A Hindi Image Caption Generation Framework Using Deep Learning

SANTOSH KUMAR MISHRA, RIJUL DHIR, SRIPARNA SAHA, and PUSHPAK BHATTACHARYYA, Indian Institute of Technology, Patna

Image captioning is the process of generating a textual description of an image that aims to describe the salient parts of the given image. It is an important problem, as it involves computer vision and natural language processing, where computer vision is used for understanding images, and natural language processing is used for language modeling. A lot of works have been done for image captioning for the English language. In this article, we have developed a model for image captioning in the Hindi language. Hindi is the official language of India, and it is the fourth most spoken language in the world, spoken in India and South Asia. To the best of our knowledge, this is the first attempt to generate image captions in the Hindi language. A dataset is manually created by translating well known MSCOCO dataset from English to Hindi. Finally, different types of attention-based architectures are developed for image captioning in the Hindi language. These attention mechanisms are new for the Hindi language, as those have never been used for the Hindi language. The obtained results of the proposed model are compared with several baselines in terms of BLEU scores, and the results show that our model performs better than others. Manual evaluation of the obtained captions in terms of adequacy and fluency also reveals the effectiveness of our proposed approach.

Availability of resources: The codes of the article are available at https://github.com/santosh1821cs03/Image_Captioning_Hindi_Language; The dataset will be made available: http://www.iitp.ac.in/~ai-nlp-ml/resources.html.

CCS Concepts: • **Information systems → Information extraction**;

Additional Key Words and Phrases: Image captioning, hindi, deep-learning, attention

ACM Reference format:
Santosh kumar Mishra, Rijul Dhir, Sriparna Saha, and Pushpak Bhattacharyya. 2021. A Hindi Image Caption Generation Framework Using Deep Learning. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 20, 2, Article 32 (March 2021), 19 pages.
https://doi.org/10.1145/3432246

1 INTRODUCTION

Automatic caption generation for an image is one of the challenging problems in artificial intelligence. Image captioning models not only solve computer vision challenges of object recognition but also capture and express their relationships in natural language. This task is more complicated

Sriparna Saha would like to acknowledge the support of SERB WOMEN IN EXCELLENCE AWARD 2018 (SB/WEA-07/2017) of the Department of Science and Technology for carrying out this research.
Authors' addresses: S. K. Mishra, R. Dhir, S. Saha, and P. Bhattacharyya, Indian Institute of Technology, Patna; emails: {santosh_1821cs03, rijul.cs15, sriparna, pb}@iitp.ac.in.

2375-4699/2021/03-ART32 $15.00
https://doi.org/10.1145/3432246

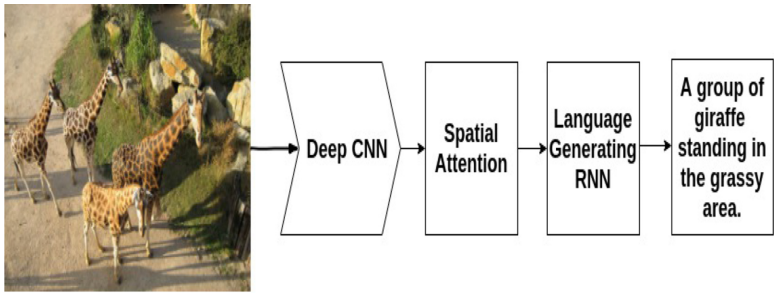

Fig. 1. Caption generation model with spatial attention, based on end-to-end neural network consisting of CNN followed by a language generating RNN.

as compared to well-studied image classification and object recognition tasks, which have been the main focus in the computer vision community. Recent advancements in language modeling and object recognition have made image captioning an essential research area in computer vision and natural language processing. Caption generation of an image has a great impact by helping visually impaired people to better understand the contents on the web [25].

Deep learning–based models for image captioning came into existence with recent advancements in machine translation, where the task is to translate a sentence S in the source language into target language T, by maximizing $p(T|S)$. In the past, machine translation has been carried out in various ways, such as translating words individually, aligning words, reordering, and so on. Recent works have shown that translation can be done in a much simpler way by using Recurrent Neural Networks (RNNs) [2, 4, 31]. RNNs can achieve state-of-the-art-performance where an encoder RNN reads the source sentence and transforms it into fixed-length vector representation, which is used as an initial hidden state of a decoder RNN that generates the target sentence.

Image captioning problem is very much similar to machine translation, where the source language is an image, and the target language is a sentence. Here, a recurrent neural network is replaced by a convolution neural network (CNN) in the encoder side. In recent works, it has been shown that CNN can be used as an encoder and can produce a rich representation of an input image by embedding it to a fixed-length vector representation, this representation can be used for a variety of computer vision tasks [29]. In this work, we have used pre-trained CNN and then used the last hidden layer as input to decoder RNN that generates sentences (shown in Figure 1).

In recent years deep learning–based techniques become popular in solving various problems of computer vision. A deep learning–based visual description framework first encodes an image into a fixed-length feature vector using a convolution neural network followed by the recurrent neural network as a decoder to generate the descriptions. In most of the deep learning frameworks, Long Short-Term Memory–based recurrent neural network is used for language modeling [7, 27]. Long Short-Term Memory (LSTM)–based recurrent neural networks were developed by Hochreiter et al. [15] in the year 1997. It is a special kind of recurrent neural network capable of learning long-term dependencies. Gated Recurrent Unit (GRU)–based neural networks, introduced by Cho et al. [4] in the year 2014, are also capable of learning long-term dependencies. The difference between LSTM and GRU is that LSTM has three gates (input, output, and forget gates) that control the behaviors of LSTM cell, whereas GRU has two gates (reset and update gates). GRU is relatively new as compared to LSTM, and it has less complex structures. GRU is computationally more efficient, trains fast, and performs better than LSTM for language modeling [37]. But we did not find any single work on image captioning where GRU-based model was applied. Inspired by this, in the current work, we

aim to develop an image captioning framework utilizing GRU-based deep learning model as a decoder.

