Multimedia Tools and Applications (2024) 83:47963–47977
https://doi.org/10.1007/s11042-023-17359-6

Optimal transformers based image captioning using beam search

Ashish Shetty[1] · Yatharth Kale[1] · Yogeshwar Patil[1] · Rajeshwar Patil[1] · Sanjeev Sharma[1]

Received: 11 August 2022 / Revised: 30 August 2023 / Accepted: 27 September 2023 /
Published online: 31 October 2023

Abstract
Image Captioning is the process of generating textual descriptions of given images. It encompasses two major fields of deep learning, computer vision, and natural language processing. This paper presents an Image Captioning model which uses the Convolution Neural Network (CNN) model for feature extraction and a transformer architecture for the generation of sequences from these feature vectors. For feature extraction, this paper uses different CNN architectures like Xception, InceptionV3, ResNet50V2, VGG19, DenseNet201, ResNet152V2, EfficientNetV2B3, EfficientNetV2B0. The proposed method takes advantage of the transformer model for faster processing, and Beam search is implemented to get the top N most probable sequences for each image. The architecture is trained on Flickr8k dataset and the model outperforms the existing methods. The proposed model achieves a BLEU_4 score of 0.2184 on the Flickr8k dataset.

Keywords Image captioning · Deep learning · Attention · Computer vision · Sequence models

1 Introduction

The process of automatically generating captions for images is called Image Captioning. Humans are capable of describing what is happening in an image or the premise of an

✉ Ashish Shetty
ashishshetty19@cse.iiitp.ac.in

Yatharth Kale
yatharthkale19@cse.iiitp.ac.in

Yogeshwar Patil
yogeshwarpatil19@cse.iiitp.ac.in

Rajeshwar Patil
rajeshwarpatil19@cse.iiitp.ac.in

Sanjeev Sharma
sanjeevsharma@iiitp.ac.in

[1] Indian Institute of Information Technology Pune, Pune, India

image by just looking at it. But it is a difficult problem to tackle by machines, since this task requires understanding the given image and outputting the understanding of the image. There are several applications of Image Captioning like image retrieval, news captioning, biomedical fields, iconographic artwork, and many more.

A lot of Deep neural network techniques are being developed and are still increasing in these years and these techniques are used in a lot of applications like Image Processing, Natural Language Processing etc. One of the applications of Deep learning is Image Captioning. In this application the machine takes an image as an input and its job is to predict the actions happening in the image. Basically, this process consist of two processes, which are Computer Vision and Natural Language Processing.

A Computer can also perform Image Captioning in a much better way using Deep learning. A lot of Deep Learning algorithms are used to solve Image Captioning Problem. A lot of researchers have worked in this field [4, 30] and produced multiple models to get captions from images.

In recent years, there has been an advancement in the area of Image Captioning, there have been several solutions to tackle this problem. One of the most common and effective ways to solve this problem has been using CNN models for feature extraction and vector-to-sequence models for converting the features to sequences. The most common datasets used for the task are FLickr8k [15], Flickr30k [39] and MSCOCO [20] datasets.

In most of the papers the vector-to-sequence model used is an RNN model like GRU [5] and LSTM [14] which give really good results but are quite slow because the input has to be processed sequentially and there are still chances of vanishing and exploding gradients. The LSTM and GRU models try to reduce the effects of vanishing and exploding gradient but are still quite slow for processing. So we are proposing to use a transformer architecture which can process the input in parallel, thus giving a significant increase in processing speed.

The following are the main contributions of our paper:

1. For the Image Captioning model we used CNN model for feature extraction and transformer encoder-decoder architecture for sequence generation.
2. Various CNN pretrained models were tested to find out the best model for the dataset. The models used were Xception, InceptionV3, ResNet50, VGG19, Densenet 201, ResNet152, EfficientNetV2.
3. The transformer takes the feature input in parallel and outputs the next most probable word. Multi-head Attention model is used in the transformer to get the dependence of one part on the rest of the input.
4. Beam Search is able to improve the performance significantly by using the top N predictions at each time step until the end of the sequence is reached.
5. We validated the research paper's results using the public Flicker8k dataset.

Our research explores various CNN models for extracting feature vectors from images. In addition, we leverage the transformer model for the caption generation task, which offers faster computation compared to RNN and LSTM models. Unlike RNN and LSTM, which perform computations serially, the transformer model executes computations in parallel, resulting in improved efficiency. Moreover, we introduce the use of Beam search technique to select the best words generated by the model. This approach was not utilized in previous studies.

The rest of the paper is organized as follows: Section 2 discusses the Literature Survey. Section Section 3 discusses the Materials and Methods used. Section 4 discusses the Proposed Methods. Section 5 discusses the Experiments and Results. Section 6 concludes the paper with future trends.

2 Related work

In this section, we have provided various details about the work that has been done on Image Captioning using different deep learning methods. For Image Captioning CNN and RNN models were used, CNN models are used for feature extraction from images, and RNN models are used for generation of captions using the feature vectors. Various methods like Attention mechanism, Encoder-decoder, etc. are used for caption generation. Different deep learning methods used for Image Captioning are discussed below.

