Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.024118
Article

Tech Science Press

Image Captioning Using Detectors and Swarm Based Learning Approach for Word Embedding Vectors

B. Lalitha[1,*] and V. Gomathi[2]

[1]CSE Department, Sethu Institute of Technology, Pulloor, Kariapatti, 626115, India
[2]CSE Department, National Engineering College, K.R. Nagar, Kovilpatti, 628503, India
*Corresponding Author: B. Lalitha. Email: lalli_j@yahoo.com
Received: 05 October 2021; Accepted: 16 December 2021

Abstract: IC (Image Captioning) is a crucial part of Visual Data Processing and aims at understanding for providing captions that verbalize an image's important elements. However, in existing works, because of the complexity in images, neglecting major relation between the object in an image, poor quality image, labelling it remains a big problem for researchers. Hence, the main objective of this work attempts to overcome these challenges by proposing a novel framework for IC. So in this research work the main contribution deals with the framework consists of three phases that is image understanding, textual understanding and decoding. Initially, the image understanding phase is initiated with image pre-processing to enhance image quality. Thereafter, object has been detected using IYV3MMDs (Improved YoloV3 Multishot Multibox Detectors) in order to relate the interrelation between the image and the object, and then it is followed by MBFOCNNs (Modified Bacterial Foraging Optimization in Convolution Neural Networks), which encodes and provides final feature vectors. Secondly, the textual understanding phase is performed based on an image which is initiated with preprocessing of text where unwanted words, phrases, punctuations are removed in order to provide a healthy text. It is then followed by MGloVEs (Modified Global Vectors for Word Representation), which provides a word embedding of features with the highest priority towards the object present in an image. Finally, the decoding phase has been performed, which decodes the image whether it may be a normal or complex scene image and provides an accurate text by its learning ability using MDAA (Modified Deliberate Adaptive Attention). The experimental outcome of this work shows better accuracy of shows 96.24% when compared to existing and similar methods while generating captions for images.

Keywords: Denoising; improved YoloV3 multishot multibox detector (IYV3MMD); modified bacterial foraging optimization in convolutional neural network (MBFOCNN); modified global vectors for word representation (MGloVE); modified deliberate adaptive attention (MDAA); encoder; decoder

1 Introduction

In the current world driven by social media, people easily produce and share rich social multimedia material online including photos and videos as they are allowed by social media players like Twitter, Amazon, Facebook and Google News. This voluminous data is explored by researches in terms of displays, retrievals, and alterations of multimedia content, specifically in IC as it tries to evaluate the visual content of input images and creates captions that verbalize images based on their most essential elements [1]. Few examples of multimodal applications used in data processing include Visual question answering, multimodal event extractions, video captioning, and cross-modal image retrievals [2,3].

Unlike image classifications or object identifications in n computer vision, ICs are multi-modal tasks that demand capture important features of images. This study links NLPs (Natural Language Processing) to ICs for describing them. ICs are processed using encoder-decoder architectures [4,5] where dense pixel-level information is encoded using the CNNs encoder and translated by decoders.

Specific visual data processing models are based on CNN's LSTMs (Long Short Term Memories) framework which can be trained [6,7]. Many collaborative CNN models have also been proposed for ICs: R-CNNs (Regional CNNs) [8,9], Fast R-CNNs (fast regional CNNs) [10], Mask R-CNNs (mask regional CNNs) [11,12], and FCN-LSTMs (Fully Convolution Network Long Short Term Memory) [13]. Mechanisms using CNN feature maps for Spatial processing have been studied more recently [14,15], where they produce spatial maps emphasizing on important picture regions and matched to words.

In order to execute ICs, these present computer vision tasks must not only collect information contained in images but also extract semantic associations of acquired visual information corresponding to verbal expressions [16,17]. Though these approaches listed above have shown substantial results, they also have resulted in a number of flaws [18,19]. In order to handle these issues, this research work proposes an efficient IC processing technique using MDAAs with the help of the Image feature and word embedding vectors using MBFOCNNs and MGloVEs.

The following is how the rest of the article is organized: Section 2 reviews and analyzes relevant ICs work, Section 3 details the suggested technique, and Section 4 explains the experimental evaluation. Lastly, Section 5 concludes planned study with future prospects.

2 Literature Survey

Xiao et al. [20] proposed deep hierarchical encoder-decoder networks for ICs, in which the encoder and decoder operations were separated using a deep hierarchical structure. The method was able to leverage deep networks' representation capabilities to combine high-level semantics of vision and language to create captions. Visual representations at the highest abstraction level were studied at the same time, and each of these levels was assigned to a single LSTM. The textual inputs were encoded using the bottom-most LSTM. The intermediate layer was used in the encoder-decoder to improve the decoder capabilities of the top-most LSTM. Tests on three benchmark databases showed that study's approach worked effectively beating existing modern techniques: Flickr8K, Flickr30K, and MSCOCO. The issue was in using it for complicated situations with many targets.

For the remote sensing issues in ICs, Shen et al. [21] suggested VRTMM. Initially, it worked with the Varied Auto encoder to fine-tune the CNNs. Secondly, the text description was created by the Transformers using both geographical and semantic data. The quality of the produced phrases was then improved using Reinforcement Learning. For Remote Sensing Image Caption Database, their method outperformed others by significant margin on entire seven values. Findings of experiment indicate that appraoch worked well with remote sensing ICs and produced new outcomes as other captioning methods ignored interrelationships between items in images.

Kinghorn et al. [22] created deep networks that included two critical phases for picture description creations as well as initial regional based developments. Their Region Proposal Networks from Faster R-CNNs generated initial regional proposals. The technique created areas of interest, which were subsequently utilized to annotate and categorize human and object characteristics. System's initial major phase involved creating label descriptions for each location of interest. In the second step, they used encoders-decoders based on RNNs (Recurrent Neural Networks) to convert these regional descriptions into a comprehensive image description. Their empirical findings showed that their strategy was equivalent to many previous studies while outperforming many approaches. Furthermore, when the number of time steps grew, RNNs had gradient vanishing issues.

Su et al. [23] integrated visual and high-level semantic information in their proposal for ICs. The bottom layer and top layers of a hierarchical DNN were created for caption creations where the former collected visual and high-level semantic information from identified areas in images while the latter combined using an adaptive attention method. On the MSCOCO dataset, their experimental findings performed competitively in comparison to other techniques. It was challenging to extract the essential characteristics from images, which required a mix of visual and language data.

Zhao et al. [24] developed a Multimodal fusion technique for generating descriptions that describe image information. The study used CNNs for image feature extractions and an attention model for image attributes extractions and language CNNs to model sentences, and a recurrent networks like LSTM for word predictions. When compared to other approaches, to model long-term interdependences of historical words, the study used image characteristics to enhance image representations and handle all prior words. However, the multimodal fusion was created using a single-layer network, which proved incapable of performing complex tasks.

ICs also include an attention function. Xu et al. [25] employed deterministic soft attention scheme while creating different words and stochastic hard attention to assist decoder's attention on highly important image regions and thus improving sentence creation quality.

