone pretrained on COCO and EfficientNetB7 with weights pretrained on ImageNet are utilized in our approach. By incorporating CNNs pretrained on diverse datasets, we harness a broad range of feature representations learned across different domains, leading to a more comprehensive understanding of the input data. This approach also helps mitigate overfitting by leveraging the ensemble's ability to average out errors from individual models. Moreover, employing CNNs pretrained on distinct datasets enhances robustness by introducing greater variability in feature representations. Through the fusion of features from multiple models, our ensemble approach enhances generalization capabilities, enabling more effective adaptation to new, unseen data.

5.1.1 YOLOv8

The YOLOv8 architecture builds upon earlier versions of YOLO algorithms, utilizing a convolutional neural network divided into two key components: a backbone for feature extraction and a head for object detection. At its core, YOLOv8 adopts a modified version of the CSPDarknet53 [39] architecture as its backbone. This architecture integrates 53 convolutional layers and incorporates cross-stage partial connections, enhancing information flow between layers. The backbone serves as a pre-trained network tasked with extracting comprehensive feature representations from images (Fig. 2).

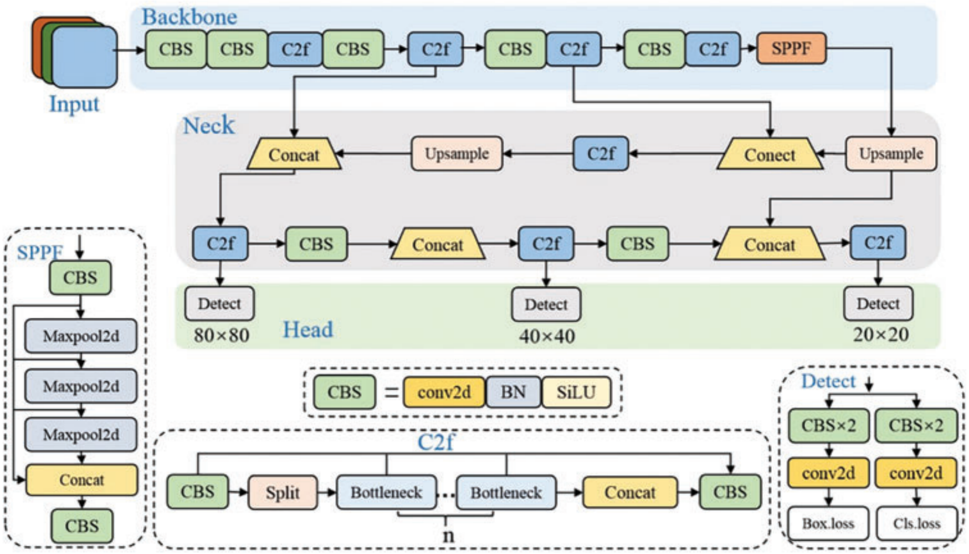


Figure 2: YOLOv8 architecture

CSPDarknet53 serves as both a convolutional neural network and a backbone for object detection, leveraging the DarkNet-53 architecture. YOLOv8 employs CSPNet to segment and merge the base layer's feature map via a cross-stage hierarchy. By adopting a split-and-merge strategy, it facilitates enhanced gradient flow throughout the network.

CSPDarknet53 functions as the backbone, contributing to reducing the spatial resolution of the image while concurrently augmenting its feature (channel) resolution. The head of YOLOv8, consisting of convolutional and fully connected layers, is responsible for predicting the location, confidence, and class of detected objects within an image [40].

The inclusion of self-attention in YOLOv8's head enables the model to adaptively weight the importance of different image features, prioritizing those most relevant to the detection task.

Furthermore, YOLOv8 demonstrates proficiency in multi-scaled object detection by employing a feature pyramid network. This network enables the model to detect objects of varying sizes and scales within an image, thereby enhancing its versatility and applicability in diverse scenarios.

To use only the backbone architecture of the YOLOv8 model (without the detection head), we can use **keras_cv.models.YOLOv8Backbone.from_preset()**.

This function allows to load a pre-trained YOLOv8 backbone directly into a Keras model object. It provides a convenient way to access the feature extraction capabilities of YOLOv8 for tasks other than object detection [41].

5.1.2 EfficientNet

EfficientNet is a family of convolutional neural network (CNN) architectures (Fig. 3) that have gained significant attention for their impressive performance and computational efficiency across various computer vision tasks. Introduced by Tan et al. [32], EfficientNet addresses the challenge of balancing model accuracy and computational resources by proposing a novel scaling method that uniformly scales network depth, width, and resolution. Here are some key characteristics and components of EfficientNet [42,43]:

