International Journal of Science & Technology (PRAGYANAM)
Volume 1 • Issue 1 • January–December 2025
ISSN:

Visual Narratives: Enhancing Image Captioning with CNN-RNN-LSTM Fusion

Kapil Kumar Choudhary[1], Naveen Kumar[2] & Dishant Kumar[3]

ABSTRACT

This paper introduces an effective image captioning system that has been developed using the most advanced deep learning techniques. The current study uses a diverse Flickr dataset with the pre-trained VGG16 model for feature extraction and LSTM networks for caption generation. The system is very effective in creating meaningful captions for all types of images, which increases accessibility for visually impaired individuals. More importantly, the text-to-speech functionality is inappropriately integrated, and thus the generated captions are accessible through verbalized words. The paper deals with system architecture, data preprocessing subtleties, and evaluation measures, thus giving an appropriate overview of the results and implications in the real world.

Keywords: VGG16, LSTM, BLEU, Caption Generation

INTRODUCTION

In recent years, the integration of computer vision and natural language processing has experienced significant advancements, particularly in the domain of AI-driven image captioning. This field specializes in the development of algorithms and models capable of automatically generating descriptive and contextually relevant captions for images. The growing prevalence of social media, ecommerce platforms, and autonomous vehicles has amplified the demand for systems that can interpret and decode visual data in a human-like manner [9, 20].

Early approaches to image captioning primarily relied on encoding visual information into a single feature vector representing the entire image [13, 23]. However, such methods often neglected critical details about objects and their spatial and semantic relationships within a scene, limiting descriptive accuracy and contextual understanding [29]. To address these challenges, recent research has focused on incorporating spatial and semantic relationships and leveraging attention mechanisms to enhance the generation of coherent and detailed captions [2, 12, 27].

This paper provides a comprehensive survey of recent advancements in AI image captioning, emphasizing innovative techniques that utilize spatial relationships, attention mechanisms, and transformer architectures to improve caption accuracy and contextual relevance [5, 20, 24]. We begin with a discussion of foundational principles and the technical challenges of implementation. Subsequently, we explore key developments in the field, including advancements in object detection [11], geometric and semantic excitation methods [25], and hybrid transformer-based architectures [5, 29].

RELATED WORK

- **Early Deep Learning Models:** This includes models that uses Convolutional Neural Networks (CNNs) for the extraction of image feature and Recurrent Neural Networks (RNNs) or alternatives like Long Short-Term Memory (LSTM) networks for generating captions.
- **CNN-RNN Based Models:** Focus on more recent developments that combine both CNNs and RNNs/ LSTMs for image captioning. It discusses how

[1] Department of Mathematics, IIIT, Kota, Rajasthan, India. E-mail: kapil.maths@iiitkota.ac.in
[2] Department of Mathematics Gurugram University , Gurugram, Haryana. E-mail: naveen@gurugramuniversity.ac.in
[3] Department of Mathematics IIIT, Kota, Rajasthan, India. E-mail: 2021kucp1068@iiitkota.ac.in

CNNs are used for encoding image features while RNNs/LSTMs are used for generating captions.

- **Evaluation Metrics:** Briefly touch on the evaluation metrics commonly used in assessing the performance of image captioning models, such as BLEU, METEOR, CIDEr and ROUGE. In this research paper, we are using BLEU as our evaluation Metrices Discuss their strengths and limitations in capturing the quality of generated captions.
- **Challenges and Limitations:** Highlight challenges and limitations faced by existing image captioning models, such as handling different domains of image content, generating descriptive and coherent captions, and scalability to large datasets.

METHODOLOGY

- **Data Collection:** We carefully Curate a diverse dataset of images and corresponding captions from reliable sources, ensuring representation across various domains and scenarios

  Dataset: We trained our model with flickr8k Dataset, which comprises of approximately 8000 images, and for each image there are up to five captions per image. The images we choose are from six different Flickr groups, and these images do not contain any well-known people or locations, but were manually selected to depict a variety of scenes and situations.
- **Data Preprocessing:** Employ robust data preprocessing techniques such as resizing, normalizing to enhance the quality and diversity of the data set for optimal model training.

