Journal of Intelligent & Fuzzy Systems xx (20xx) x–xx
DOI:10.3233/JIFS-189415
IOS Press

Deep image captioning using an ensemble of CNN and LSTM based deep neural networks

Jafar A. Alzubi[a], Rachna Jain[b], Preeti Nagrath[b], Suresh Satapathy[c,*], Soham Taneja[b] and Paras Gupta[b]
[a]*Al-Balqa Applied University, Salt, Jordan*
[b]*Bharati Vidyapeeth's College of Engineering, New Delhi, India*
[c]*KIIT Deemed to be University, Bhubaneswar, India*

Abstract. The paper is concerned with the problem of Image Caption Generation. The purpose of this paper is to create a deep learning model to generate captions for a given image by decoding the information available in the image. For this purpose, a custom ensemble model was used, which consisted of an Inception model and a 2-layer LSTM model, which were then concatenated and dense layers were added. The CNN part encodes the images and the LSTM part derives insights from the given captions. For comparative study, GRU and Bi-directional LSTM based models are also used for the caption generation to analyze and compare the results. For the training of images, the dataset used is the flickr8k dataset and for word embedding, dataset used is GloVe Embeddings to generate word vectors for each word in the sequence. After vectorization, Images are then fed into the trained model and inferred to create new auto-generated captions. Evaluation of the results was done using Bleu Scores. The Bleu-4 score obtained in the paper is 55.8%, and using LSTM, GRU, and Bi-directional LSTM respectively.

Keywords: Deep learning, LSTM, neural network, glove embedding, image captioning, bleu score

1. Introduction

With the increase of internet technology and comprehensive access to digital cameras, we're surrounded by a large number of images, followed by a lot of relevant text. However, the relationship between the surrounding text and images varies greatly, and how to close the gap between perception and language is a difficult issue for the scalable image annotation mission. It is difficult for humans at times to describe a picture in a text after one glance. It is the same with the machines. However, due to the recent progress in computer vision as well as natural language processing, it is now possible to do so.

Image captioning is a deep learning task where the objective is to produce a caption for any given image. It uses an ensemble of CNN-based computer vision and LSTM-based natural language processing methods. The generation of image captioning has emerged in recent times as a complicated research field with developments in modeling and recognition of images in statistical languages. The creation of captions from images has multiple functional advantages, ranging from assisting the visually impaired to automating image captioning on millions of photographs uploaded to the Internet. The field integrates the two state-of-the-art models within Natural Language Processing and Computer Vision, two of Artificial Intelligence's main areas. Evaluation of the trained model is carried out using the BLEU score.

In this paper, a deep learning-based model was used, which comprises an ensemble of LSTM and

*Corresponding author. Suresh Satapathy, KIIT Deemed to be University, Bhubaneswar, India. E-mail: sureshsatapathy@ieee.org.

CNN networks. Apart from LSTM, GRU as well as Bi-directional LSTM is also used to compare and analyze the results as well as the overall performance of the model.

The objective of this paper is to build a model using the concepts of CNN and LSTM and then train the model to accurately predict the captions of the given images [1]. The model was successfully built with a BLEU-4 score of 0.558. With the GRU and Bi-directional LSTM model, the BLEU-4 score of and respectively were obtained. Due to resource constraints, the Flickr 30 k dataset was avoided as its size is large. The accuracy of the model was constrained by the size of the dataset [2] and the GPU memory available to us. The results obtained were then compared with results in the researches of Quangzeng et al. [3] and Mulachery et al. [4]. Batch size of 4096–6500 was considered.

This paper is organized as Section II contains a brief description of the researches done in the field of image captioning over the recent years, Section III describes the datasets used to train the model with their sources. Section IV gives a detailed explanation of the data preprocessing, model architecture, and the working as well as the model deployment. Section V includes the results obtained and their comparison as well as analysis. Section VI comprises the conclusion drawn from the results obtained in the previous section and proposed modifications.

2. Related work

Recent improvements in Deep Learning have significantly increased the performances of various Models. In the past, different groups have conducted various studies regarding similar fields or themes.