In this work, we have developed the image captioning model for the Hindi language. It is one of the oldest languages in the world, spoken in South Asia and India. It is evolved from Sanskrit language [13]. Hindi is the fourth most spoken language in the world. India is a linguistically diverse country; there are 22 regional languages, and Hindi and English are the official languages. Most of the people in India communicate in the Hindi language. This was the primary motivation for us to develop an image captioning model in the Hindi language. Our contributions are as follows:

- We have prepared the Hindi dataset for image captioning, as there was no dataset available in the Hindi language. We have used well-known MSCOCO dataset; it is a publicly available dataset used in various computer vision tasks. We have first translated each sentence using Google translator to prepare the corpus. The translation is manually verified by human annotators and corrected accordingly to build the Hindi image captioning corpus.
- We have developed a novel image captioning model using RESNET101 as the Encoder and GRU as the decoder with spatial attention. We have experimented with many different attention mechanisms, such as Visual attention, Bahdanau attention, and Luong attention. To the best of our knowledge, there does not exist any image captioning model for the Hindi language.
- We have used various encoder-decoder architectures to explore the best model.

2 RELATED WORKS

In the literature, Image captioning problem has been solved using two approaches. The first approach is the top-down approach [2, 31, 35] and the second approach is the bottom-up approach [9, 11, 18].

In the top-down approach, the input image is first converted into words while in case of a bottom-up approach, it comes up with words that describe the various aspects of an image, and then words are combined to generate the description of an image. In the top-down approach, end-to-end formulation is used from an image to sentence, and all the parameters of networks are learned during training.

In the bottom-up approach, visual concepts, object attributes, words, and phrases are combined using the language model. Farhadi et al. [11] defined a method that can compute a score, linking an image to sentence; based on the score, a descriptive sentence is attached with the image. The score is computed by comparing the estimate of meaning obtained from an image with meaning obtained from the sentence. Kulkarni et al. [18] had generated description of an image by detection of objects, modifier (adjective), and spatial relationship (prepositions) in an image; description of an image is generated either by using n-gram language model [3] or template-based approach [39]. Li et al. [21] have given a method to automatically generate the description using the web-scale n-grams. Elliot et al. [9] have developed a template-based approach, and it operates over visual dependency representation to capture the relationships between objects and images. Kuznetsova et al. [19] developed a data-driven approach for image description, exploiting the vast amount of image data and associated natural language descriptions available on the web. Human-composed phrases are used to describe the similar images concerning the given image and then selectively combine those phrases to generate the description. Fang et al. [10] used a visual detector, language model, and multimodal similarity model learned directly from a dataset of image captions; word detector used as a conditional input to a maximum entropy language model that learns from a set of 40,000 image descriptions to capture the statistics of word usage. The final caption is generated

by re-ranking caption candidates using sentence-level features and a deep multimodal similarity model. Lebret et al. [20] used a purely bi-linear model that learns a metric between an image representation, generated from previously trained CNN, and phrases that are used to describe them. The proposed method infers phrases from a given image based on the caption syntax statistical language model, which generates the description for a given image using the phrases inferred.

The top-down approach is used nowadays for image captioning, as it is very similar to machine translation, where encoder-decoder architecture is used. Sutskever et al. [31] developed a sequence to sequence learning model where they used multilayer LSTM as an encoder to encode the input sequence to the intermediate vector of fixed dimension and then another LSTM is used to decode the target sequence from intermediate representation. Bahdanau et al. [2] proposed an encoder-decoder architecture for machine translation, and it improves the performance of basic encoder-decoder architecture by allowing the model to automatically search part of the sentence that is relevant to predict the target word. Cho et al. [4] also developed RNN encoder-decoder architecture consisting of two recurrent neural networks; one codes an input sequence to fixed-length vector representation and another decodes the representation into the sequence of symbols. The encoder-decoder of the proposed model is jointly trained to maximize the conditional probability of the target sequence given an input sequence. Wu et al. [35] have shown that a combination of CNN and RNN is effective in converting image features into text, but this approach does not explicitly represent high-level semantic concepts. They have developed a method by incorporating high-level concepts into CNN-RNN architecture and showed that it achieves a significant improvement in both image captioning and visual question answering tasks. Vinyals et al. [33] developed a generative model using a deep recurrent architecture that combines recent advances in computer vision and machine translation. The proposed model is trained to maximize the likelihood of the target description sentence given the training image. Mao et al. [27] developed a multimodal RNN model for generating the description. They modeled the probability distribution of generating a word given previous words and an image. The image description is generated according to probability distributions. Karpathy et al. [17] developed an alignment model based on a novel combination of CNN and RNN over image regions, bidirectional recurrent neural network over sentences, and a structured objective that aligns two modalities through a multimodal embedding. Donahue et al. [7] developed a novel recurrent convolutional architecture that is suitable for large-scale visual learning. The proposed model is used for video recognition, image-to-sequence generation problems, and video narration. Mao et al. [26] have proposed a method using linguistic context and visual features that are able to effectively hypothesize the semantic meaning of new words and add them to its word dictionary. Further, this dictionary is used to describe images that contain these novel concepts. They have also developed weight-sharing scheme, which improves the performance of image captioning as well as is more suitable for the novel concept learning task.

Staniute et al. [30] have presented a systematic and year-wise literature on image captioning. In this article, they have discussed precisely various encoder-decoder architecture, attention mechanisms widely used for image captioning. Zhou et al. [40] have proposed a unified version of the language pre-training model. They have used a shared multilayer transformer network as the encoder and decoder model. Cornia et al. [5] have proposed a meshed-memory transformer architecture that improves the image encoding and language generation by learning a multi-level representation of the relationship between image regions and learned prior knowledge. Liu et al. [22] have used dual generative adversely network that ensemble generation and retrieval-based approach for image captioning. Feng et al. [12] have used unsupervised learning for image captioning. They have proposed the model that does not use image and sentence pair for image captioning. Deshpande et al. [6] have used variational auto-encoder and generative adversarial networks; they