1. Compound Scaling: EfficientNet employs compound scaling to balance model depth, width, and resolution. Traditional scaling methods focus on scaling only one dimension (e.g., depth), which often leads to suboptimal performance. Compound scaling uniformly scales the network's depth, width (number of channels), and resolution simultaneously using a compound coefficient φ. This approach ensures that the model's capacity increases efficiently across all dimensions, resulting in better performance.
2. Efficient Building Blocks: EfficientNet utilizes a novel building block called the MBConv (Mobile Inverted Bottleneck Convolution). The MBConv block consists of depthwise convolutions, followed by pointwise convolutions and squeeze-and-excitation (SE) blocks. This design maximizes computational efficiency while maintaining expressive power (Fig. 4).
3. Depthwise Separable Convolutions: EfficientNet relies heavily on depthwise separable convolutions, which decompose the standard convolution operation into separate depthwise and pointwise convolutions. This decomposition reduces computational complexity significantly while preserving representational capacity.
4. Squeeze-and-Excitation (SE) Blocks: SE blocks enhance feature representation by adaptively recalibrating channel-wise feature responses. They consist of a global average pooling layer followed by fully connected layers, which learn channel-wise scaling factors to emphasize informative features and suppress less relevant ones.
5. Model Variants: EfficientNet offers several model variants (e.g., EfficientNetB0 to EfficientNetB7) with varying depths and complexities. These variants are obtained by scaling the baseline network architecture according to the compound scaling method, enabling users to choose models that best suit their computational constraints and performance requirements.

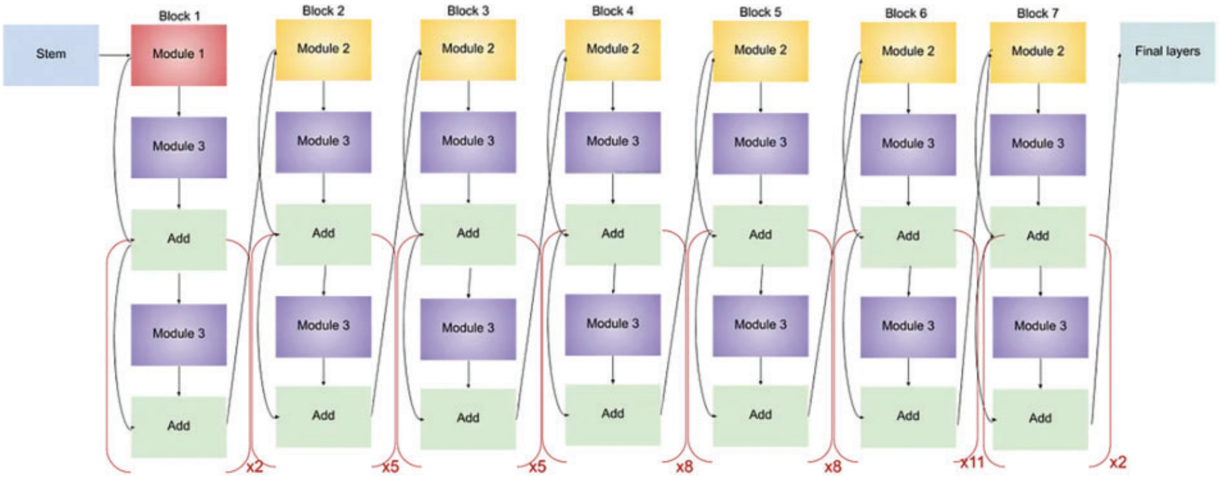


Figure 3: Architecture of EfficientNetB7

EfficientNet stands out as a highly effective and efficient architecture for a broad spectrum of computer vision tasks spanning from image classification and object detection to semantic segmentation.

Its innovative design principles and scalable architecture make it a popular choice for both research and practical applications.

EfficientNet comes in different sizes, with B0 being the smallest and B7 the largest. Each version (B0, B5, and B7) gets progressively bigger and more complex (more depth, width, and resolution) as you move up the scale. This complexity boost leads to improved accuracy, making B7 the most accurate model in the family. These EfficientNet models are built using seven key network blocks, each containing multiple modules [44].