The researches of D. Narayanswamy [5] generates the captions based on various frames in a video, F. Keller and D. Elliott [6], where they focused on visual dependency representations to identify various elements of the image. Researches related to Natural Language Processing and computer vision, Researches of UT, Austin, and Stanford (A. Karpathy, Li Fei Fei) [7] also focus on Image Caption Generation. Wang [8] and Q. Wu, C. Shen, P. Wang [9] used 2 LSTM units where a skeleton sentence is generated, and extracted attributes are added to this sentence. K. Fu, J. Jin, R. Cui, F. Sha, and C. Zhang [10] use sub-blocks or segments to generate captions for features extracted from the various segments of images. R. Vedantam, K. Murphy, S. Bengio et al. [11] have used discriminator class for generating context-aware captions. Z. Gan [12] rather than using captions used semantic concepts that were present in the image. Chuang [13] used three LSTM units with the same parameters but different styles (such as facts, humour, etc.) to generate captions. L. Yang, K. Tang [14] fed the features of different regions of the image into the LSTM for caption generation. Similar to [15], J. Krause, J. Johnson, R. Krishna [16] also used the regions of images but for generated sentences using hierarchical RNN. O. Vinyals [17] and F. Liu, T. Xiang, T. M. Hospedales [18] used different approaches in multiple training datasets for various layers in the network for caption generation. Google (conceptual captions) [19] and Facebook (image tagging) [20] have also researched and practically implemented image captioning.

As groundbreaking work, Kiro et al. [21] used neural networks to capture images using a multimodal neural language model. He developed an encoder-decoder pipeline in their follow-up work [22], where the sentence was encrypted by LSTM and decrypted with the neural language model structure-content (SCNLM). To retrieve images, Socher et al. [23] proposed a DT-RNN (Dependency Tree-Recursive Neural Network) embed sentence into a vector space.

Multi-instance learning and the conventional maximum-entropy language model were used by Fang et al. [24] to produce definitions. Chen et al. [25] suggested studying visual representation for image caption creation with RNN. In [26], Xu et al. incorporated the human visual system's process of focus into the encoder-decoder paradigm. It is seen that the focus model can imagine what the model "sees" and creates major shifts in the generation of image captions. Mao et al. [27] suggested m-RNN to replace the feed-forward paradigm of neural language [22]. NIC [17] and LRCN [28] developed similar architectures, both methods use LSTM to learn text meaning. But at the first level, NIC only feeds visual information while the model of Mao et al. [27] and LRCN [28] considers image meaning at each phase of the time.

3. Dataset collection

3.1. Flickr 8 k dataset

The Flickr 8 k dataset consists of 8092 captioned images, with five different captions for each image [2]. 6000 images are available for training, while 1000 images are available for testing and validation. The

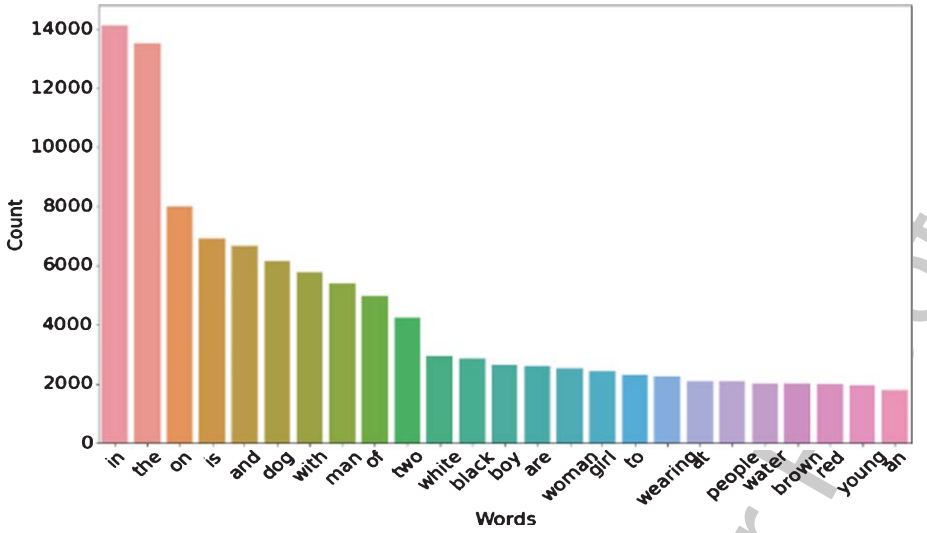

Fig. 1. Most occurring words in the given text corpus.

images are divided into testing and validation to prevent the overfitting of the model which is the main problem in the case of image captioning. Apart from this, the Flickr 30 k dataset is also available but due to the resource constraints, the shorter dataset is used.