  We converted the text into lowercases and resized all the images. we also removed the additional spaces, special characters, and digits from the texts. We added startseq in the starting of each caption and endseq in the end of each caption that is generated by our model.

```
# before preprocess of text
mapping['1000268201_693b08cb0e']
```

```
['A child in a pink dress is climbing up a set of stairs in an entry way .',
 'A girl going into a wooden building .',
 'A little girl climbing into a wooden playhouse .',
 'A little girl climbing the stairs to her playhouse .',
 'A little girl in a pink dress going into a wooden cabin .']
```

```
# preprocess the text
clean(mapping)
```

```
# after preprocess of text
mapping['1000268201_693b08cb0e']
```

```
['startseq child in pink dress is climbing up set of stairs in an entry way endseq',
 'startseq girl going into wooden building endseq',
 'startseq little girl climbing into wooden playhouse endseq',
 'startseq little girl climbing the stairs to her playhouse endseq',
 'startseq little girl in pink dress going into wooden cabin endseq']
```

- **Data Partioning:** Divide the dataset into appropriate training, validation and testing subsets to ensure a reliable evaluation of the model's performance and generalization capabilities.

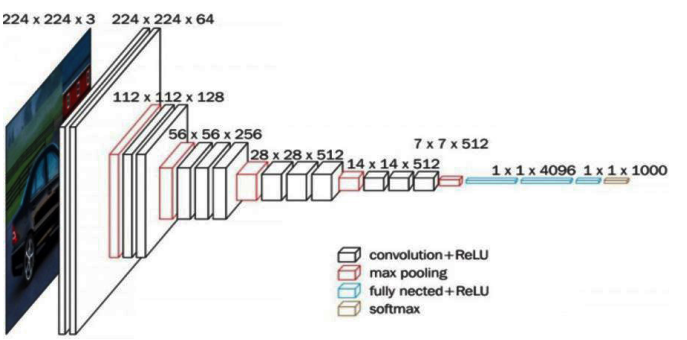
Description	*Count*
Total Images	8091
Training Data	7282
Testing Data	1009
Flickr8k Token (text)	40455
Flickr8k Lemmatized Token (text)	40455

- **Model Selection:** Choose an appropriate deep learning model architecture, such as Convolutional Neural Network (CNN) for the purpose of image feature extraction and a recurrent neural network (RNN).

Base CNN Architecture

A. **Pre-trained CNNs:** It leverages the power of pre-trained convolutional neural networks like VGG16. These models have been trained on large image datasets and they can extract rich and quality visual features from images.

B. **VGG16:** This is the pre-trained CNN model having 16 layers, which was trained in that model. The CNN-based VGG16 was trained with a dataset of ImageNet. Developed by the Visual Geometry Group of Oxford University. The 16 layers of VGG16 have 13 convolutional layers and 3 fully connected layers. It basically is used as the first stage in any image recognition processes. The fully connected layers in the network do predictions based on the characteristics learned from the convolutional layers. The layers use the retrieved characteristics as inputs for classifying input images to a predefined set of classes.



CNN is a general term used in image processing related algorithms. CNN evolves from a very simple ANN. CNN gets a better outcome over images and this simple dense network tends to work perfectly whenever there are specific features used while classifying the image with classification work. The CNN does very well with some more feature inside an image and hence used for processing it in local features too. Since images are made of repeated patterns of a particular thing any image. It takes the images as input and then it understands that input for the assigned task. Among all, CNN has basically two functions convolution and pooling. In that, Convolution is used in CNN to determine an edge of an image and reduction of an image size uses pooling. It is a technique where we will take a small size matrix called kernel or filter, and after that we will move it to over our image and convert it according to the filter values. The formula for the feature map is Thus, symbols f and h denote the input image and filter respectively, while m, n denote the row and column indices of the matrix that results from:

$$G[M, N] = ([F * H])(M, N)$$
$$S = \sum_J \sum_K H[J, K] F[M - J, N - K]$$

Two main computational procedures realize the convolution layer calculation, performed below. The step is applied to calculate the intermediate value Z, and its addition by distortion. Next in line is a non-linear activation g applied to the intermediate value.