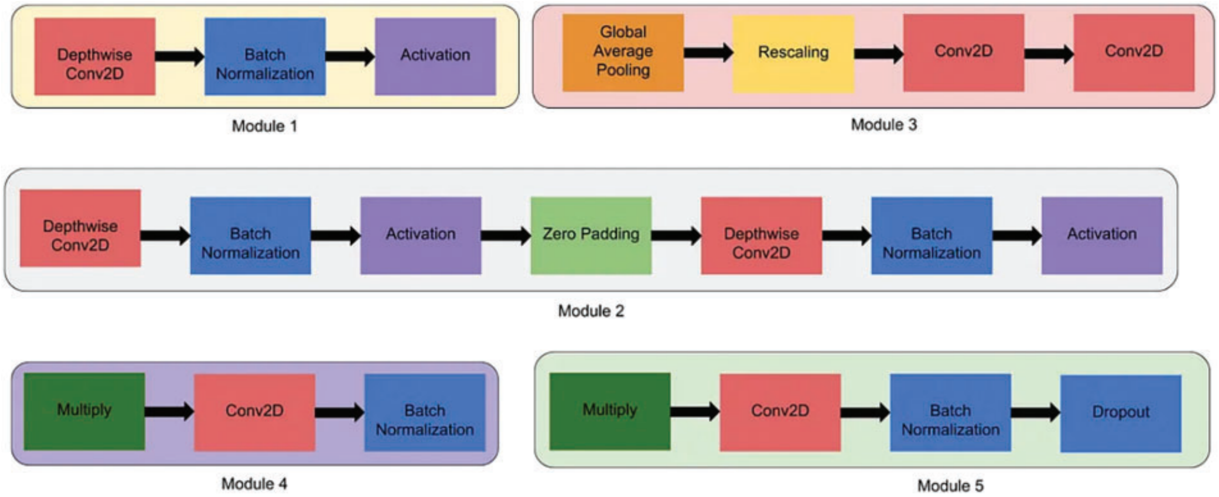


Figure 4: Modules used in our architecture

While all EfficientNet models (B0, B5, and B7) share the same starting (input) and ending (output) modules, the complexity lies within the seven central blocks. These blocks contain varying numbers of sub-blocks (modules) depending on the model version. B0, the simplest version, uses just three modules in its first and last blocks. B5 and B7, however, increase the complexity by employing six modules with repetitions within these blocks. This difference in module count translates to a significant jump in layer count, with B0 having 237 layers compared to B7's massive 813 layers [45].

5.2 *Transformer Encoder Decoder Architecture*

The Transformer architecture was first introduced in [18]. Since then, it has become the backbone of several leading models, including GPT-2 14 and BERT 15. The popularity of Transformer models stems from their versatility and ability to excel in various applications, such as language translation models and question-answer based models. One of the key advantages of Transformers is their adaptable nature, which enables them to be effectively applied to a wide range of use cases. Fig. 5 depicts the architecture of the Transformer. On the left side of the image, we observe the Encoder stack comprising N identical layers. Conversely, the right side showcases the decoder stack, also consisting of N identical layers.

The Multi-Head Attention layer is a crucial component in Transformer models, designed to capture complex relationships and dependencies within the input data. It operates by splitting the input into multiple heads and processing each head independently to extract different aspects of information. These heads allow the model to attend to different parts of the input simultaneously, enabling it to capture diverse patterns and features effectively. After processing, the outputs from all heads are concatenated and linearly transformed to generate the final attention output. This mechanism enhances the model's ability to learn rich representations and perform well on various tasks such as language modeling, translation, and image captioning [18].

In Transformers, attention is realized through the 'Scaled-Dot Product Attention' mechanism (Fig. 5). This entails creating Query (Q), Key (K), and Value (V) vectors for each word in the sentence [46]. These vectors can be described as follows:

- Q: The vector whose value is to be determined.
- K: The vector representing the features.
- V: The vector containing the actual input values.

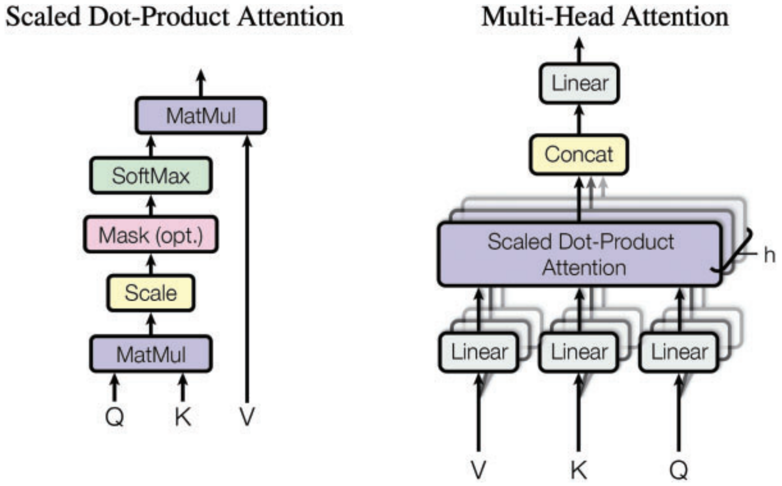


Figure 5: Formula for calculating Attention: $Atten\,(Q,K,V) = softmax\left(\frac{\mathrm{Qk^T}}{\sqrt{\mathrm{d_k}}}\right)\mathrm{v}$

A Multi-Head Attention (MHA) layer comprises 'h' self-attention heads. Each head computes its Scaled Dot Product Attention, and the resulting values are concatenated to yield the Multi-Head Attention output [47]. Employing multiple attention heads enhances the performance of the attention layer, enabling the model to concentrate on diverse positions within the input data.

Add & Norm Layer: The "Add & Norm" layer in a Transformer combines two fundamental components: residual connections and layer normalization [47].

- *Residual Connection (Addition)*: enables direct flow of input information to the output of each layer, mitigating the vanishing gradient problem and facilitating smoother gradient flow during training.
- *Layer Normalization (Norm):* normalizes the activations of each layer across the feature dimension, stabilizing the distribution of values and addressing issues like internal covariate shift.

6 Implementation

Our image captioning model processes 299 × 299 input images using a encoder to generate image vectors. These vectors are then fed into a Transformer encoder, which works in tandem with a Transformer decoder trained on corresponding captions. The encoder consists of a single multi-headed attention head and a normalization layer, while the decoder employs two multi-headed attention heads and three normalization layers. The models are built using TensorFlow Keras.

Table 3 presents the model parameters utilized in this study.

Table 3: Model parameter

Image size	(299, 299)
Maximum vocab size	15,000
Length of sequence	2%
Size of embedding	512
Size of batch	64
Used optimizer	Adam
Used loss function	Categorical-crossentropy

6.1 Proposed Pseudocode

We propose the following pseudocode for our approach (Fig. 6).