Most of the Image Captioning analysis are based on Encoder Decoder model. This model consists of an encoder which takes image as an input and provides required features to the decoder. The encoder is basically based on CNN model. A wide variety of CNN Models are used in encoders like VGG, ResNet etc.A decoder then takes the features, use Natural language processing to generate captions related to the image.A decoder is mostly based on RNN model. The most widely used RNN model is LSTM. A lot of variants of LSTM are used in the decoder. A lot of researchers like C. Zjou [40], Y. Hua [28], Y. Chu [7] used encoder decoder model to implement Image Captioning Analysis. Dash, Sandeep Kumar [8] goal was to figure out what the image's topic is so that a new deep learning-based encoder-decoder architecture can generate captions for it.

In this paper [33], the authors have presented an end-to-end neural network system that is based on convolution neural network which is responsible for encoding an image into a compact representation of the image, followed by a recurrent neural network responsible for generating corresponding sentence. The paper has proposed to the use of LSTM model for the vector to sequence generation. The BLEU_1 score on Pascal dataset is 59, on Flickr30k is 66, on SBU is 28, and lastly on the COCO dataset, the BLEU_4 score is 27.7. In this paper [37], the author has proposed a system for Image Captioning which uses attention based model that automatically learns to describe the content of images. The author has proposed two attention mechanisms Stochastic Attention(Hard) and Deterministic Attention(Soft). Both the models are trained and tested on Flicker8k, Flicker30k and MS-COCO dataset. The result obtained on Flicker8k dataset was Bleu_1 score of 67 for both Hard and soft attention, for Flicker30k Bleu_1 score of 66.7 for soft and 66.9 for hard attention and for COCO dataset Bleu_1 score of 70.7 for soft and 71.8 for hard attention. In this paper [17], a system for Image Captioning is proposed which has Inception-ResNet as Convolutional Neural Network for extracting features of image, and the authors have used Hierarchical Context based Word embedding for embedding of the caption and Deep Stacked LSTM (Long Short Term Memory) network for decoding. The proposed method has been evaluated on two Image Captioning frameworks Encoder-Decoder and soft attention.The following frameworks were tested on FLicker8k, Flicker30k and MS-COCO dataset.

The authors of this paper [9] proposed a model which is based on encoder-decoder architecture where the CNN models are used to extract features and the descriptions are generated using multimodal gated recurrent units. For weight generation in GRU the model uses part of speech (PoS) and likelihood function for weight generation in the GRU. The K nearest neighbour technique is used for knowledge transfer during validation phase. The models are trained on Flickr 30k and MSCOCO datasets.

In this paper [34], the authors have proposed a deep bidirectional LSTM model, the Image Captioning Model includes a Deep CNN and two separate LSTM networks. Two deep bidirectional varient models are proposed,in which the depth of nonlinearity transition is increased in different ways to learn hierarchical visual-language embeddings. The author uses Flickr8K, Flickr30K and MSCOCO datasets. BLEU_1 for BiLSTM is 65.7, for Bi-SLSTM is 64.2 and for Bi-F-LSTM is 63.

In this paper [2], the author compares the captions generated by attention model and transformer model using BLEU score metric. Author proposed that transformer model with multi-head attention has outperformed the attention model in Image Captioning. BLEU_1 score for attention was 0.199 and for transformer was 0.230 on Flickr8k dataset. In this paper [13], based on the spatial relationship between image regions, the authors propose an image transformer, consisting of a modified encoding transformer and an implicit decoding transformer. To adapt to the structure of images, the original transformer layer's inner architecture is widened.The proposed model was evaluated on Flickr8k dataset.

The authors of this paper [18] propose a framework based on scene graphs for Image Captioning. The CNN features are extracted from bounding box offsets of object entities for visual representation. The semantic relationship features are extracted from triples for semantic relationships. Using these features, a hierarchical-attention module learns discriminative features for word generating.

The paper [35] proposes a captioning model that explicitly explores the object semantic concepts which include category information, relative sizes of the objects and relative distances between the objects. The attention model is part of the decoder in the encoder-decoder model which is responsible for highlighting the informative regions for Image Captioning.The paper [10] proposes a deep attention based language model for learning abstract word information. The LSTM network and the transferred current attention is used to enhance spatial information. The model is trained on MSCOCO and Flickr30K datasets.

In [38] this paper author has proposed a model for Image Captioning task in which author has used denoising diffusion probabilistic model to generate caption for the image.Author has proposed a CLIP-Diffusion-Language model to generate the caption of image.The datasets used to trained the model are Flickr8k and Flickr30k dataset.The author has achieved a BLEU_4 score of 0.1876 on Flickr8k dataset and BLEU_4 of 0.2470 on Flickr30k.

Attention Mechanism has proven effectiveness in action recognition as shown by authors of [36] which proposes Symbiotic Attention with Object-centric feature Alignment (SAOA) framework. The framework described in the paper comprises three feature extractors and one interaction module. The detection model generates local object features and location proposals, incorporating location-aware information through an object-centric alignment method. In the Verb branch, the feature map is aligned with the objects by combining local motion features with corresponding object detection features. In the Noun branch, the object features are aligned with the global noun representation. The fused features from each branch then interact with the global feature from the other branch using a symbiotic attention mechanism. The outputs of this process, referred to as SAOA, are utilized for verb and noun classification, respectively. The framework leverages the communication between the verb branch, noun branch, and local object information, resulting in enhanced performance and capabilities. The framework demonstrates its superiority by achieving the state-of-the-art performance on the EPIC-Kitchens Action Recognition Challenge, which is the largest egocentric video dataset available.