Anderson et al. [26] used a bottom-up and top-down approach to enhance attention modules. Attention mechanism efficiently plugged gaps between the visual and language domains. As a result, this technique is frequently used in ICs activities.

Vaswani et al. [27] developed a Transformer model where attention blocks were stacked completely without convolutions or repetitions. The study included an encoder and a decoder. Encoders had a self-attention and a position-wise feed-forward block, whereas decoders had a self-attention and a cross-attention layer.

Zhu et al. [28] proposed a framework by modifying Transformer architecture where CNNs replace the Transformer's encoder. The spatial connections of the R-CNN identified item pairs are incorporated into the attention block in the study by Herdade et al. [29] while Huang et al. [30] presented "Attention on Attention" module to describe interactions between image objects on encoders and also refined decoder with self-attentions.

Pan et al. [31] offer a unified X-Linear attention block which fully utilizes bilinear pooling to appropriately exploit diverse visual inputs. Cornia et al. [32] proposed a Meshed Transformer with IC operations memory.

Based on the above studies, this work leverages on low/high image characteristics, for employing mesh like connections in decoding images.

3 Proposed Encoder-Decoder IC Framework

IC applications have several uses including Image retrieval, assisting visually handicapped, and intelligent human-computer interaction as they automatically create language descriptions for images. It was a difficult cross-disciplinary project that required both computer vision and NLPs in processing. Various DL algorithms has recently developed, however focusing on certain areas or objects of interest in an image is a complex issue specifically during sentence productions while disregarding previously created time steps that constitute sentences. Models may pay more attention to image's same locations in time steps and thus jeopardizing IC performances. This research work proposes an efficient IC processing technique (Illustrated in Fig. 1) by utilizing MDAAs with the aid of the image features and word embedding vectors which use MBFOCNNs and MGloves to handle aforesaid issues.

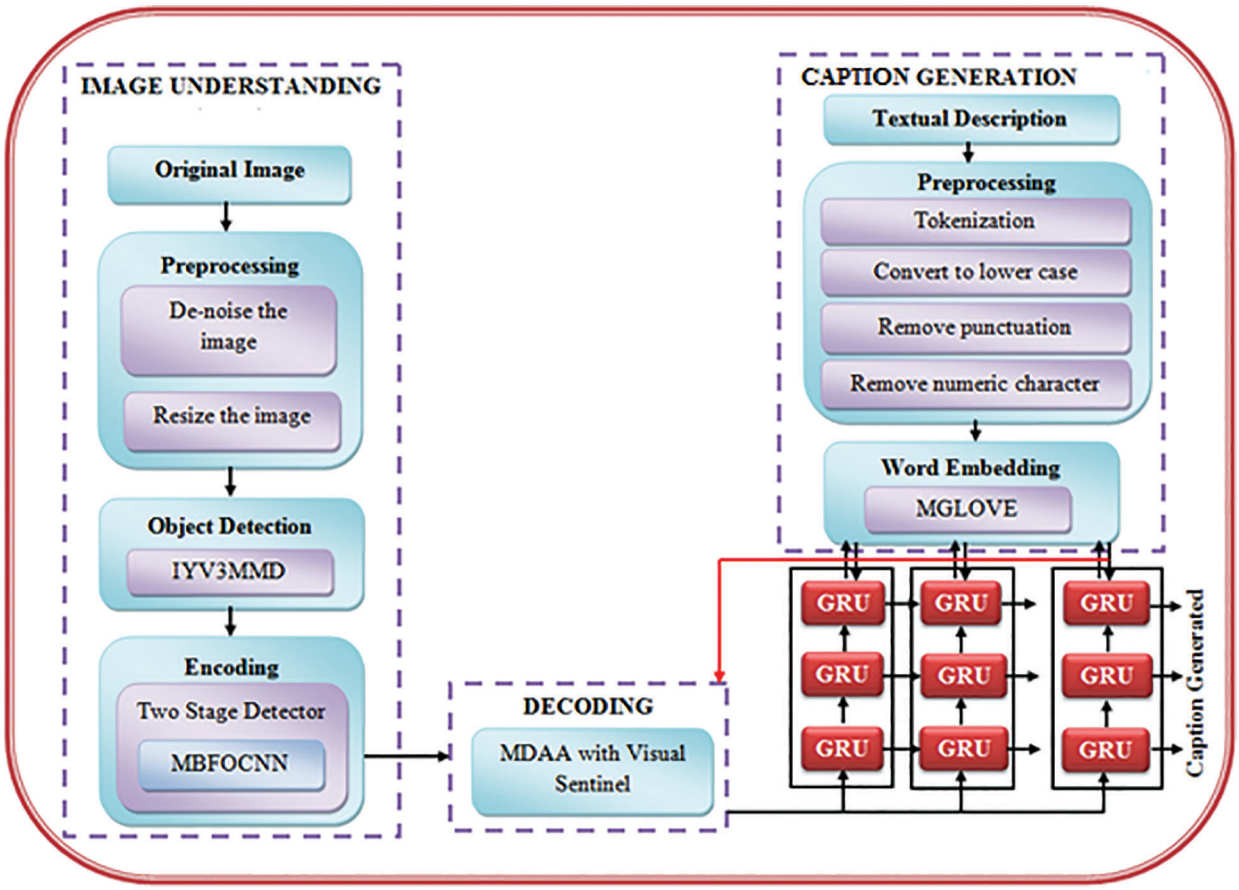

Figure 1: Proposed IC framework

3.1 Image Understanding

In Image understanding phase, initially a raw image will be processed for enhancement and size alignment. After standardizing the size as 512*512, it will undergo object detection stage and followed by encoding the detector outcomes as the text caption is generated.

3.1.1 Object Detection

Object detections identify items in images for focusing on closest ones while keeping farther ones out of focus. For accurate IC operations, it is necessary to identify minute things where IYV3MMDs are used in this

work. IYV3MMDs use a convolution feed-forward networks to generate bounding boxes of fixed-size array and boxes are scored for the presence of objects, proceeded by non-maximum suppression phase for final detection outcomes. This work's object detections are based on selective training on a default detection boxe group and sizes, along with hard negative mining and data augmentation methods. The IYV3MMD training stages are detailed below:

Step 1: Matching Strategy

Determine default boxes that relate to basic truth detections during training and train networks accordingly. Default boxe groups for each ground truth box was selected in the study based on varying positions, aspect ratios, and scales. Initially, every ground truth box is coordinated to default box with greatest jaccard overlaps followed by matching default boxes to any ground truths with jaccard overlap values greater than threshold and unlike Multi-Boxes (0.5). This strategy simplifies learning as it allows networks to forecast maximum values for many overlying default boxes instead of just ones with greatest overlaps. This is depicted as Eq. (1).

$$Mat_{i,j}^{Stat} = \begin{cases} 1, & if\,(x_{i,j}^{\Gamma} \geq 0.5) \\ 0, & otherwise \end{cases} \tag{1}$$

where, $Mat_{i,j}^{Stat}$ stands for matching strategies and $x_{i,j}^{\Gamma}$ implies matching i^{th} default box to j^{th} ground truth box of Γ image category.