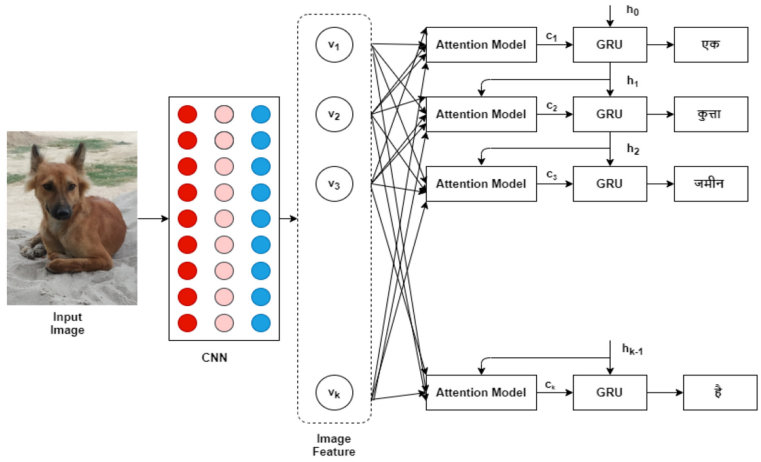

Fig. 2. Network architecture of proposed method.

have first generated a summary of an image and, later on, generated the caption based on the summary. Here, part-of-speech is used as a summary that drives the caption generation process. Anderson et al. [1] have used a top-down and bottom-up attention model for image captioning and visual question answering. Jiang et al. [16] have proposed modified encoder-decoder architecture for image captioning in which a guiding network is used on the decoder side. Guiding network models the attribute properties of the image, and its output is used to compose the input of the decoder at each timestep. All the past works have been done for image captioning in the English language. The current work is the first attempt to develop an image captioning model for the Hindi language.

3 PROPOSED METHODOLOGY

In this article, we have used conventional encoder-decoder architecture for image captioning. Recent progress in statistical machine translation has made this architecture popular. It has been shown that we can get a correct translation by maximizing the probability given an input sequence in an end-to-end fashion. In the case of language translation, an RNN is used to encode the input sentence into a fixed dimensional vector as well as to decode it to the desired sentence. Therefore, it is possible to use the same approach where the convolutional neural network (RESNET101) is used to encode the input image into a fixed dimensional vector, and the recurrent neural network (GRU) is used to decode it into the desired description. Moreover, spatial attention, visual attention, Bahdanau attention, and Luong attention further help to make the system robust (as shown in Figure 2). We have first explained encoder-decoder architecture in Section 3.1, CNN in Section 3.2, the GRU in Section 3.3, spatial attention in Section 3.4, visual attention in Section 3.5, Bahdanau attention in Section 3.6, and Luong attention in Section 3.7.

3.1 Encoder-decoder Architecture

This section contains brief introduction of encoder-decoder image captioning framework [33, 36]. The encoder-decoder model directly maximizes the probability of the correct description given an image; it directly maximizes the following objectives:

$$\theta^* = arg \max_{\theta} \sum_{(I,S)} log\, p(S|I;\theta), \tag{1}$$

where I is the image, θ is the parameter of the model, and $y = y_1, y_2 \ldots\ldots y_t$ is the corresponding caption. Here, y is the generated descriptions, and after applications of the chain rule, the log-likelihood of the joint probability distribution can be decomposed as:

$$log\, p(y) = \sum_{t=0}^{N} log\, p(y_t | y_0, y_1, \ldots\ldots\ldots\ldots y_{t-1}, I). \tag{2}$$

Here, we omit the model parameter dependency for convenience.

In conventional encoder-decoder architecture, with an RNN, each conditional probability is defined as:

$$log\, p(y_t | y_0, y_1, \ldots\ldots\ldots\ldots y_{t-1}, I) = f(h_t, c_t). \tag{3}$$

Here, f is a nonlinear function. It gives the output probability of y_t. c_t is the context vector at timestep t extracted from an Image I, h_t is the hidden state of the recurrent neural network at timestep t.

In the literature, for most of the papers using neural encoder-decoder architecture, in all the model architectures, vector c_t is an important factor that provides the evidence during caption generation [27, 33, 36, 38]. Context vector has been modeled in two different ways: vanilla encoder-decoder and attention-based encoder-decoder architecture:

- In vanilla encoder-decoder architecture, c_t is only dependent upon the CNN working as an encoder in case of image captioning. CNN is used to extract global image features from input image [27, 33]. Here, context vector, c_t, remains constant and does not depend on the hidden state of the decoder.
- In attention-based architecture, c_t depends on both the encoder and decoder architectures. Here, at each timestep, based on a hidden state, the decoder would attend a specific region of the image and compute c_t using image features from a convolution layer of CNN. In References [36, 38], it is shown that attention significantly improves the performance of encoder-decoder architecture for image captioning.

3.2 Convolutional Neural Network

CNN is used to extract features from the image. It has been widely used in computer vision for object recognition and image classification. Our model uses CNN to encode an image I into intermediate vector representation, which is extracted from a fully connected layer of CNN. This image encoding is used to initialize the initial state of the RNN language model. Here, we have used RESNET101 (Residual Neural Network) [14] to encode the image. This is a CNN model having 101 layers pre-trained on the ImageNet dataset. We have discarded the output of the last layer and stored the output of the pre-final layer, as we are interested in feature extraction instead of classification.

3.3 Gated Recurrent Unit

The most common challenge in language modeling using RNN is the vanishing gradient and exploding gradient problem [15]. To handle this problem, a special form of RNN called gated recurrent unit (GRU) was introduced by Chao et al. [4] (as shown in Figure 3)). In the literature, it has been used in sequence to sequence model for language modeling and sequence generation.

This is an updated version of RNN; it has two gates, update and reset gate, which decide the behavior of an RNN cell. These two gates determine what information should be passed to the output.