$$Z^i = W^i * A^{i-1} + b^i$$
$$A^i = g^i * Z^i$$

Recurrent Neural Network (RNN)

Generally, CNNs do not perform well on sequential data when input data is connected. In the case of CNNs, there is no interaction between previous inputs and subsequent data. Hence, all the outputs depend solely on the input at that moment. Depending on the trained model, CNN takes the input and gives an output.

For performing tasks requiring sequence-based relationships, RNNs are used. RNNs have memory, allowing them to remember what has passed or happened earlier in the data. "Earlier" refers to the previous inputs that has passed. RNN performs best on textual data because textual data is interrelated or we can say that text data is sequential data. Basic formula for RNN is written as follow:

$$h(t) = f(h^{t-1}, x(t); \theta)$$

Here:

- f represents the hidden state function.
- $h(t-1)$ is the previous hidden state.
- $x(t)$ is the current input.
- θ denotes the parameters of the function.

LSTM Network Design

LSTM is the variant of RNN. It performs much better than simple RNN because it offers the solution for the problems faced by simple RNN. Two major problems encountered with Simple RNN are:

- exploding gradient and vanishing gradient.
- long term dependency.

LSTM uses the gates for memory; however, gates are the core of LSTM. There are the following in the list of LSTM gates are:

- input gate
- forget gate
- output gate

All of them have sigmoid activation function. Sigmoid: Output is between 0 and 1, mainly 0 or 1. If output is 0, then it blocks. If output is 1, then pass everything.

Number of LSTM Layers

The number of layers of LSTM impacts the capturing of long-term dependencies in the image and more complex captions being generated .Experiment with different configurations of layers to find a good depth for your application.

Size of the Hidden State

The size of the hidden state within an LSTM cell determines at any point in time what information the model can hold and process. A greater size in the hidden state increases reasoning but requires more computation.

C. **Additionl Consideration**

Beam Search Decoding: Beam search can actually be applied to explore multiple candidate captions and choose the best one for a scoring function, although it would increase diversity and general quality of the generated captions.

MODEL TRAINING

After designing the model and preparing and before that, preprocess the data followed by fitting the data with Repeat the model and apply it in turn to the training data. The training procedure is very computer resource intensive, because of it this requires cycling through the entire training set that is available or we are using numerous times, and then we will estimate the loss function. Passing the training data set into the model numerous times is called epochs. Therefore, instead of loading the whole data set in one go, we come up with a new method or way that passes that data gradually by means of a generator, that is called Progressive Loading. We can see that the size of any data set is proportional. In terms of memory requirements. However, we are trying to design one of the efficient methods of this, and hence we will create a generator function that is called a data generator. This generator will generate a function that will accept an image array as input: embeddings and encoded word sequences, it will produce one shot encoded words one after another. These one-shot encoded words in addition to their embeddings, will feed the image progressively in two sets of neural networks and we will sequentially train the data to produce the good weights. After loading of data, we will come to the training of the model. Here we will apply 20 epochs and each of the epochs will contain 6000 images for training. It is to take a note to mention that the number of epochs that we have taken is on intuitive basis i.e. based on instinct and natural understanding. Quite fortunately, these numbers of epochs have proved to drastically reduce the model loss, and these epoch also prevent under-fitting and over-fitting of the model. Moreover, as we traditionally train our model it takes some amount of time-span, so it is worth of using model checkpoints which periodically saves the progress of model after the completion of each epoch. In this way, even if a process suddenly stops due to a reason, it resumes so that no significant progress gets lost. It works efficiently with its memory without causing a system crash even when working on large data sets, and one is able to get a full view during training and validation of loss curves, fine-tune hyper-parameters, and judge performance of a model. This would, besides monitoring validation metrics and applying the early stopping, give over-fitting another level of protection and would build a tougher model, potentially saving compute in later epochs.