Step 2: Training Objective

Training is necessary to detect multiple items. Assuming $x_{i,j}^{\Gamma} = \{1, 0\}$ is matching indicator for i-th default box and j-th ground truth box in category p, then $\sum x_{i,j}^{\Gamma} \geq 1$. The total objective loss function is represented by Eq. (2) as the weighted sum of localization loss (loc) as well as confidence loss (conf).

$$\Omega_L(x,\ C,\ P,\ \mathrm{T}) = \frac{1}{N}(\Omega_{Conf}(x,\ C) + \beta\Omega_{Loc}(x,\ P,\ \mathrm{T})) \tag{2}$$

here, N implies coordinated default boxes. When $N = 0$, loss is 0. Loss due to localizations is a smoothened loss among predicted boxes (P) and ground truth boxes (T). Regressing offsets of the centre (Cx, Cy) in default bounding box (d) with width (w) and height (h) can be depicted using the following Equations.

$$\Omega_{Loc}(x,\ P,\ \mathrm{T}) = \sum_{i \in Pos}^{N} \sum_{\mathrm{M} \in \{Cx,Cy,w,h\}} x_{i,j}^{k} smooth(P_i^M - \mathrm{T}_j^M) \tag{3}$$

$$\Omega_{Conf}(x,\ C) = -\sum_{i \in Pos}^{N} x_{i,j}^{\Gamma} \log(C_i^{\Gamma}) - \sum_{i \in Neg} \log(C_i^0) \qquad where,\ C_i^{\Gamma} = \frac{\exp(C_i^0)}{\sum_p \exp(C_i^0)} \tag{4}$$

where β stands for a term's weight and is equal to 1 in cross validations.

Step 3: Selecting Scales and Aspect Ratio for Default Boxes

The work employs scale selections in lower/higher feature maps to identify default box size fluctuations or in images to smoothed sizes. The default boxes need not match each layer's real receptive fields. This work aims at tiling default boxes in such a way that particular mappings of functions learn to adapt to different object sizes. Scaling is done for each pixel's feature map, computed using Eq. (5)

$$\Gamma_k = \Gamma_{\min} + \frac{\Gamma_{\max} - \Gamma_{\min}}{M - 1}(k - 1), \qquad k \in [1,\ M] \tag{5}$$

here, Γ_{max} stands for upper bound values and Γ_{min} for lower bound values and layers in between these bounds are evenly spaced.

Step 4: Hard Negative Mining

Most of the default boxes show negativity after the matching stage, particularly whenever there are a lot of default boxes to choose from. The results in an imbalanced training and significant losses. To overcome this issue, this research work uses sorting that selects the largest confidence loss for every default box and thus resulting in a negative-to-positive ratio value greater than 3:1 and faster optimizations/constant training.

Step 5: Augmentation of Data

To make the system highly tolerant to varying input item sizes and shapes, every training image is sampled randomly using one of following parameters:

- Using complete original input images as starting points.
- Sampling patches with objects and a minimum jaccard overlap of 0.1, 0.3, 0.5, 0.7
- Selecting patches randomly.
- Sampling patches in the interval [0.1, 1] of actual image size, with an aspect ratio ranging from 1/2 to 2.
- Preserving centre of overlapping ground truth boxes in sampled patches.
- Enlarging each sampled patch to a predetermined size and horizontally flipping it with a probability of 0.5 after executing the aforementioned sampling steps.

3.1.2 Encoder

Encoding contributes towards a rich representation of image via encoding input image content to a fixed-length vector using an internal representation. The existing work mainly used RNN for processing the encoding but due to inaccurate representation of the image, the work has developed a MBFOCNN as shown in Fig. 2. By integrating the image to a fixed-length vector, the proposed encoder delivers a significant quality improvement of image. MBFOCNN is performed based on the input ($\zeta_{i,j}^{Od}$).

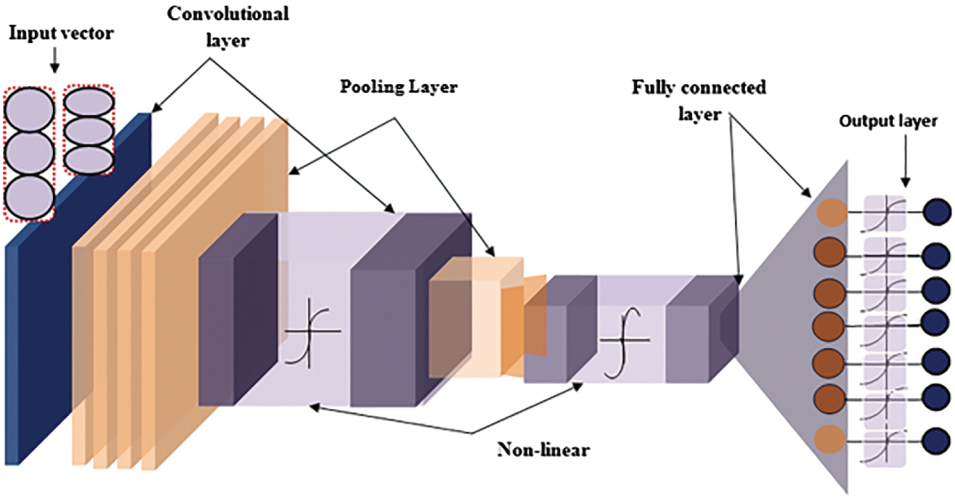

Figure 2: MBFOCNN encoder

(a) Convolutional Layer

CNNs are a deep learning method that takes an input picture and extracts information by convolving it using filters or kernels. The input image is filtered, as well as the convolution technique learns similar feature over whole image. Without pooling, Eq. (6) yields size of the resulting matrix:

$$[\Gamma_{i,j}] \times \lambda_{i,j} = \Gamma - \lambda + 1 \tag{6}$$

The window advances with every action, and indeed the feature maps discover the features. To capture the image's local receptive field, the feature maps employ common weights and biases. The convolution process is described by Eq. (7):

$$O_{CONV} = \delta_{sigmoid}\left(b + \sum_{i=0}^{2}\sum_{j=0}^{2} w_{i,j}\Psi_{a+i,b+j}\right) \tag{7}$$

Now, initialization of the weight is done using Modified Bacteria Foraging Optimization. BFA is an optimization approach inspired on E. coli bacteria's foraging behaviour. Discusses the biological features of bacterial hunting tactics and motile behaviour, and also their decision-making systems. BFA is meant to handle complicated and non-differentiable objective functions and handle non-gradient optimization issues. Chemotaxis, reproduction, and elimination dispersal activities are the three main processes used to search hyperspace. Swimming and tumbling are used in the chemotaxis mechanism. The bacteria spends its whole existence switching between these two motions.