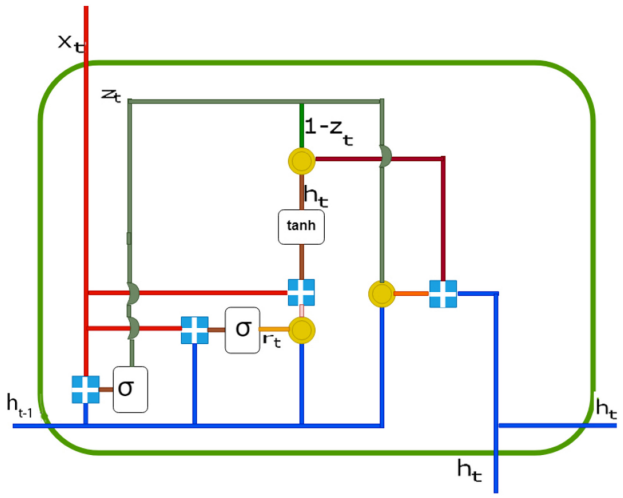

Fig. 3. Gated recurrent unit.

3.4 Spatial Attention Model

Spatial attention model is given by Lu et al. [23], which compute the context vector c_t is defined as:

$$c_t = g(V, h_t). \tag{4}$$

Here, g is attention function and $V = [v_1, v_2 \ldots\ldots\ldots v_k]$, $v_i \epsilon R^d$ is the spatial image feature vector of dimension d representing part of image. h_t is the hidden state of RNN at timestep t.

We feed the given spatial image feature, $V \epsilon R^{d \times k}$, and hidden state $h_t \epsilon R^d$ of the LSTM through a single-layer neural network followed by a softmax function to generate the attention distribution over k regions of the image:

$$z_t = w_h^T tanh(W_v V + (W_g h_t)k^T), \tag{5}$$

$$\alpha = softmax(z_t), \tag{6}$$

where, $k \epsilon R^K$ is a vector with all elements set to 1. $W_v, W_g \epsilon R^{k \times d}$, and $w_h \epsilon R^k$ are learn-able parameters. $\alpha \epsilon R^k$ is the attention weight over features in V. Context vector can be obtained using attention distribution as follows:

$$c_t = \sum_{t=0}^{N} \alpha_{ti} v_{ti}. \tag{7}$$

Here, c_t and h_t are combined to predict the next word, y_{t+1}, as given in Equation (3).

3.5 Visual Attention

This was introduced by Xu et al. [36], for image captioning. Here, the context vector is calculated using:

$$e_{ti} = f(v_i, h_t), \tag{8}$$

$$\alpha_{ti} = \frac{exp(e_{ti})}{\sum_{i=1}^{N} exp(e_{ti})}, \tag{9}$$

$$c_t = \phi(v_i, \alpha_{ti}). \tag{10}$$

Here, $V = [v_1, v_2 \ldots\ldots\ldots v_k]$, $v_i \epsilon R^d$ is the spatial image feature vector of dimension d representing part of image. h_t is the hidden state of RNN at timestep t. The weight α_{ti} is computed for

image feature v_i at timestep t by a proposed attention model f [36] that uses a multilevel perceptron applied on previous hidden state h_{t-1}, c_t is the context vector, and ϕ is function that returns a single vector given annotation vectors and their corresponding weights. Now, c_t and h_t are used together for predicting the next word, as shown in Equation (3).

3.6 Bahdanau Attention

This attention model was proposed by Bahdanau et al. [2]. It is a widely used attention model for encoder-decoder architecture. In this architecture, context vector c_t is computed as the weighted sum of image features v_i defined as:

$$c_t = \sum_{i=1}^{N} \alpha_{ti} v_i. \tag{11}$$

The weight α_{ti} for each feature v_i is computed as:

$$\alpha_{ti} = \frac{exp(e_{ti})}{\sum_{i=1}^{N} exp(e_{ti})}, \tag{12}$$

where

$$e_{ti} = f(h_t, v_i). \tag{13}$$

Here, f is proposed feed-forward neural network [2] jointly trained on all parameters, h_t is the hidden state of RNN at timestep t, and v_i is the image feature vector.

3.7 Luong Attention

Luong et al. [24] have proposed attention model for encoder-decoder architecture. In this context, model vector c_t is computed as follows:

$$c_t = \sum_{i=1}^{N} \alpha_{ti} v_i. \tag{14}$$

The weight α_{ik} for each feature v_k is computed as:

$$\alpha_{ti} = \frac{exp(e_{ti})}{\sum_{i=1}^{N} exp(e_{ti})}, \tag{15}$$

where

$$e_{ti} = h_t^{\intercal} * w * v_i. \tag{16}$$

Here, e_{ti} is referred as content-based function [24], w is the learnable weight parameter, h_t is the hidden state of RNN at timestep t, and v_i is the image feature.

The proposed method can be divided into the following parts:

- RESNET101 convolutional neural network [14] is used to extract the features from the images. It is working as an encoder to encode the images into a fixed-length vector representation.
- Spatial attention, Visual Attention, Bahdanau Attention, and Luong attention have been used, which decide where to focus in the images while generating the new word.
- GRU [4] has been used for language modeling trained on text data; it will predict the next word based on the word previously seen.

Table 1. MSCOCO Hindi Dataset Statistics

Training Set	Validation Set	Test Set
82,783 images	811 images	811 images

3.8 Hyparameters of the Model

RESNET 101 is used to extract the feature of size $196 * 512$ from input RGB images of size $224 * 224$. Input captions are fed into the embedding layer with 512 neurons; here, 0.5 dropout is used to avoid over-fitting. We have used softmax cross-entropy as loss function, Adam optimizer with a learning rate of $4e - 4$. The batch size is 128. To train our model, we have used a cluster of 8 GPU Nvidia GTX 1080, which takes approximately 14 hours with 10 epochs. To generate a caption from an image, it takes approximately 20 to 30 seconds.

4 EXPERIMENTAL SETUP

4.1 Dataset Preparation

We have created the Hindi version of the MSCOCO dataset, which is widely used for image captioning. Its annotations are based on Common Objects in Context (COCO) dataset. This is a large-scale dataset for object detection, segmentation, and captioning. COCO dataset has five different annotations used for object detection, keypoint detection, stuff segmentation, panoptic segmentation, and image captioning. COCO dataset is in JSON format, and the basic building blocks of JSON file are:

info: It has a high level of information about the dataset.

licenses: It has a list of image licenses that are applied to images in the dataset.

images: It has all the image information in the dataset. It does not have a bounding box or segmentation information. All the images have a unique image id.

annotations: Annotation file contains a list of dictionaries having the following keys: Image_id, id, and caption. Here, each image has corresponding five captions.