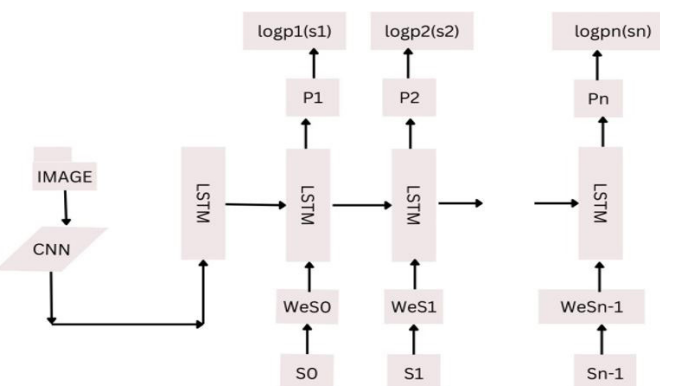
MODEL IMPLEMENTATION

We are using CNN combined with LSTM, for developing an efficient model that is require to generate captions of the images. The model will take input as an image and produce text as an output. We will use Keras functional API model to stack the model. There will a total of three sections to the structure:

A. **Feature Extractor:** This one is being used to reduce the dimension from 4096 down to 256. Dropout Layer will be used here. One of them will add the CNN and LSTM. Its characteristic is that this model will predict what we are feeding into it, with having the photographs already been pre-processed, we will use the Xception model without the final classification layer.

B. **Sequence Processor:** It processes input text through the Embedding layer and LSTM layer.

C. **Decoder:** We will combine all the outputs of the last two layers by using a dense layer to come up with our final predictions. The feature extractor and the sequence processor produce an output vector of fixed length. A dense layer processes them after combining. In the final layer, the number of nodes will be equal to our vocabulary size. Considering the words already formed and the visual context preserved in characteristics of the image, the LSTM learns to form words word-by-word. The model has attention features that enable it to focus on relevant picture areas while generating captions. The model updates its parameters by training on the paired image-caption data to give captions that accurately represent the content of the input pictures. CNN + LSTM enables the creation of relevant and coherent captions for pictures through the combination of visual feature extraction with sequential caption creation.



D. **Evaluation Metrics**

BLEU Score: BLEU is, in fact, a statistical measure used to determine the quality of automatically generated captions through comparison with human-made reference captions. BLEU that is known as Bilingual Evaluation Understudy. This evaluation mechanics or matrix is widely used in textual generation.This is a comparison of the machine-generated text with one or more texts to those are manually written by human. Basically, it says how close a generated text is to an expected text. It is majorly used in automated machine translation, though it can be used with image Captioning, text summarization, speech recognition, etc. Especially in image captioning, BLEU score is the correctness that how close a generated caption is to a manual human generated caption of that particular image. The score scale lies between 0.0 and 1.0. Here 1.0 represents a best score and 0.0 is the worst score. We know all of them use BLEU score as an evaluation matrix and they evaluated BLEU-1 to 4 where:

BLEU-1 = Precision of uni-gram matches (single words).

BLEU-2 = Precision of 2-grams (pairs of consecutive words).

BLEU-3 = Precision of 3-gram matches.

BLEU-4 = Precision of 4-gram matches.