The fundamental BFA's unit step length is fixed, ensuring good searching outcomes for modest optimization tasks. If applied to difficult situations with great dimensionality, nevertheless, it performs poorly. The run length option is crucial for managing the BFA's local and global search capabilities. Modifying the run-length unit may, in this case, be used to balance the exploration and exploitation of search.

The algorithm performs a certain mechanism to find out initial weight. The MBFO algorithm follows three important mechanisms that are chemotaxis, reproduction, and elimination-dispersal. Initialization of weight is evaluated based on the position change of the bacterium, and it is given by Eq. (8):

$$w^j(k+1,\ z,\ l) = w^j(k,\ z,\ l) + \Theta(j)\frac{\Delta(j)}{\sqrt{\Delta^T(j)\Delta(j)}} \tag{8}$$

Here, $\Delta(j)$ is k^{th} chemotactic step direction vector, $w^j(k, z, l)$ indicates bacterium at the k^{th}chemotactic, z^{th} reproductive, l^{th} elimination dispersal step. $\Theta(j)$ is chemotactic step size while every run or tumble which is formulated by using central angle formulae given by Eq. (9):

$$\Theta(j) = \frac{S_A}{r} \tag{9}$$

where S_A denotes the arc length and r is the radius length

For convolutional layer, the kernals/filters are initialized with MBFO optimizer

(b) Pooling Layer

Pooling is used in order to preserve input image size. Output image size for 'SAME' Pooling is the same as the input image size and there is no Pooling for 'True' Pooling. The size of the Pooling output matrix is illustrated as Eq. (10):

$$[\Gamma_{i,j}] \times \lambda_{i,j} = (\Gamma + 2p - \lambda)/(\gamma_s + 1) \tag{10}$$

Here, O_{CONV} is the output, p is the Pooling, γ_s is the stride, $\delta_{sigmoid}$ is sigmoid activation function, $w_{i,j}$ weight matrix of shared weights and $\Psi_{a+i,b+j}$ is input activation at location i, j.

After Pooling of output matrix, the convolution layer obtains a feature map for the text matrix as well as for image data. The obtained feature map is provided to fully connected layer.

(c) Fully Connected Layer

Result of previous segmentation mask layer is smoothed and sent into fully connected layer as an input. Flattened vector is used to train fully connected layer, that is similar to an ANN. Eq. (11) used in the training of vectors

$$\Gamma_I^T = act\left(\sum_{i=1}^{n} w_i \forall_{flattened} + \aleph_b\right) \tag{11}$$

where, $\aleph_b$ implies bias initialized randomly, w_i implies weights of respective input nodes, *act* implies activation function. To obtain vector values of input image, fully connected layer uses softmax activation function and reflects activation function. The vector image is then sent to the decoder, which handles the captioning. The of the suggested encoder approach, MBFOCNN, is outlined and depicted as pseudo-code in Fig. 3.

Input: Detected Object from IYV3MMD ($\zeta_{i,j}^{Od}$)
Output: Encoded vectors of image

Begin

Initialize the kernel ($\lambda_{i,j}$), bias (b), stride (γ_s), chemotactic (k^{th}), reproductive (z^{th}),
Elimination dispersal step (l^{th}),
Evaluate weight using,

$$w^j(k+1,z,l) = w^j(k,z,l) + \Theta(j)\frac{\Delta(j)}{\sqrt{\Delta^T(j)\Delta(j)}}$$

For ($CONV = 1 to N$)
Evaluate the convolution operation using,

$$O_{CONV} = \delta_{sigmoid}\left(b + \sum_{i=0}^{2}\sum_{j=0}^{2} w_{i,j}\Psi_{a+i,b+j}\right)$$

If ($\Gamma_{i,j} == valid\ padding$)
No padding is required
Else
Evaluate padding layer using,

$$\lfloor \Gamma_{i,j} \rfloor \times \lambda_{i,j} = (\Gamma + 2p - \lambda)/(\gamma_s + 1)$$

End If
Evaluate the fully connected layer using,

$$\Gamma_I^T = act\left(\sum_{i=1}^{n} w_i \forall_{flattened} + \aleph_b\right)$$

End for
End begin

Figure 3: Pseudo code for MBFOCNN

3.2 Caption Generation

Caption generation helps in training the model based on the text so as to obtain the respective captioning. Before giving a text as it is, from a dataset, may cause a high error rate for captioning an image. In order to overcome the flaws, the work was developed with a two major step i.e.,

- Preprocessing of textual content
- Word embedding

3.2.1 Preprocessing of Textual Content

Preprocessing of textual content helps to enhance text quality to minimize the error rate.

3.2.2 Word Embedding

Word embeddings convert single words into fully valued vectors within specified vector spaces where every word is mapped as a single vector, and vector values are learnt in a form that resemble neural networks or classifying approaches of deep learning. In order to provide the vector values for the pre-processed text, this work uses MGloVes word embedding technique. MGloVe provides word embeddings by combining both global statistics of matrix factorization approaches such as LSA with Word2Vec 's local context dependent learning. The MGloVe performs certain steps:

Step 1: Collect co-occurrence word statistics as the matrix of word co-occurrence matrix Ψ. Every component of the matrix Ψ_{ij} represents how much appears in a single word's meaning. It normally looks for background words in a certain area determined by size of the window before and after the word.' For more distant words, it normally assigns less weight using the following formula in Eq. (12):

$$\omega_{delay} = 1/offset \tag{12}$$

Step 2: A soft constraint has been defined for each word pair using Eq. (13):

$$\omega_i^T \omega_j + b_i + b_j = \log(\Psi_{ij}) \tag{13}$$

where, ω_i denotes key word, ω_j indicates vector for context word, b_i and b_j are scalar biases for main and context words. Ψ_{ij} denotes the input embedded vector.

Step 3: cost function evaluation are done using Eq. (14):

$$L = \sum_{i=1}^{N} \sum_{j=1}^{N} f(\varsigma_{ij})(\omega_i^T \omega_j + b_i + b_j - \log(\varsigma_{ij}))^2 \tag{14}$$

Here, N is vocabulary size, f is weighting feature that aids us to prevent learning from very common word pairs. MGloVe select the following feature is based on Eq. (15):

$$f(\varsigma_{ij}) = \begin{cases} \left(\frac{\varsigma_{ij}}{\varsigma_{\max}}\right)\ell, & if\ \varsigma_{ij} < \varsigma_{\max} \\ 1, & otherwise \end{cases} \tag{15}$$

Step 4: Word embedding produced must be compared to obtain semantically resemblance among two vectors. The similarity between the two vectors is found out using Eq. (16)

$$OLSCoefNoIntept(\varsigma_{ij}, \varsigma_{\max}) = \frac{\langle \varsigma_{ij}, \varsigma_{\max} \rangle}{\|\varsigma_{ij}\|^2} \tag{16}$$

3.3 Decoder

The decoder is done in order to provide a caption for an image. In the decoding process, the text gets trained up by following the textual understanding and thereafter gets the captioning of the image. Several decoding models have been developed up-to-date but accurately captioning an image remains to be a challenge and computationally complex. Existing methods/models are effective in their IC processes, but unable to determine using visual signals or language models correspondingly. In order to overcome the flaws, the work has proposed MDAAs for visual sentinels as shown in Fig. 4.