This figure outlines the main components and their interactions within our image-captioning model.

6.2 Results and Discussions

Our experimental results demonstrate that the multimodal model combining YOLOv8 and EfficientNetB7 outperforms individual models. Specifically, the YOLOv8_EfficientNetB7 model achieves superior performance compared to using EfficientNetB7 or YOLOv8 alone. Our evaluation metrics, including BLEU scores, indicate that the multimodal approach enhances captioning quality compared to using EfficientNetB7 alone. This highlights the effectiveness of leveraging multiple modalities for image captioning tasks, with YOLOv8 contributing to significant improvements in model performance. Table 4 presents a comparison between different models.

Fig. 7 illustrates some sample outputs from this work.

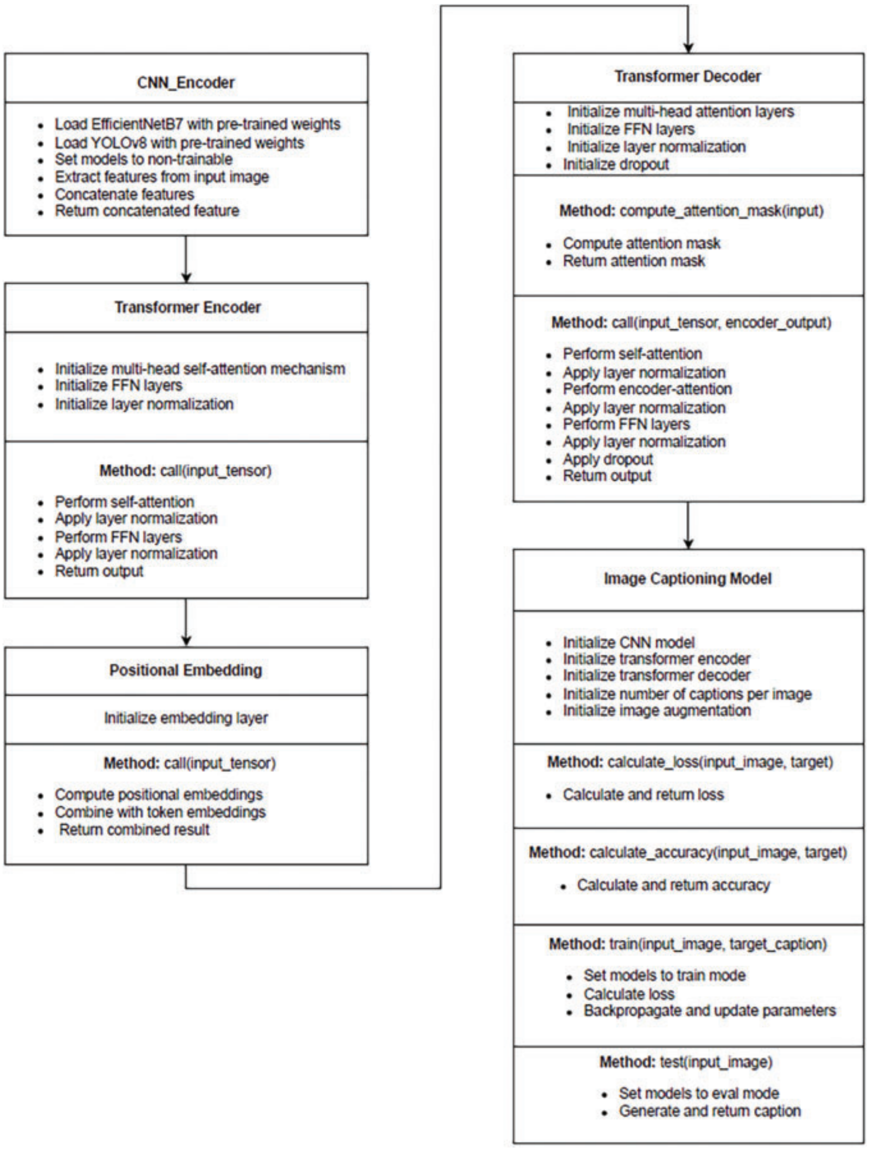

CNN_Encoder
Load EfficientNetB7 with pre-trained weights
Load YOLOv8 with pre-trained weights
Set models to non-trainable
Extract features from input image
Concatenate features
Return concatenated feature
Transformer Encoder
Initialize multi-head self-attention mechanism
Initialize FFN layers
Initialize layer normalization
Method: call(input_tensor)
Perform self-attention
Apply layer normalization
Perform FFN layers
Apply layer normalization
Return output
Positional Embedding
Initialize embedding layer
Method: call(input_tensor)
Compute positional embeddings
Combine with token embeddings
Return combined result
Transformer Decoder
Initialize multi-head attention layers
Initialize FFN layers
Initialize layer normalization
Initialize dropout
Method: compute_attention_mask(input)
Compute attention mask
Return attention mask
Method: call(input_tensor, encoder_output)
Perform self-attention
Apply layer normalization
Perform encoder-attention
Apply layer normalization
Perform FFN layers
Apply layer normalization
Apply dropout
Return output
Image Captioning Model
Initialize CNN model
Initialize transformer encoder
Initialize transformer decoder
Initialize number of captions per image
Initialize image augmentation
Method: calculate_loss(input_image, target)
Calculate and return loss
Method: calculate_accuracy(input_image, target)
Calculate and return accuracy
Method: train(input_image, target_caption)
Set models to train mode
Calculate loss
Backpropagate and update parameters
Method: test(input_image)
Set models to eval mode
Generate and return caption