We have extracted all captions from the JSON file translated into Hindi. There are 82,783 images in the training set, 811 images in the validation set, and 811 images in the test set (see Table 1). Here each image has its corresponding five captions. In the training set, validation set, and testing set, there are around 4 lakh captions, 4,000 captions, and 4,000 captions, respectively. We have translated the whole corpus into Hindi using Google translator, but Google translator has following challenges while translating from the English to the Hindi:

- Meaning of the caption is lost while translating, because there is no way to incorporate the context.
- Google translation sometimes generates the translations that are grammatically incorrect.
- Accuracy of Google translation is also dependent upon the source and target language pair.

So to avoid these mistakes, this translated corpus is corrected by human annotators. Preparing the Hindi dataset took huge human effort, and approximately three to four months is required to finish correcting the entire dataset. Annotators have manually corrected many of the captions that are incorrect in any sense, and one example is shown in Figure 4 and Figure 5.

4.2 Evaluation Metric

We have used BLEU (Bilingual Evaluation Understudy Score) as an evaluation metric; it finds the similarity between reference/gold-standard caption and generates caption. BLEU is a well-known evaluation metric proposed by Papineni et al. [28] in the year 2002. It is widely used in natural

Fig. 4. Example image for dataset preparation.

English Captions	Google translated Captions in Hindi	Corrected Captions in Hindi
closeup of bins of food that include broccoli and bread	भोजन के डिब्बे को बंद करना जिसमें ब्रोकली और ब्रेड शामिल हैं	भोजन के वो डब्बे जिनमे ब्रोकोली और ब्रेड शामिल हैं उनकी पास की तस्वीर
a meal is presented in brightly colored plastic trays	एक भोजन चमकीले रंग की प्लास्टिक ट्रे में प्रस्तुत किया जाता है	भोजन चमकीले रंग की प्लास्टिक ट्रे में प्रस्तुत किया जाता है
there are containers filled with different kinds of foods	विभिन्न प्रकार के खाद्य पदार्थों से भरे कंटेनर हैं	विभिन्न प्रकार के खाद्य पदार्थों से भरे कंटेनर हैं
a bunch of trays that have different food	ट्रे का एक गुच्छा जिसमें अलग भोजन होता है	ट्रे का एक समूह जिसमें अलग अलग भोजन है
colorful dishes holding meat vegetables fruit and bread	मांस सब्जी फल और रोटी पकड़े रंगीन व्यंजन	रंगीन व्यंजन जिनमे मांस सब्जी फल और रोटी है

Fig. 5. Example of dataset preparation.

language processing and computer vision for machine translation, image captioning, and so on. BLEU score is calculated by matching comparison between n-grams of the candidate with the n-grams of reference translation. Mathematical details of BLEU can be understood with the following example of machine translation:

Let us take the following assumption: H is the source sentence that is written in Hindi language. R1 and R2 are two reference sentences written by human annotators. C is the candidate sentence generated from the machine translation (MT) system.

H = एक कुत्ता टेबल पे बैठा है ।

R1 = The dog is on the table.

R2 = There is a dog on the table.

C = The The The The The The The

The cornerstone of the BLEU score metric is a well-known precision measure. Precision can be calculated by simply counting the number of candidate translation words (unigrams). One way to evaluate the C is the precision: comparing the number of common words between C and a reference sentence. Therefore, precision will be:

$$precision(p) = \frac{No\ of\ words\ in\ MT\ output\ which\ appear\ in\ reference\ sentences}{total\ no\ of\ words\ in\ MT\ output}. \quad (17)$$

Therefore, $p = \frac{7}{7} = 1$. This idea of using precision is not useful, as precision is very high whereas generated MT output is incorrect.

To overcome this problem, Papineni et al. [28] have used modified unigram precision in their paper. To compute this, one should count the maximum number of times a word has appeared in any reference sentence (maximum reference count). Further, the total count of each word in the candidate sentence is clipped to the maximum reference sentence count. Here, for example: “the”

has appeared one time in R1 and two times in R2; therefore, the modified unigram precision will be $p = \frac{2}{7}$

We can evaluate the MT system on the entire corpus, but the basic evaluation unit will be the sentence; it means we compute the n-gram matches sentence-by-sentence. To compute the modified precision of a test corpus, we add the total clipped n-gram count for all candidates, and it is divided by the number of candidate n-grams in the test corpus. This modified precision can be generalized with respect to n-grams as follows:

$$p_n = \frac{\sum_{S \in candidate\ sentences} \sum_{n-gram \in S} Count_{clip}(n-gram)}{\sum_{S' \in candidate\ sentences} \sum_{n-gram' \in S'} Count_{clip}(n-gram')}. \tag{18}$$

Papineni et al. have calculated geometric mean of test corpus's modified precision score, then multiplied this with the brevity penalty factor. Here, the brevity penalty is used to match the high scoring candidate translation with a reference translation in length. First, geometric average of modified n-gram precision is computed by using n-grams up to length N and positive weights w_n. Here, the sum of positive weights equals to one.

Now, let us assume that c is the length of the candidate translation, and r is the length of the reference corpus. Then brevity penalty (BP) is defined as:

$$BP = \begin{cases} 1, & if c > r \\ e^{1-\frac{r}{c}}, & if c \leq r, \end{cases}$$

then

$$BLEU = BP.exp\left(\sum_{n=1}^{N} w_n log(p_n)\right). \tag{19}$$

5 RESULTS AND DISCUSSION

This section reports the evaluations of generated captions using qualitative and quantitative analysis. To the best of our knowledge, there does not exist any work on image captioning in the Hindi Language.

As there was no existing work for image captioning in the Hindi language, we have compared our results with different baseline models, which are frequently used in image captioning in the English language.