In the paper [11], the authors propose a language CNN model designed for statistical language modeling tasks, demonstrating competitive performance in image captioning. The model takes a query image as input and estimates the probability distribution of the next word based on previous words and the image. It comprises four components: a CNNI for extracting image features, a deep CNNL for language modeling, a multimodal layer (M) that connects the CNNI and CNNL, and a Recurrent Network (such as RNN or LSTM) for word prediction. The weights are shared across all time frames. Their language CNN model takes into account all the previous words, allowing it to capture long-range dependencies in the history of words, which is crucial for image captioning. The effectiveness of the approach is validated on two datasets: Flickr30K and MS COCO. The proposed method outperforms vanilla recurrent neural network-based language models and achieves competitive performance comparable to state-of-the-art methods.

In [21] this paper author has proposed a new method for video question answering.The proposed method involved embedding of both visual instances and textual representation into a graph nodes.The method also uses graph casual convolution (GCC) on graph-structured sequences.The proposed method is evaluated on multiple datasets, including TVQA+. The method outperform the state-of-the-art methods on theses benchmark dataset.

Most of the existing studies have relied on RNNs and LSTMs to generate captions. These models handle the input data sequentially, processing each feature vector from images and token from sentences one after another. However, the transformer model we employed takes a different approach by processing these inputs in parallel, allowing for more efficient and effective caption generation.

3 Materials and methods

3.1 Methodology

As shown in Fig. 1, we first looked into the literature regarding Image Captioning to see what approaches were popular and how they could be improved. We selected the Flickr8k dataset for our experiment because it is one of the most popular datasets publicly available.

We developed an image captioning model that combines the power of a CNN model for feature extraction with a transformer encoder-decoder architecture for sequence generation. This two-stage approach allows us to effectively capture image information and generate descriptive captions.

To identify the most suitable CNN model for our dataset, we conducted extensive experiments with various pretrained models. The models we evaluated included Xception, InceptionV3, ResNet50, VGG19, Densenet 201, ResNet152, and EfficientNetV2. Through this evaluation process, we determined the best-performing model that produces high-quality image features for our image captioning task.

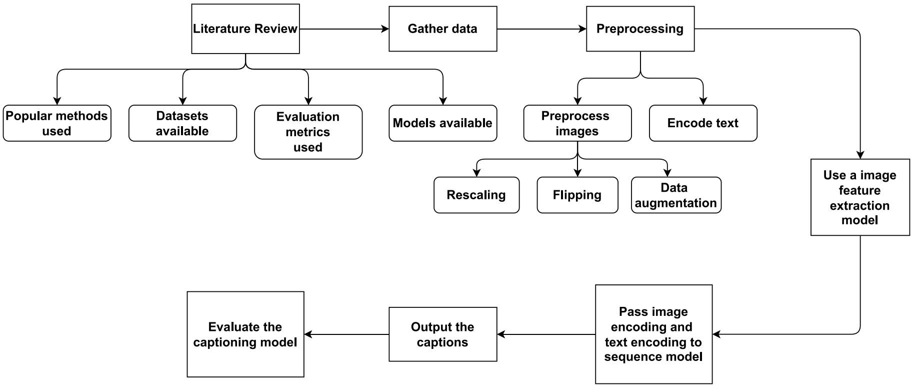


Fig. 1 Methodology flow graph

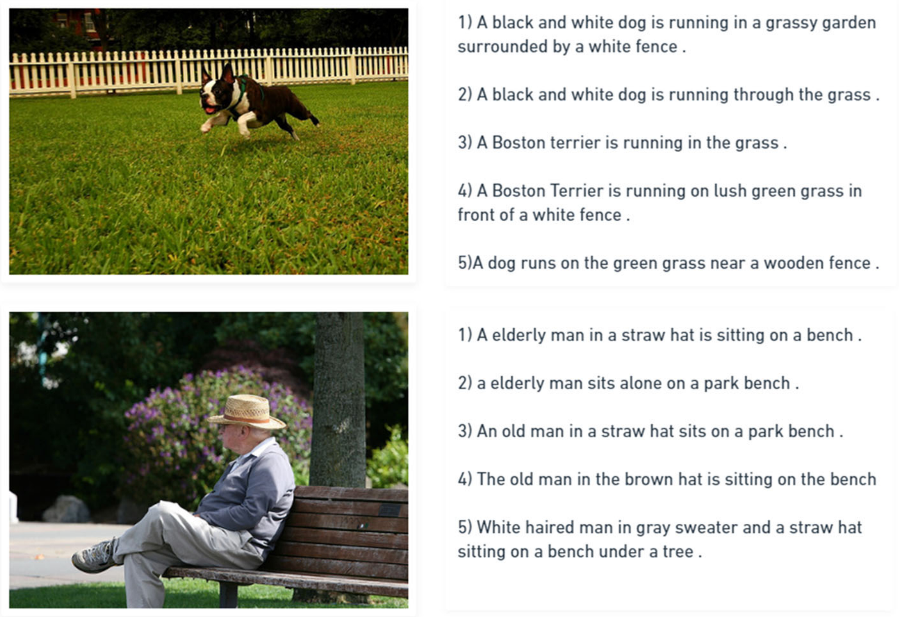
Our transformer architecture takes the extracted image features as input and generates the most probable next word in the caption sequence. To capture the dependencies between different parts of the input, we incorporated a multi-head attention model within the transformer. This enables the model to effectively understand the relationships and context within the image features, leading to improved caption generation.