Figure 6: Pseudocode for the proposed approach

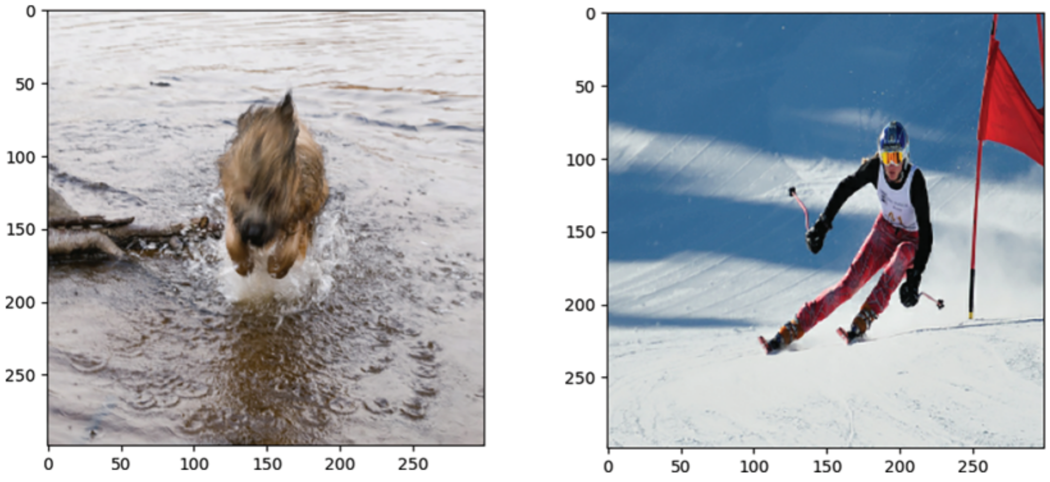
Table 4: Performance of different models

Models	B-1	B-2	B-3	B-4	BLEU with BP	Average	Std Dev
EfficientNetB7	0.67	0.67	0.66	0.65	0.596250	0.6625	0.009574
YOLOv8	0.73	0.72	0.71	0.7	0.6435	0.7150	0.012910
YOLOv8-EfficientNetB7	0.76	0.74	0.73	0.72	0.66375	0.7375	0.017078

(Continued)

Table 4 (continued)

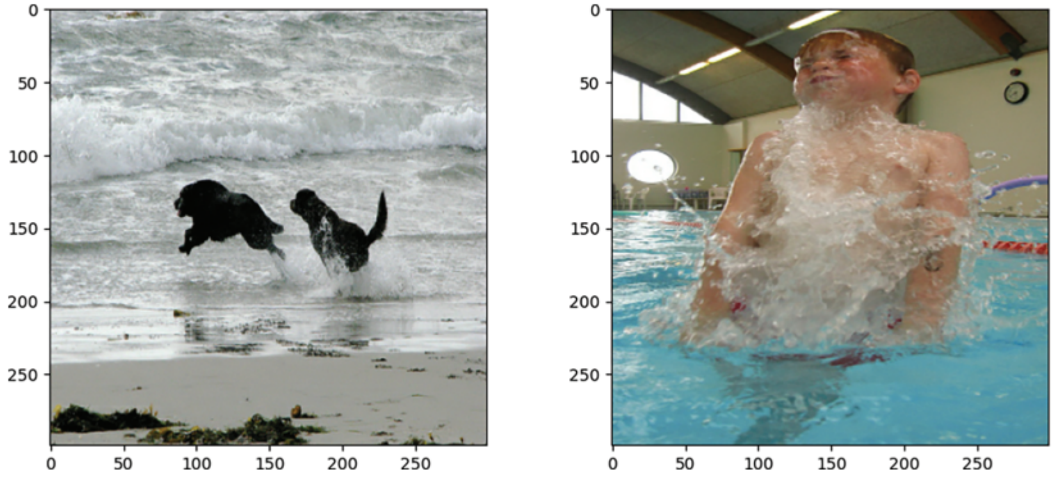
Models	B-1	B-2	B-3	B-4	BLEU with BP	Average	Std Dev
Inception [48]	0.51	0.28	0.2	0.1	0.245250	0.2725	0.174619
VGG16 [48]	0.49	0.27	0.19	0.09	0.234	0.26	0.170098
ResNet50 [48]	0.51	0.28	0.19	0.096	0.242100	0.269	0.177362
Xception [48]	0.513	0.282	0.2	0.098	0.245925	0.27325	0.176668
Soft-Attention [37]	0.667	0.434	0.288	0.197	0.356850	0.3965	0.205060
Hard-Attention [37]	0.669	0.439	0.296	0.199	0.360675	0.40075	0.204203
Log Bilinear [37]	0.6	0.38	0.254	0.171	0.316125	0.35125	0.186771
ResNet50–BERT [49]	0.532143			0.126316		NAN	NAN
Deep VS [50]	0.579	0.383	0.245	0.16	0.16	0.3417	0.1829
emb-gLSTM [51]	0.647	0.459	0.318	0.212	0.212	0.409	0.188
SCA-CNN [52]	0.682	0.496	0.359	0.258	0.258	0.488	0.183