- Baseline-1: In this baseline, RESNET 101 is used as an encoder for feature extraction from an image, and LSTM is used as a decoder for language modeling. Here, the extracted feature is used as an input to the decoder. RESNET 101 deep neural network can avoid the problem of vanishing and exploding gradient in deep neural networks. It helps in minimizing training errors. Finally, the decoder generates the caption by reading the encoded image feature.
- Baseline-2: In this, we have used Inception V4 [32] as an encoder and LSTM as a decoder. The inception V4 model is proposed by Google to increase the performance in terms of speed and accuracy.
- Baseline-3: Here, Inception V4 [32] is used as an encoder, and GRU is used as a decoder.
- Baseline-4: In this baseline, RESNET 101 is used as encoder and GRU as a decoder, as this was the best model among various encoder-decoder architectures for English caption generation, so we have translated generated English caption into Hindi (as shown in Figure 7).

5.1 Qualitative Analysis

This section illustrates the generated captions of test images. Generated captions are shown in Figure 6. Here, the generated captions very closely describe actual objects and activity within

(a) एक व्यस्त सड़क पर एक चौराहे पर एक कार (b) एक टेनिस खिलाड़ी ने टेनिस कोर्ट पर एक रैकेट को घुमाया (c) एक जिराफ़ एक बाड़ के बगल में एक गंदगी के मैदान में खड़ा है

(d) एक नदी के किनारे पर नावों का एक समूह (e) एक सड़क पर एक ट्रैफिक लाइट और एक लाल बत्ती (f) सर्फबोर्ड के साथ समुद्र में एक लहर की सवारी करने वाला व्यक्ति

(g) एक गाय एक गाय के पास खड़ी है (h) एक स्केट पार्क में एक स्केटबोर्ड पर एक आदमी (i) एक शौचालय और एक शौचालय के साथ एक बाथरूम

Fig. 6. Generated captions on test images.

the image. In this section, we have done an analysis of the quality of generated captions based on adequacy and fluency and error analysis. We have accomplished the qualitative analysis for spatial attention only as all the attention models have similar performance in terms of BLEU score; hence, similar conclusions can be drawn from other attention mechanisms also.

5.1.1 Adequacy and Fluency. We have accomplished human evaluation of generated captions using adequacy and fluency metrics. *Human evaluation* is one of the important evaluation

Table 2. Different Scales of Adequacy and Fluency

Scale	Meaning
0	Poor
1	Bad
2	Moderate
3	Good
4	Excellent

techniques used in machine translation and various natural language processing problems. Here, adequacy reveals information preserved in the generated caption, and fluency evaluates generated captions based on grammatical rules. In another way, adequacy and fluency can be understood as:

Adequacy: Is all the information preserved in the generated caption? Fluency: Is the generated caption grammatically valid as per the Hindi grammar rules?

Adequacy: This is calculated in the range of [0–4] as shown in Table 2; the meanings of different values are provided below:

- Score 0: Poor: None of the information is preserved in the generated caption.
- Score 1: Bad: Little information is preserved in the generated caption.
- Score 2: Moderate: Much of the information is preserved in the generated captions.
- Score 3: Good: Most of the information is preserved in the generated caption.
- Score 4: Excellent: All of the information is preserved in the generated caption.

Fluency: This is again calculated in the range of [0–4], as shown in Table 2. The interpretations of different values are provided below:

- Score 0: Poor: Generated Hindi caption is incomprehensible.
- Score 1: Bad: Generated Hindi caption is dis-fluent.
- Score 2: Moderate: Generated Hindi captions are like non-native Hindi.
- Score 3: Good: Generated Hindi captions are good in terms of Hindi grammar rules.
- Score 4: Excellent: Generated Hindi captions are flawless Hindi sentences, and all are correct in terms of Hindi grammar rules.

This human evaluation is done by two annotators with an inter-annotator agreement of 84%.

Here, we have computed adequacy and fluency of captions generated by two different models:

- Proposed architecture is trained on the Hindi dataset.
- Proposed architecture trained on English dataset generating English captions, which are further translated into Hindi captions.

We have evaluated adequacy and fluency values for 1,000 captions generated by our proposed method and then considered their average values. We have done the comparison between the model trained on the Hindi data and the model trained on English data in which generated captions are further translated into Hindi using Google translator. From the obtained values of adequacy and fluency, as shown in Table 3 and Table 4, it is evident that our approach (generating captions using a model trained on Hindi dataset) is better than the approach where we generate English captions and later on those captions are translated into Hindi. We can also observe that the quality of the generated caption by the proposed model (C1) is better than the caption C3 (as shown in Figure 7).

Table 3. Adequacy and Fluency Values of 1,000 Captions, Generated by Our Model Trained on Hindi Dataset

Adequacy	Fluency
3.193	3.413

Table 4. Adequacy and Fluency Values of 1,000 Captions Generated by Model Trained on English Dataset; Later on Generated English Captions are Translated Using Google Translator in Hindi

Adequacy	Fluency
2.373	2.108

5.1.2 Error Analysis. This section reports a detailed analysis of errors in generated captions.

- **Error in object recognition:** In Figure 6(a), the generated caption has correctly described the image, but the "Man" in the image has not been recognized, as feature similarity model has attended car first in the input image. In Figure 6(c), the model has accurately recognized giraffe, and it has mentioned dirty ground in the caption because the dirty ground is mostly associated with the giraffe in the dataset. In Figure 6(g), the model has reported about a single cow in the caption; this is because the input image is blurry, and objects are not clear. In Figure 6(i), the generated description is almost correct, but it has predicted the bathroom also because most of the time the word "toilet" is associated with bathroom in the corpus.
- Error in Activity: In Figure 6(h), the model has predicted objects successfully, but anticipated activity is not correct; this may be because there are very few images where the word "jumping" has been done by the "skateboard," so the probability of its prediction is very low.
- Correctly generated captions: In Figure 6(b), the generated caption is almost close to the activity. In Figure 6(d), the generated caption is almost correct; model successfully recognized boats in the input image. In Figure 6(e), generated captions correctly describe the object and the activity. In Figure 6(f), generated caption accurately predicts the object and the activity.