Sample Captions for Fig. 1 in the dataset:

- Two youths are jumping over a roadside railing, at night.
- Two men in Germany jumping over a rail at the same time without shirts.
- Boys dancing on poles in the middle of the night.
- Two men with no shirts jumping over a rail.
- Two guys jumping over a gate together.

3.2. Glove embeddings

The glove embeddings contain the vector representation of words. It is an unsupervised approach to create a vector space of a given word corpus. This is accomplished by projecting terms into a coherent space where the difference between terms is related to semantic similarity [29]. Teaching is conducted on compiled global word-word co-occurrence statistics from a corpus, and interesting geometric substructures of the word vector space are exposed by the resulting representations. It was established as an open-source project at Stanford [1]. As a log-bilinear regression model for unsupervised learning of word representations [30], it integrates the features of two method families, such as global matrix factorization and local context window approaches.

Table 1
Cleaned captions

Original Captions	Captions after Data cleaning
A woman in a green sweater is sitting on the bench in a garden.	woman in a green sweater is sitting on a bench in the garden.
Axguyxstitchingxup anotherxmans coat.	guy stitching up another man coat.
A woman with a large purse is walking by the gate.	woman with a large purse is walking by the gate.

For this paper, the 6 B 100 d version was used, which consists of 6 billion words with 100 vectors for each word. Glove embeddings were loaded into an embedding matrix. The input to this layer would be a sentence of length 35, equal to the maximum length considered above.

4. Methodology

4.1. Preprocessing data

To feed data as input to the neural network, every image in the dataset needs to be converted into fixed-size vectors. The given images were scaled down to 299×299, which is the input shape expected by the Inception Network. The preprocessing function was applied to the given images. The given captions were read and the genism library was used to preprocess the captions. All captions were converted to lowercase and from them, the genism library removed numbers,

Fig. 2. Sample Image from flickr8k.

symbols, and punctuations. All words of length 1 were removed. The given captions were padded with zeros, considering a maximum length of 35. Start and End indicators were added to all the captions. Only those words were considered in the vocabulary which appears at least 10 times in the data. This was done to reduce model underfitting and reducing data size for training.

4.2. Image encoding

Inception V3 CNN-model was initialized using the imagenet weights without the final dense (inferencing) layer. Thus, the output of the model would be a tensor of shape (1,2048) after applying global average pooling. Feeding a 299 × 299 RGB image to this model encodes the image.

$$\text{CNN: } Z = X * f \tag{1}$$

Equation 1 describes how the CNN model works where, X = input, f = filter, *=convolution operation and Z is the extracted feature. A filter is an array of values with specific values for feature detection. The filter moves to every part of the image and returns a high value if the feature is detected. If the feature is not present then a low value is returned. In more elaboration, the filter can be described as a kernel which is basically a matrix with specific values [31]. The image is also a matrix with different values corresponding to the intensity of the pixel at that point ranging from 0 – 255 [32, 33]. The kernel moves from left to right and top to bottom and multiplies with each term in the image matrix and the summation of all the terms is stored in the output matrix. This helping in extracting features from the image. For the model used in this research, A 3 × 3 size kernel was adopted experimentally, as it offered greater accuracy in the time required to traverse the whole scan.

4.3. Proposed model

LSTM models are used in the domain of NLP. These models have a memory cell associated with them, which enables them to retain the information from previous training examples. Models like LSTM, GRU, and Bi-directional LSTM consist of multiple gates in the form of mathematical equations, which enable them to retain sequential data. In this paper, for the caption generation part, all the three models – LSTM, GRU, and Bi-directional LSTM were implemented. The GLoVe embeddings were loaded into an embedding matrix and an LSTM layer was initialized with GLoVe embeddings. The input to this layer would be a sentence of length 35, equal to the maximum length considered above.

$$C'(t) = \tanh\ (Wc\ [a(t-1),\ x(t)] + bc) \tag{2}$$

$$U(t) = \text{sigmoid}\ (Wu\ [a(t-1), x(t)] + bu) \tag{3}$$

$$F(t) = \text{sigmoid}\ (Wf\ [a(t-1), x(t)] + bf) \tag{4}$$

$$O(t) = \text{sigmoid}\ (Wo\ [a(t-1), x(t)] + bo) \tag{5}$$

$$C(t) = U(t)*C'(t) + F(t)*C(t-1) \tag{6}$$

$$a(t) = O(t) * \tanh(C(t)) \tag{7}$$

Equation 2, 3, 4, 5, 6, and 7 represent the functions for Candidate Cell, Update Gate, Forget Gate, Output Gate, Cell State, and Output with Activation respectively.