Overall, our contributions encompass the development of an image captioning model that combines a CNN for feature extraction, exploration of various CNN pretrained models, the utilization of a transformer architecture with multi-head attention, and the validation of our approach using the Flicker8k dataset.

3.2 Dataset

The Flickr8k dataset [15] is a benchmark collection for sentence-based image description and search. The dataset contains 8000 images and five different captions describing each image, providing clear descriptions of the key entities and events. The images were chosen from six different Flickr groups and were manually selected to depict a variety of scenes and situations. The dataset does not contain any well-known individuals or locations. Figure 2 shows some of the images and their corresponding captions from the above dataset.

From the images and captions, a dictionary was generated which has key as the location of the image and its values as the corresponding caption of the images. This step was done to easily pass input to the model.



1) A black and white dog is running in a grassy garden surrounded by a white fence .

2) A black and white dog is running through the grass .

3) A Boston terrier is running in the grass .

4) A Boston Terrier is running on lush green grass in front of a white fence .

5)A dog runs on the green grass near a wooden fence .

1) A elderly man in a straw hat is sitting on a bench .

2) a elderly man sits alone on a park bench .

3) An old man in a straw hat sits on a park bench .

4) The old man in the brown hat is sitting on the bench

5) White haired man in gray sweater and a straw hat sitting on a bench under a tree .

Fig. 2 Sample dataset images and their captions

3.3 Pre-processing

The dataset consists of images and 5 captions for each image. The images of the dataset are resized to 224 × 224 to obtain a new pre-processed dataset. These images are rescaled to transform every pixel value from the range [0-255] to [0-1] in order to treat high-resolution and low-resolution images in the same manner. Data augmentation is applied to the dataset namely horizontal-flip, rotation range, and random contrast to increase the dataset and also make the trained model more robust to real-life data.

The captions are preprocessed by removing the captions from the dataset that have a length greater than 30 words. This is done because more than 95% of the captions have length less than or equal to 30 words. This step avoids any outlier captions with large descriptions.

4 Proposed methods

The Image Captioning model is responsible for generating captions that describe the image. Figure 3 shows the key components of the Image Captioning model and how it works.

4.1 Input and positional embedding

The Image Captioning model takes as input images and their captions. The images are passed through the Convolutional Neural Network to extract feature vectors from the images. These feature vectors are then passed to the encoder layer of the transformer. The output of the encoder is then passed to the decoder, where the caption generation will happen.

The input captions are converted into tokens using MPNetv2 tokenization technique. MPNetv2 uses the WordPiece tokenizer [24] for the tokenization process. WordPiece Tokenization is a subword tokenization technique used in natural language processing (NLP) tasks, including machine translation, language modeling, and text classification. It breaks down words into smaller units called subword tokens or word pieces.

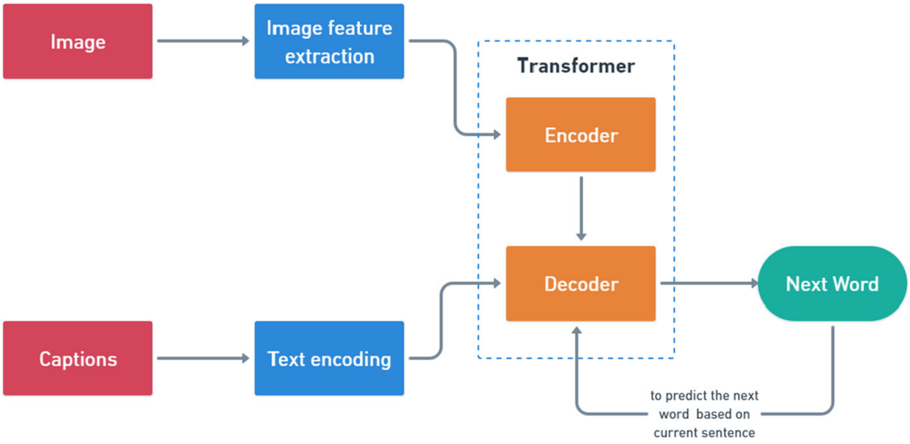


Fig. 3 The architecture of the Image Captioning model

The main idea behind WordPiece Tokenization is to handle out-of-vocabulary (OOV) words and improve the coverage of rare or unseen words. Instead of treating each word as a single token, Word Piece Tokenization divides words into subword units based on the training corpus.

The decoder takes the encoder's output and an encoding for the caption as its input. Embedding vectors are created for each token (word) forming an input sequence. Each embedding vector representing an input word is augmented by element-wise summation with a positional encoding vector of the same length, thereby introducing positional information into the input. These vectors are then passed on to the decoder layer of the transformer.

4.2 Image feature extraction

The features of the images are extracted using Convolutional Neural Networks like efficientnet [29], xception [6], inceptionv3 [27], resnet50v2 [25] , vgg19 [12], densenet201 [16], and resnet152 [12] . If the images are directly converted to feature vectors pixel by pixel, the feature vector would be sparse with a lot of redundant information. The CNN models help find the key features from the images enabling the captioning process.

Several CNN models pretrained on the ImageNet dataset were used. We freeze the weights of these models and remove the fully connected layers in order to get the feature vectors for the image.