Epoch	5	15	20
Loss	3.2707	2.1933	2.7436
Bleu1	0.453804	0.553867	0.51345
Bleu2	0.236148	0.335146	0.304321

MODEL ARCHITECTURE

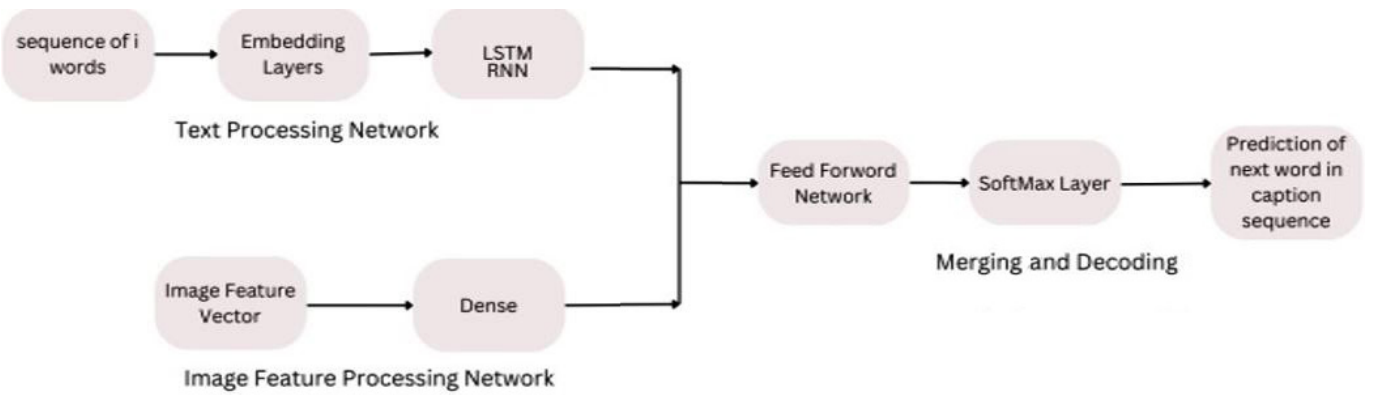
In the above data preparation segment, we had already prepossessed two distinct units of data. one of them is image data and the other one is caption data and now both of them will create encoded image feature vectors and encoded text embeddings, respectively. Both of the sets represent the preprocessed data, and both should be able to suitable for fitting into the neural network. Regardless image feature vectors and text embedding both cannot be combined in the same input layer of a singular neural network. As a result of this it is essential to have distinct input layers for the two types of data inputs. Following this model, the idea of the 'merge model' becomes relevant.

Merge Model

In the merge model, we will integrate two distinct types of encoded input data neural networks which is then subsequently or sequentially processed by a simpler decoder network to produce the next or upcoming word in the caption sequence. The preprocessed text is then taken as input into an embedding layer, followed by a recurrent neural network known as LSTM (Long Short-Term Memory). While that process continues in the same time the image features are processed through a densely connected neural network layer, followed by a feed-forward network, culminating in a softmax layer. The outputs encoded by the LSTM in the first neural network are combined with the encoded image embeddings from the second neural network in the second dense layer, allowing for the decoding process to predict the next word in the caption sequence.

Long Short-Term Memory (LSTM)

In the first network, we're working with text data, so we're using Natural Language Processing (NLP). For NLP, the best type of neural network is usually a recurrent neural network (RNN). Among RNNs, the long shor-term memory (LSTM) network is preferred because it helps in the vanishing and exploding



gradient problems. LSTMs can back-propagate errors over many time steps and have three main gates: input, forget, and output gates.