4.4. Model architecture

The concatenate layer provided in Keras was used to combine the Image layer as well as the LSTM layer. A dense layer was added as a hidden layer with activation function a dense layer of size equal to our vocabulary size was used to act as an inference layer, using a softmax activation as shown in Fig. 2.

4.5. Working of the proposed model

Figure 3 summarizes the complete working of the model. It can be divided into three phases:

4.5.1. Image feature extraction

The features present in the images from the Flickr-8K dataset are encoded using the Inception-V3 CNN model as it possesses optimal weights for the task of image classification. The Inception-V3

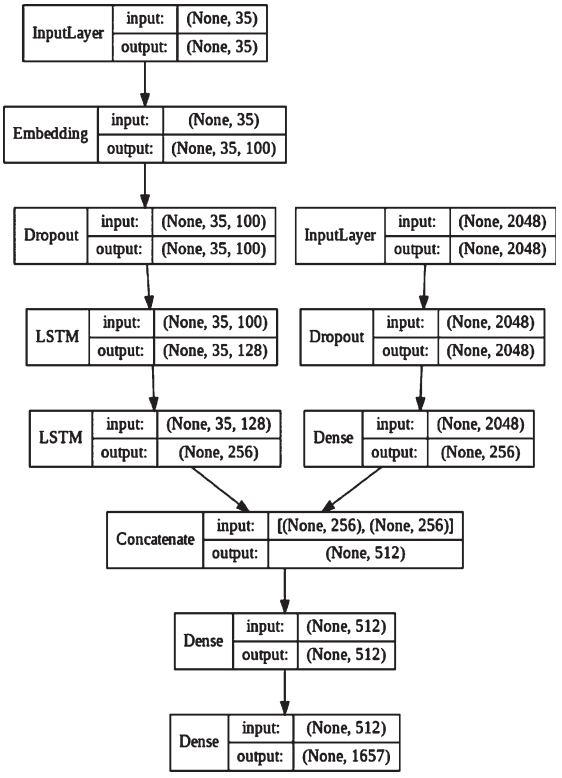

Fig. 3. Model Architecture for the CNN-LSTM model.

Table 2
The input sequence of captions and the next word in the sequence for the training phase

Input Sequence	Next word
<start>	Two
<start>, two	kids
<start>, two, kids	are
<start>, two, kids, are	playing
<start>, two, kids, are, playing	on
<start>, two, kids, are, playing, on	seesaw
<start>, two, kids, are, playing, on, seesaw	<end>

is deep learning-based the convolutional neural network which has multiple inception layers along with dropouts and the final fully connected inference layers. To counter the issue of overfitting, the dropout layers were added, which ignore some neurons while training. Doing so reduces the over-fitting. These are then passed to the dense layers, which generates a 2048 vector element representation of the image, which are then passed on to the LSTM layer.

4.5.2. Text extraction

In this step, the cleaned captions are converted to tokenized form and padded. These tokens are then passed to the LSTM model. This model is initialized with glove vector embeddings. The model was now concatenated with the rest.

The function of this model is to extract useful sequential features from the given caption embeddings and update them to the optimal weights, which are responsible for the generation of captions in the inference step.

4.5.3. Decoder

This part of the model concatenates the CNN and LSTM parts. The concatenated data is fed to a 256 unit dense layer followed by a dropout layer to prevent overfitting. For the inferencing part, another dense layer has been added with the softmax activation. The number of units in this layer has been set equal to the vocabulary size. This outputs a sequence of next probable words to create a generated caption. For more complex models, LSTM layers can also be added to this part of the model.