4.3 Transformer

The transformer enables us to process input data in parallel, allowing an increase in speed over the traditional approach of sequence models. The transformer model used by us borrows its idea from [31]. The feature vector generated from the CNN model is passed to the encoder, where self-attention is applied to the vector as described in [31]. Multi head attention layer is responsible for understanding how much one item or component of input is dependent on the other components of the input.

$S(X) = Attention(W_q X, W_k X, W_v X)$ where W_q, W_k, W_v are matrices of learnable weights and X is the feature vector. Next we pass it to $L(X) = LayerNorm(X + S(X))$ [1]. $L(X)$ is passed to feed forward network FF and final output is derived from $Enc_{out} = LayerNorm(L(X) + FF(L(X))$.

In the Decoder phase, the tokenized caption is passed to the positional embedding which gives the sentence embeddings $C_X$for the captions. These are then passed to the masked multi-head attention layer which is called so because the complete output is not visible. The sentence processed so far is used to determine how much the current word is dependent on the rest of the sentence. The output of this layer, along with encoder output Enc_{out} is passed to the multi-head attention model of the decoder as shown in Fig. 4. In this layer, it is determined how much the current word is dependent on the processed sentence and also on the encoder output Enc_{out}.

All of our models make use of a single encoder layer and two decoder layers. The output of the transformer is a vector of probabilities of the next word, the softmax layer outputs the next most probable word.

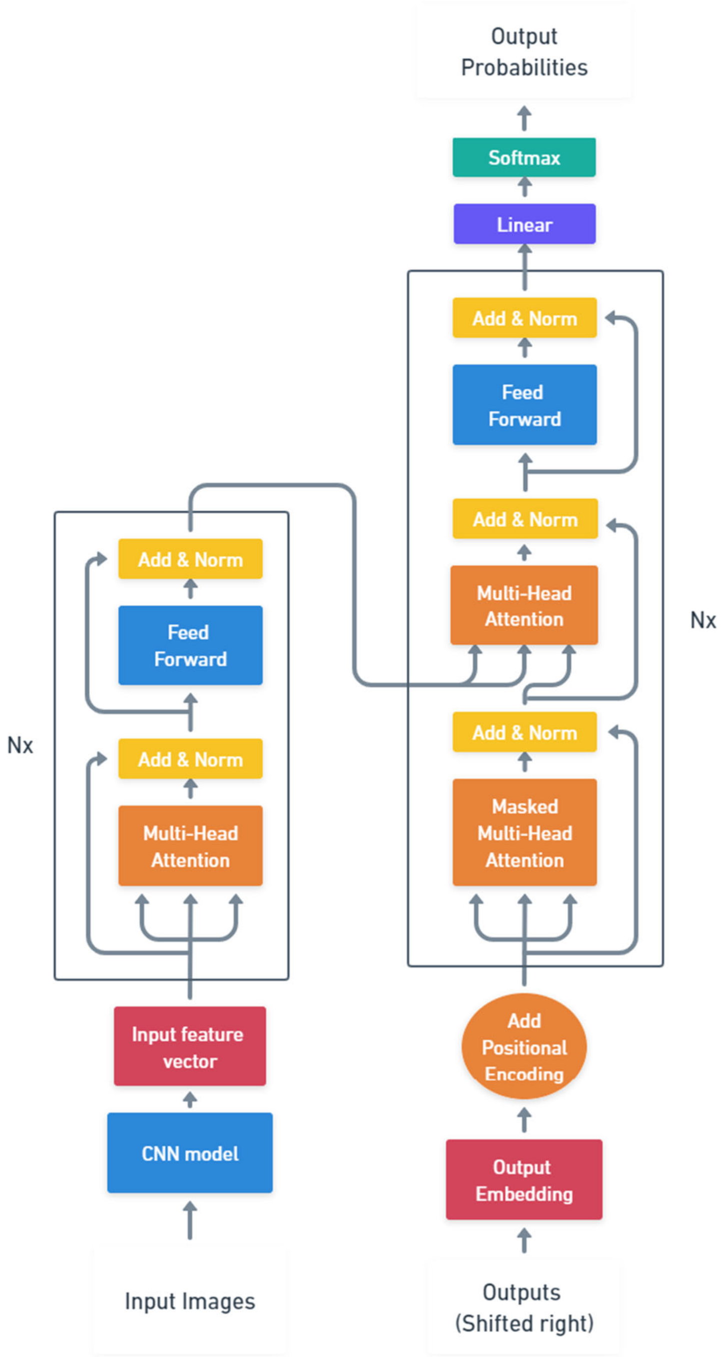


Fig. 4 The architecture of transformer [31]

4.4 Embedding matrix

The embedding matrix is used for word embedding in the decoder layer. For creating the embedding matrix we have used MPNetv2 model [26] with pre-trained weights from sentence transformers [23]. The word embeddings created by this model are a 768-dimensional vector that changes depending on the context in which it appears in a phrase. The changing word embeddings were dealt with by averaging the embedding generated throughout the entire training dataset to construct a final embedding matrix that included every word embedding. i.e

$$W_f = \sum_{i=1}^{N} \frac{W_i}{N} \tag{1}$$

W_f is the final word embedding, W_i is the word embedding for the i^{th} sentence and N is the number of sentence the word exists in.