5.2 Quantitative Analysis

Although we can evaluate the quality of the generated caption manually, a subjective score is required to describe the quality of the generated captions. In this section, we have evaluated the generated captions with respect to the reference captions. We have used the BLEU score as an evaluation metric; it is the most popular evaluation metric in machine translation. We have also conducted statistical t-test to prove the statistical significance of our proposed method. Here, We have accomplished the statistical test for spatial attention only as all the attention models have similar performance in terms of BLEU score, hence similar conclusion can be drawn for other attention also.

There are 811 images in the test set of the dataset, and we have tested the performance of the proposed architectures with different attention on test images. BLEU-1, BLEU-2, BLEU-3, and BLEU-4 scores are calculated, and we have compared those values with other baseline models, which are shown in Table 5. Results show that our proposed model performs better than other baseline models.

(A)

C1: एक टेनिस खिलाड़ी एक गेंद को हिट करने के लिए तैयार हो रहा है
C2: a man is playing tennis on a court with a tennis racket
C3: एक आदमी एक टेनिस रैकेट के साथ एक अदालत में टेनिस खेल रहा है

(B)

C1: एक हवाई जहाज एक बादल के दिन आकाश में उड़ता है
C2: a plane flying through a cloudy sky
C3: एक बादल के आकाश के माध्यम से उड़ान भरने वाला विमान

(C)

C1: एक केला और एक फल के साथ फल का एक कटोरा
C2: a banana and a banana sitting in a bowl
C3: एक केला और एक केला एक कटोरी में बैठे

(D)

C1: एक इमारत के शीर्ष पर एक घड़ी के साथ एक लंबा टॉवर
C2: a clock tower with a clock on its face
C3: इसके चेहरे पर घड़ी के साथ एक घड़ी टॉवर

C1: Generated caption based on model trained on Hindi dataset
C2: Generated caption based on model trained on English dataset
C3: Generated caption based on model trained on English dataset then translated into Hindi.

Fig. 7. Generated captions of test images with different models.

We have explored the proposed method with various attention mechanisms also on raw data, which are translated by Google translator only. BLEU scores obtained are shown in Table 6. It can be concluded from Table 5 and Table 6 that correction done by human annotators helps in obtaining better performance.

5.2.1 Performance of Proposed Method with Various Attention Models on the English Corpus. The proposed architecture has been tested on the English corpus. Here, we have trained our model

Table 5. Performance of the Proposed Method with Different Attention Mechanism Trained on Corrected Corpus, and Its Comparison with Different Baselines

State-of-the-Art	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Spatial Attention [23]	66.1	46.7	31.6	20.8
Visual Attention [36]	66.9	47.2	31.6	20.8
Bahdanau Attention [2]	67.0	47.8	31.9	21.2
Luong Attention [24]	65.7	46.6	31.4	20.3
Baseline 1	63.3	44.6	30.9	20.4
Baseline 2	62.2	43.0	29.2	19.3
Baseline 3	62.5	43.4	28.3	19.4
Baseline 4	63.4	44.1	30.0	20.2

Table 6. Performance of the Proposed Method with Different Attention Mechanism Trained on Raw Corpus

Attention Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Spatial Attention [23]	64.3	44.7	29.5	19.1
Visual Attention [36]	63.9	45.8	31.0	20.3
Bahdanau Attention [2]	65.9	47.1	31.9	21.0
Luong Attention [24]	64.2	45.7	31.0	20.3

Table 7. Performance of Proposed Method and Various Attention Mechanisms Trained on English Corpus

Attention Model	BLUE-1	BLEU-2	BLEU-3	BLEU-4
Spatial Attention [23]	69.1	48.8	34.1	23.8
Visual Attention [36]	69.6	49.6	34.9	24.7
Bahdanau Attention [2]	69.3	48.9	33.8	23.3
Luong Attention [24]	69.7	48.9	33.7	23.2

on the original MS COCO dataset, which is in the English language. BLEU scores obtained are shown in Table 7.

5.2.2 Statistical Test. We have conducted Welch's t-test [34] hypothesis at 5%(0.05) significance level. This test is accomplished to indicate that the best BLEU score obtained by our proposed method is statistically significant and has not happened by chance (as shown in Table 8). For the MSCOCO dataset, the evaluation metrics are calculated by 10 consecutive runs of our proposed method and other state-of-the-art methods. In all these runs, all the methods are trained using the training set and tested on the test set (as shown in Table 1). To establish the statistical significance of proposed method, the p-value is [8] calculated using Welch's t-test [34] to accomplish the comparison between two groups. One group belongs to the list of BLEU scores of our method, and the other group belongs to a list of BLEU scores of other state-of-the-art methods. In the null hypothesis (Δ), it is assumed that there is no difference between the average BLEU scores of both the methods:

$$\Delta_0 : \eta_1 = \eta_2 = \eta_3. \tag{20}$$

Table 8. p-values Produced by Welch's t-test Comparing Our Proposed Method with Other Baselines

State of Arts	BLEU-1 Score	BLEU-2 Score	BLEU-3 Score	BLEU-4 Score
Baseline 1	1.26044e-58	6.20838e-54	6.20838e-54	1.0313e-27
Baseline 2	1.08494e-54	3.46869e-63	4.14083e-56	1.79601e-48
Baseline 3	9.7493e-63	2.59167e-61	2.59167e-61	2.34115e-47
Baseline 4	4.95048e-58	2.04615e-57	1.62021e-49	6.86249e-34

On the contrary, there is an opposite (Δ_1) hypothesis that there is a significant difference between average BLEU scores of any two methods:

$$\Delta_1 : \exists \gamma_1, \gamma_2 : \gamma_1 \neq \gamma_2 \Rightarrow \eta_{\gamma_1} \neq \eta\gamma_2. \tag{21}$$

Where Δ_k is the average BLEU score of kth method. Now, the t-statistic formula is used to calculate the difference between the average BLEU scores:

$$t = \frac{\overline{\chi_1} - \overline{\chi_2}}{\sqrt{\frac{\sigma_1^2}{\mu_1} + \frac{\sigma_2^2}{\mu_2}}}. \tag{22}$$

Where χ, $\sigma_i 2$ and μ_i are the mean, variance, and size of the ith example, respectively. Here, the p-value denotes the probability with the assumption of the null hypothesis, and a smaller p-value indicates strong evidence against this null hypothesis. Statistical test results are shown in Table 8, where all the values are less than 0.05 (5% significance level), which indicates better performance of our proposed method in comparison to other methods.