4.6. Model deployment

Model deployment basically involves incorporating a system of machine learning into a current development environment in which an input can be taken and the output produced. The definition of data science and machine learning implementation relates to the development of a process using new data for prediction. The aim of implementing your model is to make the model accessible to others, such as customers, administrators, or other programmers, the predictions from a pre-trained ML model.

For deployment of the model, Node Js for Backend support along with the EJS template is used. EJS is a template language that generates plain JavaScript HTML markups. Node.js is one of the most popular technologies for web development, and Python provides support for deep learning, artificial intelligence, and machine learning libraries.

Though Node JS and python aren't generally used together, using the spawn method from the child process module we have integrated these two powerful languages. Node JS's Child Process module offers features for running Python scripts or commands. Machine learning algorithms, deep learning algorithms, and several functionalities supplied to the Node JS framework from the Python library are introduced. Child Method allows the Node JS programmer to run a Python script and stream in / out data into/from a Python script.

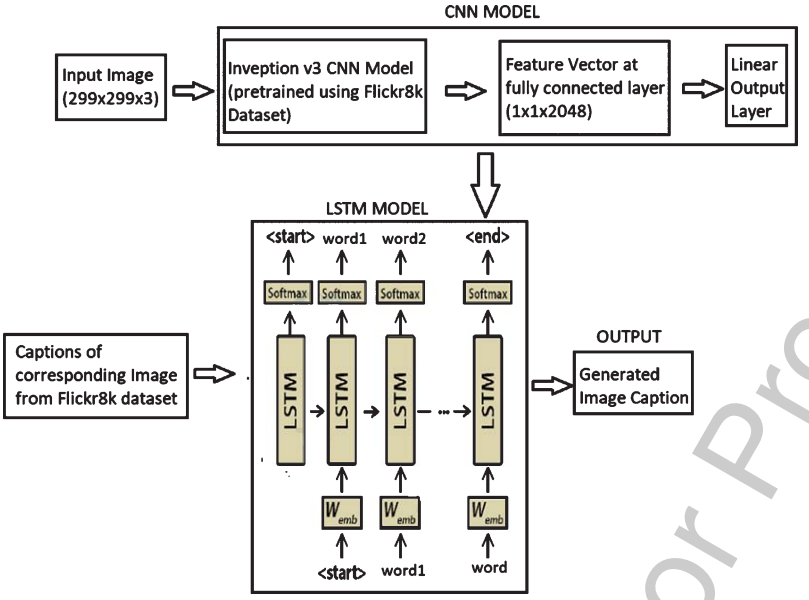

Fig. 4. Flow Diagram of the working model.

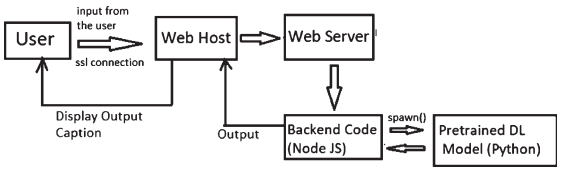

Fig. 5. Model Deployment on Web using Cloud-based server.

Web hosting is a facility that requires a web site or web page to be put to the Internet by organizations and people. A provider of web hosting services is an organization that offers the technology and services necessary for the website or database to be accessed on the Internet. Websites are hosted on separate machines called servers or are kept there.

A shared cloud-based infrastructure (PaaS) hosting system with a managed container system, optimized application services, and a strong architecture comprising LINUX servers is used for web hosting and deployed on dynos. A custom domain name with a '.ml' domain along with custom nameservers and HTTPS protocol with an SSL certificate is used for the model deployment. To create an encrypted connection between the two systems, SSL is the standard security technology. There may be user-to-server browsers, application-to-server, or user clients. Essentially, SSL means that the data transmission remains encrypted and private between the two systems. A secure SSL connection from the web host is requested by a user accessing the page. The host responds with a legitimate SSL certificate and provides a secure connection with encrypted data transfer. For a local test server, 'ngrok' is used on the backend. Ngrok requires a website server that is operating on a local computer to be open to the public. This sets up an HTTP connection to a local server port. The program allows the locally managed web server appears to be managed on a ngrok.com subdomain, which means there is no need for a public address or domain name on the local computer.