4.5 Beam search

Beam search is a heuristic search algorithm used as a final decision-making layer in many NLP and speech recognition models to find the best output based on the previous prediction and current prediction. For a Beam search of width N, the algorithm selects the top N predictions at each step until it encounters the end sequence token. We have applied a beam search of width 5 for the best performing model and a simple greedy search for the rest. We have observed that increasing the width of the beam search consistently leads to improved results. However, determining the appropriate width involves striking a balance between computational complexity and accuracy. Historically, a commonly employed beam width has been 5. It is important to note that while a higher beam width can yield better outcomes, it also entails increased computational time.

5 Experiments and results

5.1 Hardware and software setup

Tesla K80 GPU and 13 GB RAM used for training along with TensorFlow, Keras, and Scikit-learn libraries in Google Colab, coded in Python 3.7.10.

5.2 Evaluation criteria

In the prediction phase, four quantitative performance measures were computed to access the reliability of trained models using the validation data, including BLEU score [22], METEOR score [3] , CIDEr score [32] and ROGUE_L score [19]. BLEU score is the most commonly used metrics to evaluate the quality of generated captions. While BLEU is widely used, other metrics such as CIDEr, METEOR and ROGUE_L are also used.

BLEU Score:

$$\log Bleu = min\left(1 - \frac{r}{c}, 0\right) + \sum_{n=1}^{4} \frac{\log p_n}{4} \tag{2}$$

(where r is target length = number of words in the target sentence)
(and c is predicted length = number of words in the predicted sentence)
(p_n = Precision n-gram)

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

TP: True Positive
FP: False Positive
FN: False Negative

$$F_{mean} = \frac{10 * Precision * Recall}{(9 * Precision) + Recall} \tag{5}$$

Chunk Penalty:

$$p = 0.5 * \left(\frac{c}{u_m}\right)^3 \tag{6}$$

METEOR score:

$$M = F_{mean}(1 - p) \tag{7}$$

CIDEr score:

$$CIDEr(c_i, S_i) = \frac{1}{m} \sum_{j}^{T} \frac{g^n(c_i) \cdot g^n(s_{ij})}{\|g^n(c_i)\| \, \|g^n(s_{ij})\|} \tag{8}$$

T = (TF-IDF vector(n-gram))

$$CIDEr(c_i, S_i) = \sum_{n=1}^{N} w_n CIDEr_n(c_i, S_i) \tag{9}$$

ROGUE_L: Longest Common Subsequence (LCS) based statistics. Longest common subsequence problem takes into account sentence level structure similarity naturally and identifies longest co-occurring in sequence n-grams automatically.

5.3 Results

Initially, we built the Image Captioning model by trying various CNN models to find the best one for captioning purposes. The CNN model's output was used by the encoder layer of the transformer. The captions were encoded in the first step by simple tokenization, giving a unique number for each word. The decoder layer received the encoded captions. The next word was predicted based on the prior statement and the encoder's output. The model is trained on Flickr8k dataset.

The problem with this method was that since the words were directly tokenized, their numbers had no significance or information; for example, two close numbers did not mean that the corresponding words might have a close relationship. The Table 1 shows the results of the proposed Image Captioning models with different CNN models for image encoding.

To improve the model's performance, we encoded the captions using the MPNetv2 sentence transformer. [26]. By doing so, the word embedding adds meaning to the number given to the word; words with similar meaning or features have closer numbers compared to others. For example, if there are three words signifying two athletes and a dog the word embedding of the two athletes would be much closer as compared to an athlete and a dog, since the two

Table 1 Results of image captioning models with different CNN models

Model	Bleu_1	Bleu_2	Bleu_3	Bleu_4	CIDEr OR	METE E_L	ROUG
EfficientNetB0	**0.5146**	**0.3458**	**0.2265**	**0.1441**	**0.3835**	**0.1174**	**0.3276**
Xception	0.4085	0.2291	0.1259	0.0721	0.1596	0.0841	0.2557
InceptionV3	0.4102	0.2269	0.1191	0.0644	0.1502	0.0861	0.2568
ResNet50	0.4960	0.3247	0.2060	0.1287	0.3409	0.1101	0.3181
VGG19	0.4826	0.3119	0.1916	0.1148	0.3350	0.1139	0.3122
Densenet201	0.4269	0.2501	0.1437	0.0833	0.2080	0.0948	0.2712
ResNet152	0.5055	0.3346	0.2141	0.1359	0.3528	0.1131	0.3238

athletes have more common characteristics. The rest of the architecture is the same as the prior model. The model is trained on the Flickr8k dataset.

A Greedy approach is implemented to find the most similar captions, and to improve the results, we implemented beam search. In beam search we set a parameter N (beam width), the algorithm selects the top N predictions at each step until the end sequence is encountered. The best performing model, which is EfficientNetV2B0, was used and beam search was applied to find the top N predictions at each step.

The improvement in the result can be seen in above Table 2. The Table 2 shows the results of Image Captioning models with different CNN models and new Word Embedding.