Predicted Caption: a dog is running through the water

Predicted Caption: man playing skis on snow

Figure 7: (Continued)



Predicted Caption: two dogs run through the water

Predicted Caption: a young boy in a pool

Figure 7: Sample results obtained with our model

7 Conclusions

This paper presents a novel image-captioning methodology that leverages ensemble-learning techniques to integrate the strengths of three diverse deep learning models: Transformer encoder-decoder architecture, YOLOv8 for object detection, and EfficientNetB7 for feature extraction. Our novel approach, integrating these models, achieves state-of-the-art results in generating accurate and contextually rich image captions. Our experimental results demonstrate that our method generates high-quality captions that effectively capture the nuances of image content, achieving a BLEU score of 0.72, surpassing individual models. Our proposed methodology offers a promising direction for advancing the state-of-the-art in image captioning by harnessing the synergy of multiple deep learning architectures.

Acknowledgement: The authors sincerely acknowledge the support from ISEN Toulon and the Toulon Provence Méditerranée Regional Metropolitan Center (Energix Project).

Funding Statement: This research was funded by Researchers Supporting Project number (RSPD2024R698), King Saud University, Riyadh, Saudi Arabia.

Author Contributions: All authors contributed to this work as follows: Study conception and design: Rihem Farkh, Ghislain Oudinet; Data collection: Yasser Foued; Analysis and interpretation of results: Rihem Farkh, Ghislain Oudinet; Draft manuscript preparation: Rihem Farkh, Yasser Foued, Ghislain Oudinet. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] T. Baltrušaitis, C. Ahuja, and L. -P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 423–443, Feb. 1, 2019. doi: 10.1109/TPAMI.2018.2798607.

[2] R. Albano, L. Giusti, E. Maiorana, and P. Campisi, "Explainable vision transformers for vein biometric recognition," *IEEE Access*, vol. 12, pp. 60436–60446, 2024. doi: 10.1109/ACCESS.2024.

[3] J. Xia, X. Yang, Q. Ni, and D. Gao, "Research on image tibetan caption generation method fusion attention mechanism," in *IEEE 4th Int. Conf. Pattern Recognit. Mach. Learn. (PRML)*, Urumqi, China, 2023, pp. 193–198. doi: 10.1109/PRML59573.2023.10348351.

[4] K. Song, F. Zhu, and L. Song, "Moving target detection algorithm based on SIFT feature matching," in *Int. Conf. Front. Artif. Intell. Mach. Learn. (FAIML)*, Hangzhou, China, 2022, pp. 196–199. doi: 10.1109/FAIML57028.2022.00045.

[5] Y. Y. Sun, S. Chen, and L. Gao, "Feature extraction method based on improved linear LBP operator," in *IEEE 3rd Inf. Technol., Netw., Electronic Autom. Control Conf. (ITNEC)*, Chengdu, China, 2019, pp. 1536–1540. doi: 10.1109/ITNEC.2019.8729320.

[6] J. Zhang, Y. Liu, B. Wang, and C. Chen, "A SAR remote sensing image change detection method based on DR-UNet-CRF model," in *2022 IEEE Int. Conf. Smart Internet Things (SmartIoT)*, Suzhou, China, 2022, pp. 180–184. doi: 10.1109/SmartIoT55134.2022.00037.

[7] A. Z. Al-Jamal, M. J. Bani-Amer, and S. Aljawarneh, "Image captioning techniques: A review," in *Int. Conf. Eng. MIS (ICEMIS)*, Istanbul, Turkey, 2022, pp. 1–5. doi: 10.1109/ICEMIS56295.2022.9914173.

[8] J. Sudhakar, V. V. Iyer, and S. T. Sharmila, "Image caption generation using deep neural networks," in *Int. Conf. Adv. Technol. (ICONAT)*, Goa, India, 2022, pp. 1–3. doi: 10.1109/ICONAT53423.2022.9726074.

[9] V. Jayaswal, S. Ji, Satyankar, V. Singh, Y. Singh and V. Tiwari, "Image captioning using VGG-16 deep learning model," in *2nd Int. Conf. Disrupt. Technol. (ICDT)*, Greater Noida, India, 2024, pp. 1428–1433. doi: 10.1109/ICDT61202.2024.10489470.

[10] G. Hoxha, F. Melgani, and J. Slaghenauffi, "A new CNN-RNN framework for remote sensing image captioning," in *Med. Middle-East Geosci. Remote Sens. Symp. (M2GARSS)*, Tunis, Tunisia, 2020, pp. 1–4. doi: 10.1109/M2GARSS47143.2020.9105191.

[11] D. A. Navastara, D. B. Ansori, N. Suciati, and Z. F. Akbar, "Combination of DenseNet and BiLSTM model for Indonesian image captioning," in *Int. Conf. Adv. Mechatron., Intell. Manuf. Ind. Autom. (ICAMIMIA)*, Surabaya, Indonesia, 2023, pp. 994–999. doi: 10.1109/ICAMIMIA60881.2023.10427729.