5. Results analysis and comparative study

Table 3 performs all the three models using CNN for image encodings. LSTM performs the best out of the three in all terms. It achieved an accuracy of 85.84% while the other to models' accuracies were 56.37 for GRU and 71.22 for Bi-directional LSTM implying that Bi-directional LSTM performed better than GRU. Similar results were shown in the loss values. LSTM model has the least loss followed by the Bi-directional LSTM and then GRU. The accuracy and loss plots are displayed in Figs. 6 and 7 respectively. Each of the models was trained for 200 epochs. All of them showed the same curve for the increase in the accuracy however LSTM had the smoothest steep implying higher accuracy in lower epochs. It achieved 80% accuracy after 180 epochs. Similarly, in the loss plots, all of them again showed the same

Table 3
Model performance results

Model	Accuracy (%)	Loss	Bleu 4 score
LSTM	85.84	0.4093	0.558
GRU	56.37	1.5495	0.535
Bi-directional LSTM	71.22	0.9563	0.534

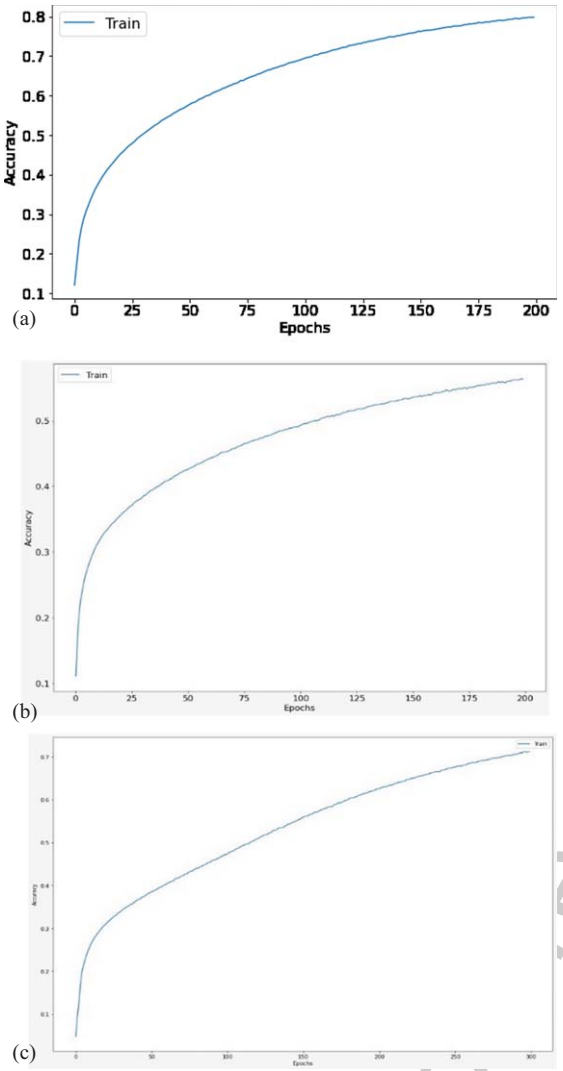

Fig. 6. Accuracy Plots for (a) LSTM (b) GRU and (c) Bi-directional LSTM.