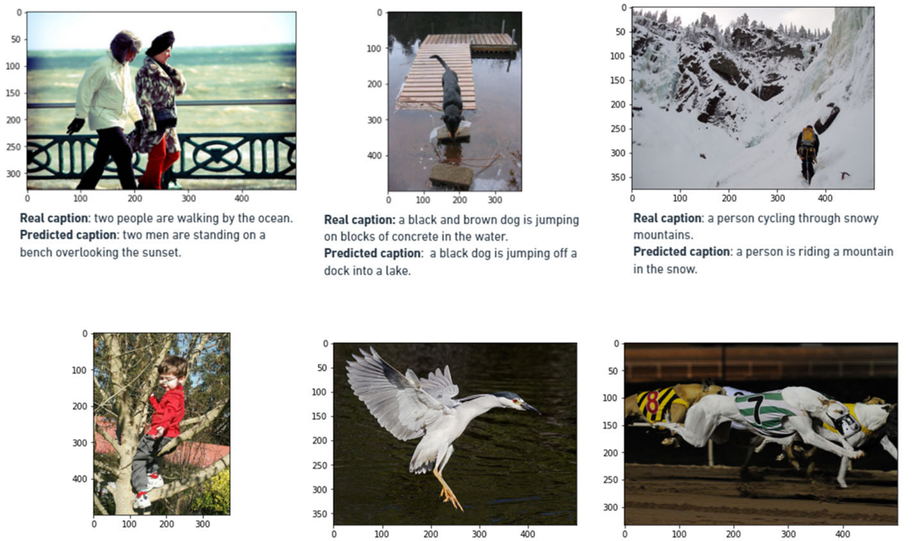
Some of the results of our model are presented in the Fig. 5.The figure shows the image and their corresponding caption genrated by the model.

6 Comparative study

We have evaluated our Image Captioning model with existing work. In this comparison, the evaluation metrics used are Bleu_1, Bleu_2, Bleu_3, Bleu_4, Meteor, and CIDEr. The

Table 2 Results of image captioning models with different CNN models and new word embedding

Model	Bleu_1	Bleu_2	Bleu_3	Bleu_4	CIDEr	METE OR	ROUG E_L
Xception	0.4522	0.2640	0.1500	0.0907	0.1293	0.1449	0.3463
InceptionV3	0.4483	0.2552	0.1357	0.0771	0.1178	0.1410	0.3392
ResNet50V2	0.4594	0.2684	0.1510	0.0921	0.1296	0.1446	0.3503
VGG19	0.5223	0.3476	0.2239	0.1450	0.3085	0.1902	0.4119
Densenet 201	0.4803	0.2996	0.1842	0.1160	0.2133	0.1674	0.3740
ResNet152V2	0.4404	0.2598	0.1490	0.0901	0.1446	0.1506	0.3489
EfficientNetV2B3	0.5648	0.3753	0.2416	0.1558	0.3391	0.1895	0.4117
EfficientNetV2B0	0.5938	0.4150	0.2833	0.1929	0.4283	0.2106	0.4417
EfficientNetV2B0 "Beam search"	**0.6346**	**0.4550**	**0.3175**	**0.2184**	**0.5209**	**0.1987**	**0.4590**



Real caption: two people are walking by the ocean.
Predicted caption: two men are standing on a bench overlooking the sunset.

Real caption: a black and brown dog is jumping on blocks of concrete in the water.
Predicted caption: a black dog is jumping off a dock into a lake.

Real caption: a person cycling through snowy mountains.
Predicted caption: a person is riding a mountain in the snow.

Real caption: a little boy in a red sweater is climbing a tree.
Predicted caption: a young child in a red jacket climbs a tree.

Real caption: a beautiful white bird takes off in flight.
Predicted caption: a bird flies over the water.

Real caption: a group of greyhound dogs racing with muzzles covering their noses.
Predicted caption: a greyhound wearing a muzzle and a muzzled greyhound is racing on a track.

Fig. 5 Sample output of captioning model

Table 3 shows the comparison of our Image Captioning model with the present models. And we can see the performance of our model outperforms other methods.

From the above table, we can see that our model, which consists of EfficientNetV2B0 and a transformer model with beam search technique is better than most of the present work. Also, in comparison to the Soft Attention model [17] our BLEU_2, BLEU_3 and BLEU_4 is better.

Table 3 The comparison between our methods and the comparable models on Flickr8k dataset

Model	Bleu_1	Bleu_2	Bleu_3	Bleu_4	METEOR	CIDEr
Karpathy et al.[16]	0.579	0.383	0.245	0.160	-	-
Mao et al.[22]	0.5778	0.2751	0.2307	-	-	-
Vinyals et al[35]	0.63	0.41	0.27	-	-	-
Soft-Attention[38]	0.67	0.448	0.299	0.195	0.1893	-
Hard-Attention[38]	0.67	0.457	0.314	0.213	0.203	-
Encoder-Decoder[17]	0.628	0.442	0.307	0.211	0.199	0.527
Soft Attention[17]	0.635	0.452	0.313	0.214	0.200	0.555
EfficientNetV2B0 "Beam search"	**0.6347**	**0.4551**	**0.3176**	**0.2184**	**0.1987**	**0.5209**

7 Conclusion and future scope

For building an Image Captioning model, we have implemented CNN models for feature extraction and transformers for sequence generation. Using an attention mechanism instead of RNN-based models like GRU and LSTM allows for faster training of the model, and it also solves the problem of vanishing and exploding gradients. We have also implemented Beam search to find the N most probable sequences for each image.

For Image Captioning, we have used the Flicker8k dataset. The best result achieved for image captioning on the Flicker8k dataset was BLEU_4 score of 0.2184. The BLEU_4 score for Flicker8k has beaten all the state-of-the-art methods in Image Captioning.

For future scope, we intend to increase the size of the model or make use of modern transformer architectures like Meshed-Memory Transformers.

Funding There has been no significant financial support for this work that could have influenced its outcome.