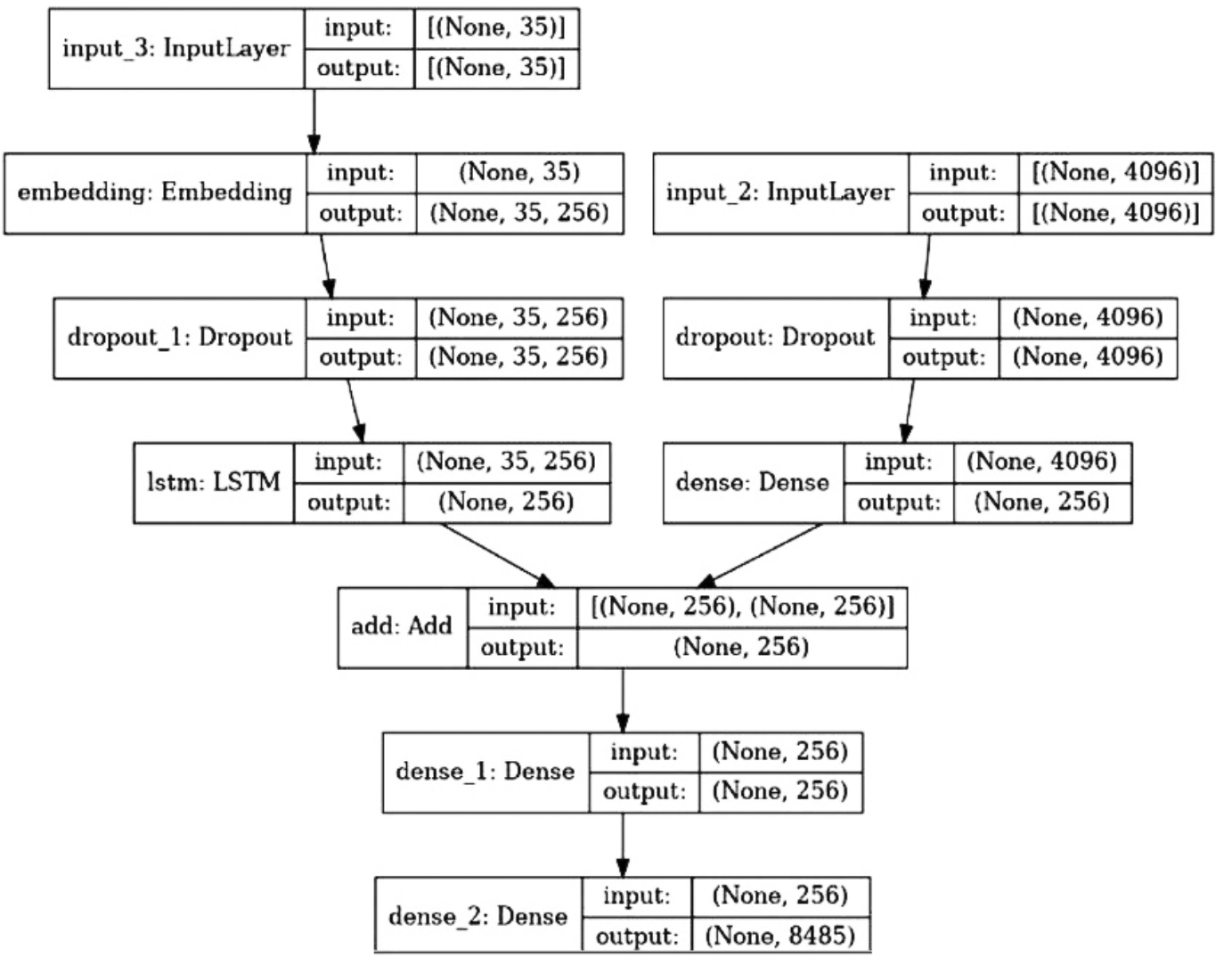
Model Designing

After combining all the parts inthe merge model and after that we will focus on making of architecture of neural network structure.The image feature processing network takes a 1D image embedding (4096 dimensions) as input. Next, it uses a regularization layer with 50% dropout to stop the model from learning too fast and overfitting during training. Then, a dense layer shrinks the 1D vector from 4096 to 256 dimensions before it reaches the addition layer. The text processing network handles a 1D text input vector (34 dimensions). It sends this through an embedding layer to turn words into word embeddings. Again, it uses 50% dropout to prevent overfitting while training. After that, it goes through anLSTM network of the same size to create rich detailed sentences for a given image. Once we have the results from both networks, we build the merging and decoding part of the neural network. This section has an addition layer to join the two outputs. Then, it has a dense 1D layer (256 dimensions) with a 'Rectified Linear Unit' activation followed by another dense layer of the same size with 'softmax' activation.