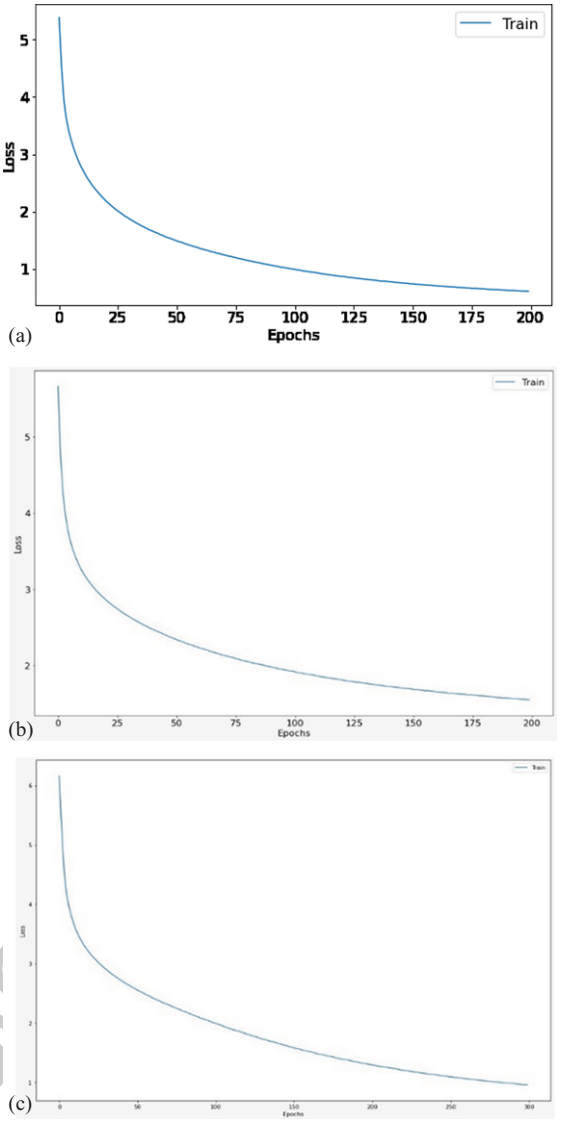

Fig. 7. Loss plots for (a) LSTM (b) GRU and (c) Bi-directional LSTM.

Table 4
Bleu scores

Bleu-n gram	Average	Best match
Bleu - 1	0.401	0.4251
Bleu - 2	0.405	0.6741
Bleu - 3	0.489	0.749
Bleu - 4	0.558	0.8029

downward curve. In LSTM, the loss decayed below 1.0 after 200 epochs.

Another evaluation metric that was used to test the performance of the model is the Bleu Score. Bleu score measures the similarity in the predicted captions and the machine-translated captions and gives a score between 0–100, 100 being absolutely accurate. Table 4 shows the ngram overlap for the range 1–4 Bleu score for the LSTM model. The Bleu 4 score obtained is 55.8 which is showing great performance. Table 3 shows the Bleu 4 score for other models which are slightly lower than the LSTM model.

Table 5
Comparative results with other research works

Research Work	Bleu 4 Score
Quang Zeng et al. [3]	0.23
Mulachery et al. [4]	0.225
Proposed CNN-LSTM model	0.558

Fig. 8. Dog is running through the grass.

Fig. 9. Man in red shorts is standing in the middle of a rocky cliff.

Table 5 compares the Bleu 4 Score results obtained by the proposed CNN-LSTM model with the research works of Quangzeng et al. who obtained a bleu 4 score of 0.23 and Mulachery et al. who obtained a bleu 4 score of 0.225. Compared to the other two research work's results, the bleu 4 score obtained by the proposed CNN-LSTM was 0.558 which is pretty high and shows greater performance as well as utility in real-life applications.

Figures 8 and 9 shows the image and the corresponding caption generated by the proposed CNN-LSTM model.

6. Conclusion and future work

Image captioning is an emerging field and adds to the corpus of tasks that can be achieved by using deep learning models. Various research published over recent years were discussed. For each part, a modified or replaced component is added to see the influence on the final result. The Flick8k dataset was used for training and evaluated the model using the BLEU score. To extend the work, some of the proposed modifications are using the bigger Flickr 30 k dataset along with the coco dataset, performing hyperparameter tuning and tweaking layer units, switching the inference part to beam search, which can look for more accurate captions, using scores like meteor or glue. Also, the codes can be organized using encapsulation and polymorphism.