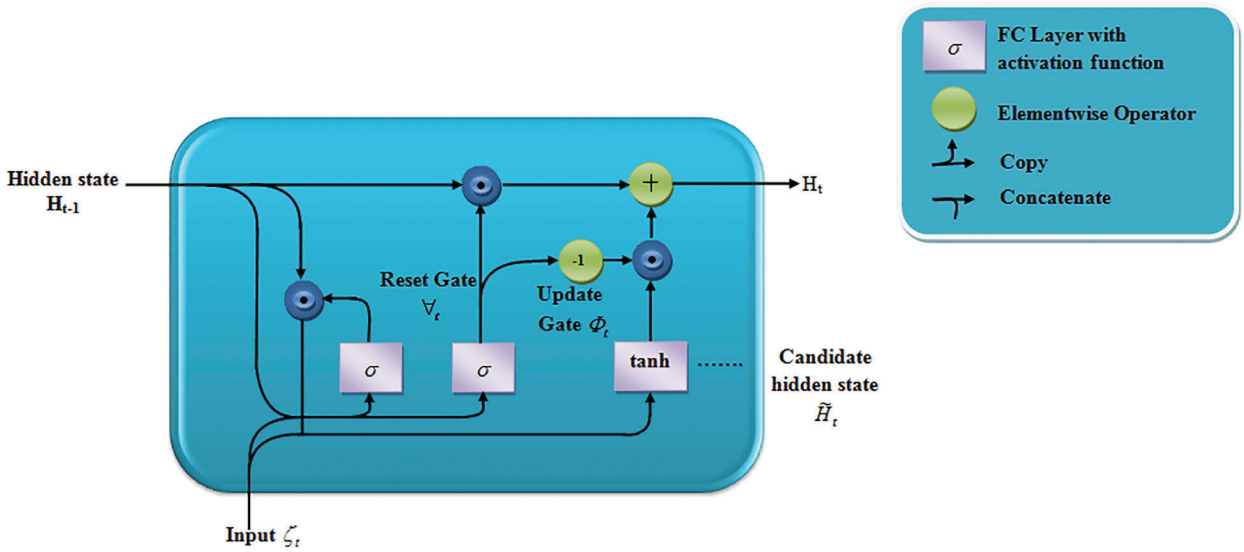

Figure 4: MDAA decoder

MDAA provides a decoding process by accepting the image vector as well as embedding text vectors to caption an image. The MDAA uses a Gated Recurrent Unit (GRU) as a decoder, which avoids the gradient vanishing or exploding problems and remains to be less complex as compared to the LSTM.

3.3.1 Reset and Update Gate

current time step is first supplied as input to both the reset and update gates in GRU, whereas prior time step is given to hidden state. Two fully-connected layers with sigmoid activation function provide outputs of two gates.

If the input is a minibatch $\Im_t \in \Re^{n\times d}$ (count of instances: n, count of inputs: d) and the previous time step's t hidden state is $H_{t-1} \in \Re^{n\times h}$ (total hidden units: h), then reset and update gates $\forall_t \in \Re^{n\times h}$ and $\Phi_t \in \Re^{n\times h}$ are calculated as follows:

$$\forall_t = \sigma(\Im_t W_{X\forall} + H_{t-1} W_{h\forall} + b_\forall) \tag{17}$$

$$\Phi_t = \sigma(\Im_t W_{X\Phi} + H_{t-1} W_{h\Phi} + b_\Phi) \tag{18}$$

where, $W_{X\forall} W_{X\Phi} \in \Re^{d\times h}$ and $W_{h\forall} W_{h\Phi} \in \Re^{h\times h}$ is weight parameters $b_\forall b_\Phi \in \Re^{1\times h}$ is bias, σ is the sigmoid function to transform the input interval from (0, 1).

3.3.2 Candidate Hidden State

In candidate hidden state, reset gate is integrated with regular latent state updating mechanism, which is given by

$$\tilde{\mathrm{H}}_t = \tanh(\Im_t W_{\zeta h} + (\forall_t \otimes \mathrm{H}_{t-1}) W_{hh} + b_h \tag{19}$$

where, $\tilde{\mathrm{H}}_t \in \Re^{n \times h}$ is the candidate hidden state, $W_{\zeta h} \in \Re^{d \times h}$, $W_{hh} \in \Re^{h \times h}$ are weight factors $b_h \in \Re^{1 \times h}$ is bias and $\otimes$ is element-wise product operation. The candidate hidden state uses tanh to guarantee that all values remain between interval [−1, 1].

3.3.3 Hidden State

Lastly, update gate Φ_t has to be included as it tells on new hidden states $\mathrm{H}_t \in \Re^{n \times h}$ and candidate states $\tilde{\mathrm{H}}_t$ which is used by old states H_{t-1}. Hence, Φ_t is used for element wise convex combinations of H_{t-1} and $\tilde{\mathrm{H}}_t$. T resulting in final GRU update Eq. (20):

$$\mathrm{H}_t = \Phi_t \otimes \mathrm{H}_{t-1} + (1 - \Phi_t) \otimes \tilde{\mathrm{H}}_t \tag{20}$$

Old state is simply retained when the update gate Φ_t is near to 1, In this case, data from ζ_t in the dependency chain is basically ignored and effectively skip the time step*t*. In comparison, the current latent state H_t reaches the latent candidate state $\tilde{\mathrm{H}}_t$ once Φ_t is close to 0. These designs address RNN's vanishing gradient issue while resulting in better capturing dependencies for long time-step distance sequences. If the update gate has been close to 1 for all of a subsequence's time steps, for instance, the old hidden state will be disclosed. Will be easily kept and transferred to its end at the time stage of its beginning, regardless of the subsequence's duration. Finally, the model decodes the image and provides the captioning. Thus, the overall outline for suggested decoder approach that is MDAAs is illustrated in pseudo-code form as stated in Fig. 5.