[12] S. K. Satti, G. N. V. Rajareddy, P. Maddula, and N. V. Vishnumurthy Ravipati, "Image caption generation using ResNET-50 and LSTM," in *IEEE Silchar Subsection Conf. (SILCON)*, Silchar, India, 2023, pp. 1–6. doi: 10.1109/SILCON59133.2023.10404600.

[13] A. Kumari, A. Chauhan, and A. Singhal, "Vision 360: Image caption generation using encoder-decoder model," in *12th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence)*, Noida, India, 2022, pp. 312–317. doi: 10.1109/Confluence52989.2022.9734167.

[14] R. Ramos and B. Martins, "Using neural encoder-decoder models with continuous outputs for remote sensing image captioning," *IEEE Access*, vol. 10, pp. 24852–24863, 2022. doi: 10.1109/ACCESS.2022.3151874.

[15] A. Garg, D. Gowda, A. Kumar, K. Kim, M. Kumar and C. Kim, "Improved multi-stage training of online attention-based encoder-decoder models," in *IEEE Automat. Speech Recognit. Underst. Workshop (ASRU)*, Singapore, 2019, pp. 70–77. doi: 10.1109/ASRU46091.2019.9003936.

[16] F. Cong, W. Hu, Q. Huo, and L. Guo, "A comparative study of attention-based encoder-decoder approaches to natural scene text recognition," in *Int. Conf. Doc. Anal. Recognit. (ICDAR)*, Sydney, NSW, Australia, 2019, pp. 916–921. doi: 10.1109/ICDAR.2019.00151.

[17] X. Feng, L. Wang, and Y. Zhu, "Video summarization with self-attention based encoder-decoder framework," in *Int. Conf. Culture-Oriented Sci. Technol. (ICCST)*, Beijing, China, 2020, pp. 208–214. doi: 10.1109/ICCST50977.2020.00046.

[18] A. Vaswani *et al.*, "Attention is all you need," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 6000–6010, 2017.

[19] K. N. Lam, H. T. Nguyen, V. P. Mai, and J. Kalita, “Deep vision transformer and T5-based for image captioning,” in *RIVF Int. Conf. Comput. Commun. Technol. (RIVF)*, Hanoi, Vietnam, 2023, pp. 306–311. doi: 10.1109/RIVF60135.2023.10471815.

[20] C. Liu, R. Zhao, H. Chen, Z. Zou, and Z. Shi, “Remote sensing image change captioning with dual-branch transformers: A new method and a large scale dataset,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–20, 2022, Art. no. 5633520. doi: 10.1109/TGRS.2022.3218921.

[21] H. Kandala, S. Saha, B. Banerjee, and X. X. Zhu, “Exploring transformer and multilabel classification for remote sensing image captioning,” *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022, Art. no. 6514905. doi: 10.1109/LGRS.2022.3198234.

[22] L. Huang, W. Wang, J. Chen, and X. -Y. Wei, “Attention on attention for image captioning,” in *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, Republic of Korea, 2019, pp. 4633–4642. doi: 10.1109/ICCV.2019.00473.

[23] Z. Lu, H. Yang, and H. Zha, “Image captioning with semantic attention,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit., CVPR*, Las Vegas, NV, USA, Jun. 27–30, 2016, pp. 4651–4659.

[24] S. Kotyan and D. V. Vargas, “Improving robustness for vision transformer with a simple dynamic scanning augmentation,” *Neural Comput. Appl.*, vol. 36, no. 1, 2024, Art. no. 127000. doi: 10.1016/j.neucom.2023.127000.

[25] W. Guo, J. Wang, and S. Wang, “Deep multimodal representation learning: A survey,” *IEEE Access*, vol. 7, pp. 63373–63394, 2019. doi: 10.1109/ACCESS.2019.2916887.

[26] X. Hu *et al.*, “Scaling up vision-language pretraining for image captioning,” in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, 2022, pp. 17959–17968. doi: 10.1109/CVPR52688.2022.01745.

[27] Z. Lian, H. Li, R. Wang, and X. Hu, “Enhanced soft attention mechanism with an inception-like module for image captioning,” in *IEEE 32nd Int. Conf. Tools Artif. Intell. (ICTAI)*, Baltimore, MD, USA, 2020, pp. 748–752. doi: 10.1109/ICTAI50040.2020.00119.

[28] W. Harvey, M. Teng, and F. Wood, “Near-optimal glimpse sequences for improved hard attention neural network training,” in *Int. Joint Conf. Neural Netw. (IJCNN)*, Padua, Italy, 2022, pp. 1–8. doi: 10.1109/IJCNN55064.2022.9892112.

[29] X. Yang, H. Zhang, and J. Cai, “Auto-encoding and distilling scene graphs for image captioning,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2313–2327, May 1, 2022. doi: 10.1109/TPAMI.2020.3042192.

[30] D. Zhao, Z. Chang, and S. Guo, “A multimodal fusion approach for image captioning,” *Neurocomputing*, vol. 338, no. 10, pp. 139–148, 2019. doi: 10.1016/j.neucom.2018.11.004.