Data availability statement The Flickr8k [15] dataset.

Declarations

Conficts of interest The authors confirm that there are no known conflicts of interest associated with this publication.

References

1. Ba JL, Kiros JR, Hinton GE (2016) Layer normalization. arXiv:1607.06450
2. Balasubramaniam D (2021) Evaluating the performance of transformer architecture over attention architecture on image captioning
3. Banerjee S, Lavie A (2005) Meteor: an automatic metric for mt evaluation with improved correlation with human judgments. In: Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, pp 65–72
4. Carrara F, Falchi F, Caldelli R, Amato G, Becarelli R (2019) Adversarial image detection in deep neural networks. Multimedia Tools Appl 78(3):2815–2835
5. Cho K, Van Merriënboer B, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation: Encoder-decoder approaches. arXiv:1409.1259
6. Chollet F (2017) Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1251–1258
7. Chu Y, Yue X, Yu X, Wang Z (2020) Automatic image captioning based on RESNET50 and LSTM with soft attention
8. Dash SK, Acharya S, Pakray P, Das R, Gelbukh A (2020) Topic-based image caption generation
9. do Carmo Nogueira T, Vinhal CDN, da Cruz Júnior G, Ullmann MRD (2020) Reference-based model using multimodal gated recurrent units for image captioning. Multimedia Tools Appl 79(41):30615–30635
10. Fang F, Wang H, Chen Y, Tang P (2018) Looking deeper and transferring attention for image captioning. Multimedia Tools Appl 77(23):31159–31175
11. Gu J, Wang G, Cai J, Chen T (2017) An empirical study of language cnn for image captioning. In: Proceedings of the IEEE international conference on computer vision, pp 1222–1231
12. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
13. He S, Liao W, Tavakoli HR, Yang M, Rosenhahn B, Pugeault N (2020) Image captioning through image transformer. In: Proceedings of the Asian conference on computer vision
14. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
15. Hodosh M, Young P, Hockenmaier J (2013) Framing image description as a ranking task: data, models and evaluation metrics. J Artif Intell Res 47:853–899
16. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4700–4708

17. Katiyar S, Borgohain SK (2021) Image captioning using deep stacked lstms, contextual word embeddings and data augmentation. arXiv:2102.11237
18. Li X, Jiang S (2019) Know more say less: image captioning based on scene graphs. IEEE Trans Multimedia 21(8):2117–2130
19. Lin C-Y, Och FJ (2004) Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In: Proceedings of the 42nd annual meeting of the association for computational linguistics (ACL-04), pp 605–612
20. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft COCO: common objects in context. In: European conference on computer vision, Springer, pp 740–755
21. Liu R, Han Y (2022) Instance-sequence reasoning for video question answering. Front Comput Sci 16(6):166708
22. Papineni K, Roukos S, Ward T, Zhu W-J (2002) BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the association for computational linguistics, pp 311–318
23. Reimers N, Gurevych I (2019) Sentence-BERT: sentence embeddings using siamese BERT-networks. In: Proceedings of the 2019 conference on empirical methods in natural language processing, association for computational linguistics, p 11
24. Schuster M, Nakajima K (2012) Japanese and Korean voice search. In: 2012 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 5149–5152
25. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
26. Song K, Tan X, Qin T, Lu J, Liu T-Y (2020) MPNET: masked and permuted pre-training for language understanding. Adv Neural Inf Process Syst 33:16857–16867
27. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2818–2826
28. Tan HY, Chan SC (2018) Phrase-based image caption generator with hierarchical LSTM network
29. Tan M, Le Q (2019) Efficientnet: rethinking model scaling for convolutional neural networks. In: International conference on machine learning, pp 6105–6114
30. Tiwary T, Mahapatra RP (2022) An accurate generation of image captions for blind people using extended convolutional atom neural network. Multimedia Tools Appl 1–30
31. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, vol 30
32. Vedantam R, Zitnick CL, Parikh D (2015) CIDER: consensus-based image description evaluation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4566–4575
33. Vinyals O, Toshev A, Bengio S, Erhan D (2015) Show and tell: a neural image caption generator. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3156–3164
34. Wang C, Yang H, Bartz C, Meinel C (2016) Image captioning with deep bidirectional LSTMS. In: Proceedings of the 24th ACM international conference on multimedia, pp 988–997
35. Wang S, Lan L, Zhang X, Dong G, Luo Z (2020) Object-aware semantics of attention for image captioning. Multimedia Tools Appl 79(3):2013–2030
36. Wang X, Zhu L, Wu Y, Yang Y (2020) Symbiotic attention for egocentric action recognition with object-centric alignment. IEEE Trans Pattern Anal Mach Intell
37. Xu K, Ba J, Kiros R, Cho K, Courville A, Salakhudinov R, Zemel R, Bengio Y (2015) Show, attend and tell: Neural image caption generation with visual attention. In: International conference on machine learning, pp 2048–2057
38. Xu S (2022) Clip-diffusion-LM: apply diffusion model on image captioning, arXiv:2210.04559
39. Young P, Lai A, Hodosh M, Hockenmaier J (2014) From image descriptions to visual denotations: new similarity metrics for semantic inference over event descriptions. Trans Assoc Computat Linguist 2:67–78
40. Zhou C, Lei Z, Chen S, Huang Y, Xianrui L (2016) A sparse transformer-based approach for image captioning

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Terms and Conditions