Input: Vector Image and Text
Output: Image Captioning

Begin
Initialize the weight parameters of input layer ($W_{X\forall}$, $W_{X\Phi}$) and hidden layer ($W_{h\forall}$, $W_{h\Phi}$), bias ($b_\forall$, b_Φ), previous hidden layer (H_{t-1}).
While ($t < L(text)$)
Evaluate reset gate ($\forall_t$) using,
$\forall_t = \sigma(\zeta_t W_{X\forall} + \mathrm{H}_{t-1} W_{h\forall} + b_\forall)$
If ($\forall_t == \forall_{t-1}$)
Reset the gate
Else
Update the gate value using,
$\Phi_t = \sigma(\zeta_t W_{X\Phi} + \mathrm{H}_{t-1} W_{h\Phi} + b_\Phi)$
End if
If ($\Phi_t \geq 1$)
Evaluate the hidden candidate activation vector using,
$\tilde{\mathrm{H}}_t = \tanh(\zeta_t W_{\zeta h} + (\forall_t \otimes \mathrm{H}_{t-1}) W_{hh} + b_h$
Else
Evaluate the hidden state that is output vector
$\mathrm{H}_t = \Phi_t \otimes \mathrm{H}_{t-1} + (1 - \Phi_t) \otimes \tilde{\mathrm{H}}_t$
End if
End while
End begin

Figure 5: Pseudo code for MDAA

4 Results and Discussion

This work's suggested framework's performance in ICs is evaluated on publicly available datasets with current methodologies and using various performance indicators to determine its efficacy. The framework is implemented in PYTHON. The performance of the proposed MBFOCNN encoder and MDAA decoder is analyzed with the existing methods: BRNNs (Bidirectional Recurrent Neural Networks) [33]; CNN-LSTM [34]; Fast RCNNs [35] and CNN-AAs (CNNs with Adaptive Attentions) [36]. The analysis has been made based on the metrics: Accuracy, Specificity, Sensitivity, F-Measure, precision; NPVs (Negative predictive values); MCCs (Matthew's Correlation Coefficients);FPRs (False Predictive Rates), and FNRs (False Negative Rates). Based on the metrics, such as Accuracy, Specificity, Sensitivity, and Precision, the evaluation has been done for the various techniques. The evaluation of the techniques has been given in Tab. 1.

Table 1: Inference of existing methods for breast cancer detection

S. no	Author details	Methodology	Merits	Demerits
1.	Kinghorn et al. (2017)	Region proposal network from faster R-CNN	Selective search, for example, is less reliable.	However it has issue with time complexity
2.	Su et al. (2020)	Hierarchical deep neural network for IC	Solves the computational problems from different fields	Fast prediction is not possible
3	Anderson et al. (2018)	Faster convolutional neural net (CNN) R-CNN	High detection rate	Time consumption is high
4	Zhu et al. (2018)	CNN classifier	High sensitivity and accuracy	Requires high number of mammogram images for training process
5	Xu et al. (2015)	Deep learning algorithm	It achieves high level of classification accuracy Overall efficiency is greater	This technique has issue with longer training time

Tab. 2 shows how the proposed MBFOCNN-MDAA was evaluated using performance criteria and current methods specified above. Fast RCNN-RNN, and CNN-AA. For obtaining an efficient IC, the encoder and decoder should work together better in order to minimize the error and to obtain an accurate captioning of an image. The proposed model is also evaluated with the metrics by achieving an Accuracy value of 96.24%, Specificity value of 96.49%, Sensitivity value of 97.63%, and Precision value of 97.63%, which ranges between 96.24%–97.63%. But the existing BRNN, CNN-LSTM, Fast RCNN-RNN, and CNN-AA methods often achieve a value for the metrics ranging from 81.39 percent to 92.52 percent, which is lower than the suggested technique. By limiting the incidence of mistakes, the suggested MBFOCNN-MDAA technique provides superior encoding and decoding for creating an image-based caption, and it is proven to be efficient when compared to existing methods. Fig. 6 depicts a graphical depiction of the suggested technique alongside several current methods.

Table 2: Analysis of suggested MBFOCNN-MDAA depending on accuracy, specificity, sensitivity, and precision

Performance metrics/techniques	BRNN	CNN-LSTM	Fast RCNN–RNN	CNN-AA	Proposed MBFOCNN-MDAA
Accuracy	90.86	88.19	81.39	83.81	96.24
Specificity	91.48	88.92	81.82	84.14	96.49
Sensitivity	92.22	91.71	88.71	84.88	97.63
Precision	92.52	91.71	88.23	84.86	97.63

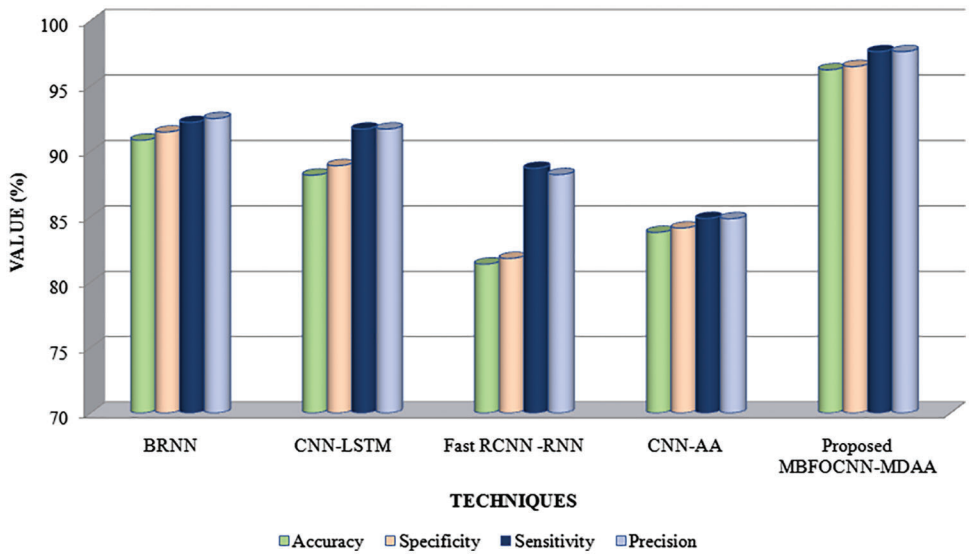

Figure 6: Graphical analysis of proposed MBFOCNN-MDAA based on accuracy, specificity, sensitivity, and precision

Figs. 7a and 7b gives a graphical representation of the proposed MBFOCNN-MDAA method evaluated with the existing BRNN, CNN-LSTM, Fast RCNN-RNN, and CNN-AA methods. Graphically, it can be stated that proposed methods tend to achieve a high metrics value as compared to existing methods. Thereafter, the BRNN techniques perform better after the proposed technique, and the remaining CNN-LSTM, Fast RCNN-RNN, and CNN-AA differ by a wide range while compared with the suggested approach. MBFOCNN-MDAA is analyzed depending on metrics, such as F-Measure, NPV, MCC, FPR, and FNR, and evaluated with the existing methods, such as BRNN, CNN-LSTM, Fast RCNN-RNN, and CNN-AA. The evaluation of the techniques has been tabulated in Tab. 3.