6 CONCLUSIONS AND FUTURE WORK

We have presented spatial attention, visual attention, Bahdanau attention, and Luong attention-based model for image captioning in the Hindi Language. We have used encoder-decoder architecture similar to machine translation to solve this task. A manually annotated Hindi image captioning dataset is also generated as a part of this work. The generated captions are somewhat identical to input images, correctly describing the object and the activity, but still, there is the scope of improvement.

Some of the application areas of our proposed model are the following: (1) tweet classification: during any disaster event, many people used to tweet about the situations. Automatically classifying these tweets into two categories, informative and non-informative, is a difficult task. Moreover, tweets are often associated with images to describe the situation in a more comprehensive way. Tweets are shorter in length. Thus, the captions generated for the images associated with the tweets can be utilized as the additional context information. If the tweets are available in the Hindi language, our proposed model can be utilized to generate captions in Hindi. (2) Development of a caption generation model for natural disaster images, as there is no image captioning work available in the literature for such images. In the future, we will extend our work for dense image captioning to generate more than one caption for an input image. The dense image captioning task can be further extended for paragraph generation given an input image. We will also work for medical report generation, given a medical image in Indian languages.

REFERENCES

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[3] Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. 2007. Large language models in machine translation. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL'07)*. 858–867.

[4] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[5] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. 2020. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'20)*.

[6] Aditya Deshpande, Jyoti Aneja, Liwei Wang, Alexander G. Schwing, and David Forsyth. 2019. Fast, diverse and accurate image captioning guided by part-of-speech. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'19)*.

[7] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2625–2634.

[8] Pratik Dutta and Sriparna Saha. 2017. Fusion of expression values and protein interaction information using multi-objective optimization for improving gene clustering. *Comput. Biol. Med.* 89 (2017), 31–43.

[9] Desmond Elliott and Frank Keller. 2013. Image description using visual dependency representations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 1292–1302.

[10] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K. Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C. Platt, et al. 2015. From captions to visual concepts and back. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1473–1482.

[11] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. 2010. Every picture tells a story: Generating sentences from images. In *Proceedings of the European Conference on Computer Vision*. Springer, 15–29.

[12] Yang Feng, Lin Ma, Wei Liu, and Jiebo Luo. 2019. Unsupervised image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'19)*.

[13] Jane Gary and Carl Rubino. 2001. *Facts About the World's Languages: An Encyclopedia of the World's Major Languages*. H.W. Wilson, NY.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[15] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput* 9, 8 (1997), 1735–1780.

[16] Wenhao Jiang, Lin Ma, Xinpeng Chen, Hanwang Zhang, and Wei Liu. 2018. Learning to guide decoding for image captioning. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

[17] Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3128–3137.

[18] Girish Kulkarni, Visruth Premraj, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C. Berg, and Tamara L. Berg. 2011. Baby talk: Understanding and generating image descriptions. In *Proceedings of the 24th IEEE Conference on Computer Vision and Pattern Recognition*. Citeseer.

[19] Polina Kuznetsova, Vicente Ordonez, Alexander C. Berg, Tamara L. Berg, and Yejin Choi. 2012. Collective generation of natural image descriptions. In *Proceedings of the 50th Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 359–368.

[20] Rémi Lebret, Pedro O. Pinheiro, and Ronan Collobert. 2014. Simple image description generator via a linear phrase-based approach. *arXiv preprint arXiv:1412.8419* (2014).

[21] Siming Li, Girish Kulkarni, Tamara L. Berg, Alexander C. Berg, and Yejin Choi. 2011. Composing simple image descriptions using web-scale n-grams. In *Proceedings of the 15th Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 220–228.

[22] Junhao Liu, Kai Wang, Chunpu Xu, Zhou Zhao, Ruifeng Xu, Ying Shen, and Min Yang. 2020. Interactive dual generative adversarial networks for image captioning. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 11588–11595.

[23] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. 2017. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 375–383.

[24] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).

[25] Haley MacLeod, Cynthia L. Bennett, Meredith Ringel Morris, and Edward Cutrell. 2017. Understanding blind people's experiences with computer-generated captions of social media images. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 5988–5999.

[26] Junhua Mao, Xu Wei, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan L. Yuille. 2015. Learning like a child: Fast novel visual concept learning from sentence descriptions of images. In *Proceedings of the IEEE International Conference on Computer Vision*. 2533–2541.

[27] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. 2014. Deep captioning with multimodal recurrent neural networks (m-RNN). *arXiv preprint arXiv:1412.6632* (2014).

[28] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 311–318.

[29] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. 2013. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229* (2013).

[30] Raimonda Staniūtė and Dmitrij Šešok. 2019. A systematic literature review on image captioning. *Appl. Sci.* 9, 10 (2019), 2024.

[31] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the International Conference on Advances in Neural Information Systems*. 3104–3112.

[32] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. 2017. Inception-v4, inception-ResNet and the impact of residual connections on learning. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*.

[33] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3156–3164.

[34] Bernard L. Welch. 1947. The generalization of students' problem when several different population variances are involved. *Biometrika* 34, 1/2 (1947), 28–35.

[35] Qi Wu, Chunhua Shen, Lingqiao Liu, Anthony Dick, and Anton Van Den Hengel. 2016. What value do explicit high level concepts have in vision to language problems? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 203–212.

[36] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International Conference on Machine Learning*. 2048–2057.

[37] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. 2017. Comparative study of CNN and RNN for natural language processing. *arXiv preprint arXiv:1702.01923* (2017).

[38] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4651–4659.

[39] Liang Zhou and Eduard Hovy. 2004. Template-filtered headline summarization. In *Proceedings of the Text Summarization Branches Out*. 56–60.

[40] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J. Corso, and Jianfeng Gao. 2020. Unified vision-language pre-training for image captioning and VQA. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 13041–13049.

Received September 2019; revised July 2020; accepted October 2020