References

[1] GloVe: Global Vectors for Word Representation (Stanford) "We use our insights to construct a new model for word representation which we call GloVe, for Global Vectors, because the global corpus statistics are captured directly by the model."
[2] Flickr8k dataset from Kaggle website.
[3] Quangzeng, et al., "Image Captioning with Semantic Attention, 2016 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*"
[4] V. Mullachery and V. Motwani, "Image Captioning, 2016 Arxiv"
[5] N. Krishnamoorthy, S. Guadarrama, S. Venugopalan, R. Mooney, G. Malkarnenkar, K. Saenko and T. Darrell, "*The IEEE International Conference 2013*"
[6] D. Elliott and F. Keller, "Image description using visual dependency representations. *EMNLP*, 2013."
[7] A. Karpathy and L. Fei-Fei, "The *IEEE Conference on Computer Vision and Pattern Recognition*"
[8] Y. Wang, G. Cottrell, Z. Lin, X. Shen and S. Cohen, "Skeleton Key: Image Captioning by Skeleton-Attribute Decomposition, 2017 *IEEE Conference*".
[9] Q. Wu, C. Shen, P. Wang, A. Dick and A.V. Hengel, "Image Captioning and Visual Question Answering, in *IEEE Transactions on Pattern Analysis*".
[10] K. Fu, J. Jin, R. Cui, F. Sha and C. Zhang, "Aligning Where to See and What to Tell: Image Captioning with Region-Based Attention and Scene-Specific Contexts, in *IEEE Transactions* 2017".
[11] R. Vedantam, S. Bengio, K. Murphy, D. Parikh and G. Chechik, "Context-Aware Captions, 2017 *IEEE Conference*".
[12] Z. Gan, et al., "Semantic Compositional Networks for Visual Captioning, 2017 *IEEE Conference*".
[13] C. Gan, Z. Gan, X. He, J. Gao and L. Deng, "StyleNet: Generating Attractive Visual Captions with Styles, 2017 *IEEE Conference*".
[14] L. Yang, K. Tang, J. Yang and L. Li, Dense Captioning with Joint Inference and Visual Context, 2017 *IEEE Conference*.

[15] R. Krishna, J. Krause, L. Fei-Fei and J. Johnson, A Hierarchical Approach for Generating Descriptive Image Paragraphs, 2017 *IEEE Conference*.

[16] S. Venugopalan, L.A. Hendricks, M. Rohrbach, R. Mooney, T. Darrell and K. Saenko, "Captioning Images with Diverse Objects,".

[17] O. Vinyals, A. Toshev, S. Bengio and D. Erhan, "Show and tell: A neural image caption generator, 2015 *IEEE Conference on Computer Vision and Pattern Recognition*".

[18] F. Liu, T. Xiang, T.M. Hospedales, C. Sun and Yang, "Semantic Regularisation for Recurrent Image Annotation, 2017 *IEEE Conference*".

[19] P. Sharma, N. Ding and S. Goodman, Radu Soricut Google AI Venice.

[20] M. Zuckerberg and A. Sittig, S Marlette - US Patent 7,945,653, 2011 - Google Patents.

[21] R. Kiros, R. Salakhutdinov and R. Zemel, Multimodal neural language models, In *ICML*, (2014), pp. 595–603.

[22] R. Salakhutdinov and R. Zemel, Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, (2014).

[23] R. Socher, A. Karpathy, Q.V. Le, C.D. Manning and A.Y. Ng, Grounded compositional semantics for finding and describing images with sentences, *Trans of the Association for Computational Linguistics(TACL)*, **2** (2014), 207–218.

[24] H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, P. Doll´ar, J. Gao, X. He, M. Mitchell and J. Platt, From captions to visual concepts and backm *In CVPR* (2015), pp. 1473–1482.

[25] X. Chen and C. Lawrence Zitnick, Mind's eye: A recurrent visual representation for image caption generation, *In CVPR* (2015), pp. 2422–2431.

[26] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel and Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, *ICML*, (2015).

[27] M. Elhoseny and K. Shankar, Optimal Bilateral Filter and Convolutional Neural Network based Denoising Method of Medical Image Measurements, *Measurement* **143** (2019), 125–135. DOI:https://doi.org/10.1016/j.measurement.2019.04.072

[28] J. Donahue, L.A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko and T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, In *CVPR*, (2015), pp. 2625–2634,

[29] A.A. Shah, S.A. Parah, M. Rashid and M. Elhoseny, Efficient image encryption scheme based on generalized logistic map for real time image processing, *Journal of Real-Time Image Processing*, (2020), In Press DOI:https://doi.org/10.1007/s11554-020-01008-4

[30] S. Kalajdziski, ICT Innovations 2018. *Engineering and Life Sciences. Cham: Springer*. p. 220. ISBN 9783030008246. (2018).

[31] S. Albawi, T.A. Mohammed and S. Al-Zawi, Understanding of a convolutional neural network. In *Proceedings of the 2017 International Conference on Engineering and Technology (ICET)*, Antalya, Turkey, **21–23** (2017), 1–6.

[32] J. Gomes and L. Velho, Image Processing for Computer Graphics and Vision, *Springer-Verlag*, (2008).

[33] R.C. Gonzalez and R.E. Woods, Digital Image Processing, *Third Edition. Prentice Hall*, (2007).