Table 3: Analysis of the proposed MBFOCNN-MDAA based on F-Measure, NPV, MCC, FPR, and KNN

Performance metrics/techniques	BRNN	CNN-LSTM	Fast RCNN–RNN	CNN-AA	Proposed MBFOCNN-MDAA
F-measure	92.22	91.71	88.71	84.88	97.63
NPV	96.48	95.92	95.82	96.24	97.49
MCC	92.22	91.71	89.71	84.58	97.63
FPR	14.59	35.92	34.24	21.59	2.65
FNR	21.11	51.41	35.43	25.14	13.04

Based on the metrics, such as F-Measure, NPV, MCC, FPR, and FNR, the evaluation of the proposed MBFOCNN-MDAA has been done with various existing techniques, such as BRNN, CNN-LSTM, Fast RCNN-RNN, and CNN-AA as shown in Tab. 3. In order to depict a better model, not every measure acquired must be of greater value; nevertheless, achieving lower values for specific measures, like FPR and FNR, can help to achieve an effective system. As per this, suggested system attains an FPR of 2.65 percent and a FNR of 13.04 percent, while current approaches try to obtain FPR and FNR values between 14.59 percent to 51.41 percent, indicating a high level of false detection of the text to the respective image when contrasted with suggested technique. Suggested approach, conversely, achieves high F-Measure, NPV, and MCC value of 97.63 percent, 97.49 percent, and 97.63 percent, respectively, whereas the existing method achieves F-Measure, NPV, and MCC values between 84.88 percent to 96.48 percent, indicating lower model efficiency when compared to the proposed MBFOCNN-MDAA method. The suggested MBFOCNN-MDAA technique improves encoding quality by eliminating erroneous detection, and decoding for creating an image-based caption, it was found to be more efficient than previous techniques which is due to faster prediction and good convergence. Fig. 7 depicts a graphical depiction of the suggested technique alongside several current methods.

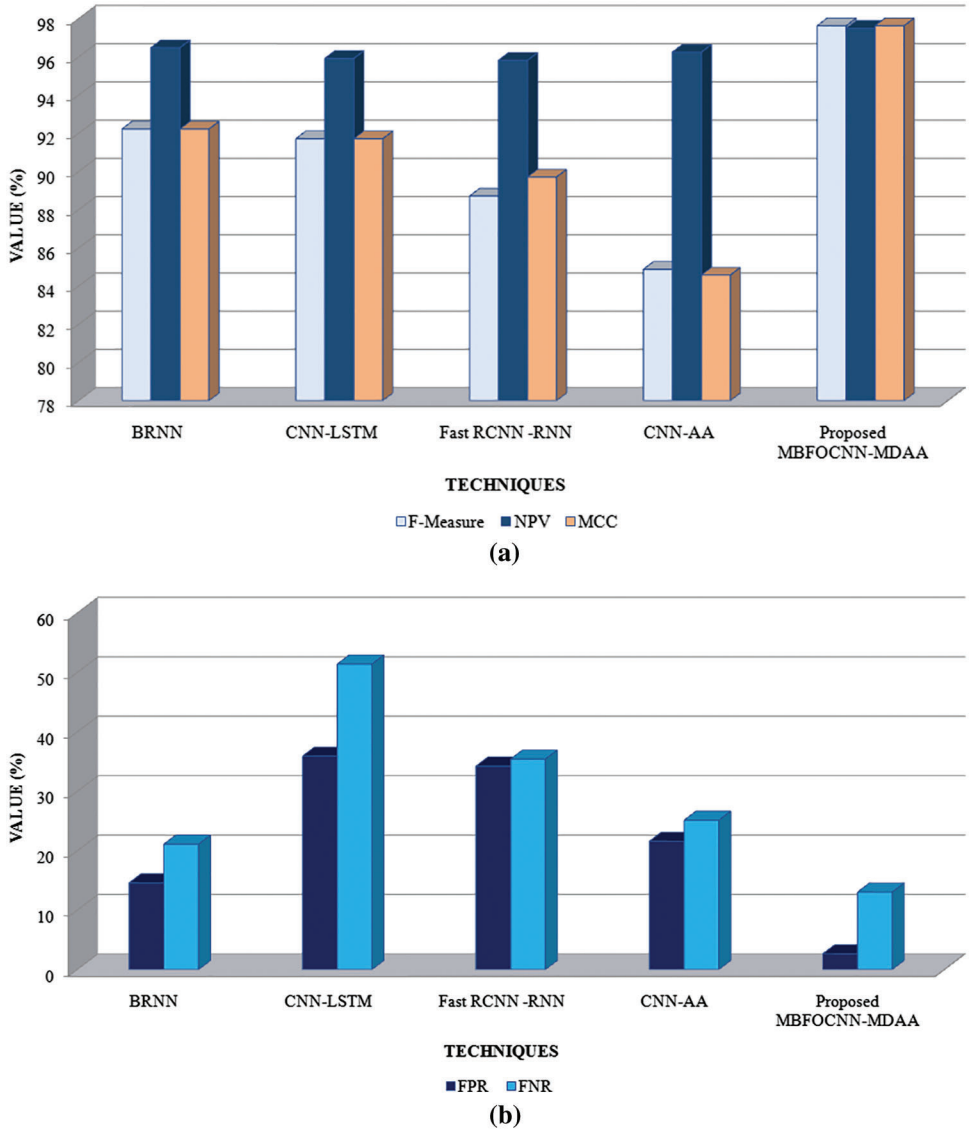

Figure 7: Graphical analysis of suggested MBFOCNN-MDAA formed on (a) F–Measure, NPV, MCC (b) FPR, FNR

Fig. 7 displays the suggested MBFOCNN-MDAA method's graphical analysis depending on performance metrics. Fig. 7a indicates graphical analysis of techniques depending on F–Measure, NPV, MCC, which states that the proposed technique achieves a better metrics value as compared to the existing techniques. The CNN-LSTM technique tends to give better results after the proposed method and the remaining existing techniques vary within a huge difference. Fig. 7b states the FPR and FNR metrics analysis for the proposed technique with the existing methodologies. The proposed method tends to achieve a lower value of FPR and FNR as compared to existing methods, and it is considered to be efficient due to avoiding the false prediction for IC. Thus, the proposed MBFOCNN-MDAA method outperforms the existing methods and remains to perform accurate IC.

5 Conclusion

The proposed work has developed a framework for ICs consisting of two important components that are image encoding using MBFOCNN and decoding of the image using MDAA. The two major components provide an accurate IC and also extract semantic correlations of acquired visual information with corresponding language expressions. In addition to that, the proposed work provides accurate IC for complex scenes and concentrates on every feature by keeping in mind the quality of the image. Contrasted with modern method, suggested system attends to interrelationships between objects in an image and provides Automatic caption or description generation from images. The experimental outcome showed that suggested approach achieved 96.24% accuracy, 96.49% Specificity, 97.63% Precision, and obtains a minimized false detection by achieving an FPR and FNR value of 2.65% and 13.04% respectively. In the future, the work will be directed in improving the encoder and decoder mechanism for captioning an image.

Acknowledgement: We thank anonymous referees for their helpful suggestions.

Funding Statement: The authors received no specific funding for this study.

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

References

[1] H. Liu, G. Wang, T. Huang, P. He, M. Skitmore *et al.,* "Manifesting construction activity scenes via image captioning," *Automation in Construction*, vol. 119, no. 6, pp. 01–19, 2020.

[2] X. Li and S. Jiang, "Know more say less: Image captioning based on scene graphs," *IEEE Transactions on Multimedia*, vol. 21, no. 8, pp. 2117–2130, 2019.