[31] V. G. Morelli, M. P. Barbato, F. Piccoli, and P. Napoletano, “Multimodal fusion methods with vision transformers for remote sensing semantic segmentation,” in *13th Workshop Hyperspectral Imaging Signal Process.: Evol. Remote Sens. (WHISPERS)*, Athens, Greece, 2023, pp. 1–5. doi: 10.1109/WHISPERS61460.2023.1043078.

[32] M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *Proc. 36th Int. Conf. Mach. Learn.*, Long Beach, CA, USA, 2019, pp. 6105–6114.

[33] Y. Quan, P. Wang, Y. Wang, and X. Jin, “GUI-based YOLOv8 license plate detection system design,” in *5th Int. Conf. Control Robotics (ICCR)*, Tokyo, Japan, 2023, pp. 156–161. doi: 10.1109/ICCR60000.2023.10444859.

[34] J. Terven, D. -M. Córdova-Esparza, and J. -A. Romero-González, “A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS,” *Mach. Learn. Knowl. Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023. doi: 10.3390/make5040083.

[35] Kaggle, “Flickr 8k dataset,” Apr. 1, 2024. Accessed: Jul. 31, 2024. [Online]. Available: https://www.kaggle.com/datasets/adityajn105/flickr8k

[36] O. Vinyals *et al.*, “Show and tell: A neural image caption generator,” in *Proc. IEEE Conf. Comput. Visi. Pattern Recognit.*, San Juan, PR, USA, 2015, pp. 3156–3164.

[37] K. Xu *et al.*, "Attend and tell: Neural image caption generation with visual attention," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML-15)*, Lille, France, 2015, pp. 2048–2057.

[38] P. Anderson *et al.*, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 6077–6086.

[39] B. Alexey *et al.*, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[40] H. Huang, B. Wang, J. Xiao, and T. Zhu, "Improved small-object detection using YOLOv8: A comparative study," *Appl. Comput. Eng.*, vol. 41, no. 1, pp. 80–88, 2024. doi: 10.54254/2755-2721/41/20230714.

[41] Keras, "YOLOV8 backbones," Apr. 1, 2024. Accessed: Jul. 31, 2024. [Online]. Available: https://keras.io/api/keras_cv/models/backbones/yolo_v8/

[42] V. -T. Hoang and K. -H. Jo, "Practical analysis on architecture of EfficientNet," in *14th Int. Conf. Human Syst. Interact. (HSI)*, Gdańsk, Poland, 2021, pp. 1–4. doi: 10.1109/HSI52170.2021.9538782.

[43] B. L. Menai and M. Chaouki Babahenini, "Recognizing the style of a fine-art painting with EfficientNet and transfer learning," in *7th Int. Conf. Image Signal Process. Appl. (ISPA)*, Mostaganem, Algeria, 2022, pp. 1–6. doi: 10.1109/ISPA54004.2022.9786371.

[44] J. Wang, L. Yang, Z. Huo, W. He, and J. Luo, "Multi-label classification of fundus images with EfficientNet," *IEEE Access*, vol. 8, pp. 212499–212508, 2020. doi: 10.1109/ACCESS.2020.3040275.

[45] H. Alhichri, A. S. Alswayed, Y. Bazi, N. Ammour, and N. A. Alajlan, "Classification of remote sensing images using EfficientNet-B3 CNN model with attention," *IEEE Access*, vol. 9, pp. 14078–14094, 2021. doi: 10.1109/ACCESS.2021.3051085.

[46] T. Chen *et al.*, "TransRNAm: Identifying twelve types of RNA modifications by an interpretable multi-label deep learning model based on transformer," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 20, pp. 3623–3634, Nov.–Dec. 2023. doi: 10.1109/TCBB.2023.3307419.

[47] Z. Tan, Z. Yang, C. Miao, and G. Guo, "Transformer-based feature compensation and aggregation for DeepFake detection," *IEEE Signal Process. Lett.*, vol. 29, pp. 2183–2187, 2022. doi: 10.1109/LSP.2022.3214768.

[48] S. Veena *et al.*, "Comparison of various CNN encoders for image captioning," *J. Phys.: Conf. Ser.*, vol. 2335, no. 1, 2022, Art. no. 012029. doi: 10.1088/1742-6596/2335/1/012029.

[49] D. -H. Hoang, A. -K. Tran, D. N. Minh Dang, P. -N. Tran, H. Dang-Ngoc and C. T. Nguyen, "RBBA: ResNet-BERT-Bahdanau attention for image caption generator," in *14th Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Jeju Island, Republic of Korea, 2023, pp. 430–435. doi: 10.1109/ICTC58733.2023.10392496.

[50] Z. Deng, Z. Jiang, R. Lan, W. Huang, and X. Luo, "Image captioning using DenseNet network and adaptive attention," *Signal Process.: Image Commun.*, vol. 85, no. 12, 2020, Art. no. 115836. doi: 10.1016/j.image.2020.115836.

[51] J. X. Jia, E. Gavves, B. Fernando, and T. Tuytelaars, "Guiding the long-short term memory model for image caption generation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, 2015.

[52] Y. Pan, T. Yao, Y. Li, and T. Mei, "X-linear attention networks for image captioning," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, 2020.