[3] Q. Wu, C. Shen, P. Wang, A. Dick and A. V. D. Hengel, "IC and visual question answering based on attributes and external knowledge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1367–1381, 2017.

[4] S. He and Y. Lu, "A modularized architecture of multi-branch convolutional neural network for image captioning," *Electronics*, vol. 8, no. 12, pp. 1–15, 2019.

[5] G. Hoxha, F. Melgani and B. Demir, "Toward remote sensing image retrieval under a deep IC perspective," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, no. 3, pp. 4462–4475, 2020.

[6] S. Ye, J. Han and N. Liu, "Attentive linear transformation for IC," *IEEE Transactions on Image Processing*, vol. 27, no. 11, pp. 5514–5524, 2018.

[7] J. Yu, J. Li, Z. Yu and Q. Huang, "Multimodal transformer with multi-view visual representation for image captioning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 12, pp. 4467–4480, 2019.

[8] J. Wang, W. Wang, L. Wang, Z. Wang, D. D. Feng *et al.,* "Learning visual relationship and context-aware attention for IC," *Pattern Recognition*, vol. 98, no. 10, pp. 01–11, 2020.

[9] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, H. Laga and Bennamoun M., "Bi-SAN-CAP:Bi-directional self-attention for image captioning," in *IEEE 2019 Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–7, 2019.

[10] S. Cao, G. An, Z. Zheng and Q. Ruan, "Interactions guided generative adversarial network for unsupervised IC," *Neurocomputing*, vol. 417, no. 12, pp. 419–431, 2020.

[11] P. Xia, J. He and J. Yin, "Boosting image caption generation with feature fusion module," *Multimedia Tools and Applications*, vol. 79, no. 33, pp. 24225–24239, 2020.

[12] C. Wang, H. Yang and C. Meinel, "IC with deep bidirectional LSTMs and multi-task learning," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 2s, pp. 1–20, 2018.

[13] J. H. Tan, C. S. Chan and J. H. Chuah, "COMIC: Toward a compact IC model with attention," *IEEE Transactions on Multimedia*, vol. 21, no. 10, pp. 2686–2696, 2019.

[14] M. Yang, W. Zhao, W. Xu, Y. Feng, Z. Zhao *et al.,* "Multitask learning for cross-domain IC," *IEEE Transactions on Multimedia*, vol. 21, no. 4, pp. 1047–1061, 2018.

[15] Z. Zhang, Q. Wu, Y. Wang and F. Chen, "High-quality IC with fine-grained and semantic-guided visual attention," *IEEE Transactions on Multimedia*, vol. 21, no. 7, pp. 1681–1693, 2018.

[16] X. Xiao, L. Wang, K. Ding, S. Xiang and C. Pan, "Dense semantic embedding network for IC," *Pattern Recognition*, vol. 90, pp. 285–296, 2019.

[17] H. Chen, G. Ding, Z. Lin, Y. Guo, C. Shan *et al.,* "Image captioning with memorized knowledge," *Cognitive Computation*, vol. 13, no. 4, pp. 807–820, 2021.

[18] X. Zhang, S. He, X. Song, R. W. H. Lau, J. Jiao *et al.,* "IC via semantic element embedding," *Neurocomputing*, vol. 395, no. 6, pp. 212–221, 2020.

[19] F. Xiao, X. Gong, Y. Zhang, Y. Shen, J. Li *et al.,* "DAA: Dual LSTMs with adaptive attention for IC," *Neurocomputing*, vol. 364, no. 10, pp. 322–329, 2019.

[20] X. Xiao, L. Wang, K. Ding, S. Xiang and C. Pan, "Deep hierarchical encoder–decoder network for IC," *IEEE Transactions on Multimedia*, vol. 21, no. 11, pp. 2942–2956, 2019.

[21] X. Shen, B. Liu, Y. Zhou, J. Zhao and M. Liu, "Remote sensing IC via variational autoencoder and reinforcement learning," *Knowledge-Based Systems*, vol. 203, no. 4, pp. 01–11, 2020.

[22] P. Kinghorn, L. Zhang and L. Shao, "A hierarchical and regional deep learning architecture for image description generation," *Pattern Recognition Letters*, vol. 119, no. 9, pp. 77–85, 2017.

[23] Y. Su, Y. Li, N. Xu and A. Liu, "Hierarchical deep neural network for IC," *Neural Processing Letters*, vol. 52, no. 2, pp. 1057–1067, 2020.

[24] D. Zhao, Z. Chang and S. Guo, "A multimodal fusion approach for IC," *Neurocomputing*, vol. 329, pp. 476–485, 2019.

[25] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville *et al.,* "Show, attend and tell: Neural image caption generation with visual attention," in *Int. Conf. on Machine Learning*, Lille, France, pp. 2048–2057, 2015.

[26] P. Anderson, X. He, C. Buehler, D. Teney M. Johnson *et al.,* "Bottom-up and top-down attention for image captioning and visual question answering," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 6077–6086, 2018.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit L. Jones *et al.,* "Attention is all you need. MIT Press," in *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

[28] X. Zhu, L. Li, J. Liu, H. Peng and X. Niu, "Captioning transformer with stacked attention modules," *Applied Sciences*, vol. 8, no. 5, pp. 01–11, 2018.

[29] S. Herdade, A. Kappeler, K. Boakye and J. Soares, "IC: Transforming objects into words," *MIT press in Advances in Neural Information Processing Systems*, Cambridge, MA, USA, pp. 11135–11145, 2019.

[30] L. Huang, W. Wang, J. Chen and X. Wei, "Attention on attention for IC," in *Proc. of the IEEE Int. Conf. on Computer Vision*, Seoul, Korea, pp. 4634–4643, 2019.

[31] Y. Pan, T. Yao, Y. Li and M. Tao, "X-linear attention networks for IC," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Nashville, TN, USA, pp. 10971–10980, 2020.

[32] M. Cornia, M. Stefanini, L. Baraldi and R. Cucchiara, "Meshed-memory transformer for IC," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Nashville, TN, USA, pp. 10578–10587, 2020.

[33] Y. Fan, Y. Qian, F. L. Xie and F. K. Soong, "TTS synthesis with bidirectional LSTM based recurrent neural networks," in *Fifteenth Annual Conf. of the Int. Speech Communication Association*, Singapore, pp. 1–5, 2014.

[34] M. Soh, "*Learning CNN-LSTM Architectures for Image Caption Generation*," Department of Computer Science Stanford University, CA, USA, pp. 1–9, 2016.

[35] Z. C. Fei, "Fast image caption generation with position alignment," *Computer Vision and Pattern Recognition*, pp. 1–8, 2019.

[36] A. Hani, N. Tagougui and M. Kherallah, "Image caption generation using a deep architecture," in *Int. Arab Conf. on Information Technology (ACIT)*, Al Ain, United Arab Emirates, pp. 246–251, 2019.