RESULT

BLEU Performancez

After training the model, we try to determine the accuracy and efficiency of the model in generating the respective captions of the respective images. As discussed earlier in the above section BLEU metric is widely used to determine the accuracy and efficiency of word generation. We would work out the procedure



for generating text with the help of 1 and 2-gram BLEU scores. Each gram represents a different weight. Below are some observations on BLEU evaluation metrics on different BLEU gram scores.

BLEU-1	1.0,1.0,1.0,1.0	.553867
BLEU-2	0.5,0.5,0,0	.335146

Table. Comparison between original and predicted descriptionof images

Image	*Original Description*	*Predicted Description*
101669240_b2d3e7f17b.jpg	Man skis past another man displaying painting in the snow	Two people are hiking up snowy
1002674143_1b742ab4b8.jpg	Small grid in the grass plays with finger paints in front of white canvas with rainbow on it	Little girl in pink dress is lying on the side of the grass

CONCLUSION

The results have shown the successfulness of the deep architecture used in this research. Together, the CNN and the LSTM model were able to capture and synchronize their functions in recognizingrelationships between objects in an image. Therefore, this synchronization between the two architectures signifies the ability of deep learning to extract complex features for processing.Our experiments reveal that this method significantly works well towards generating an image caption, and this could be significantly improved further with the growth of dataset size and during training when there are even more varied images. A huge benefit the integration of this text-to-speech ability gives towards a visually-impaired individual to perceive surroundings much more clearly through clear and meaningful captions of an image.

The present model has been trained on the relatively compact and homogeneous Flickr8K dataset. It could be very interesting work in the future to train the model in larger and more diverse datasets, such as the Flickr30K and MSCOCO, which is highly expected to improve the robustness and accuracy of the system.

REFERENCES

[1] Jafar A Alzubi, Rachna Jain, Preeti Nagrath, Suresh Satapathy, Soham Taneja, and Paras Gupta. 2021. Deep image captioning using an ensemble of CNN and LSTM based deep neural networks. *Journal of Intelligent & Fuzzy Systems* 40, 4 (2021), 5761–5769.

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2017. Bottom-up and top-down attention for image captioning and vqa. arXiv preprint arXiv:1707.07998 2, 4 (2017), 8.

[3] 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*. 6077–6086.

[4] K AnithaKumari, C Mouneeshwari, RB Udhaya, and R Jasmitha. 2020. Automated image captioning for flickr8k dataset. In *Proceedings of International Conference on Artificial Intelligence, Smart Grid and Smart City Applications: AISGSC 2019*. Springer, 679–687.

[5] Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. 2022. Explaining transformer-based image captioning models: An empirical analysis. *AI Communications* 35, 2 (2022), 111–129.

[6] Yang Feng, Lin Ma, Wei Liu, and JieboLuo. 2019. Unsupervised image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4125–4134.

[7] Seung-Ho Han, Min-Su Kwon, and Ho-Jin Choi. 2020. EXplainable AI (XAI) approach to imagecaptioning. *The Journal of Engineering 2020*, 13 (2020), 589– 594.

[8] Sen He, Hamed R Tavakoli, Ali Borji, and Nicolas Pugeault. 2019. Human attention in image captioning: Dataset and analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 8529–8538.

[9] MD ZakirHossain, FerdousSohel, Mohd Fairuz Shiratuddin, and Hamid Laga. 2019. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys* (*CsUR*) 51, 6 (2019), 1–36.

[10] Feicheng Huang, Zhixin Li, Shengjia Chen, Canlong Zhang, and Huifang Ma. 2020. Image captioning with internal and external knowledge. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 535–544.

[11] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.

[12] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. 2019. Attention on attention for image captioning. In *Proceedings of the IEEE/CVF international conference on computer vision*. 4634–4643.

[13] 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.

[14] BurakMakav and VolkanKılıç. 2019. A new image captioning approach for visually impaired people. In *2019 11th International Conference on Electrical and Electronics Engineering* (*ELECO*). IEEE, 945–949.

[15] BorneelBikashPhukan and Amiya Ranjan Panda. 2021. An efficient technique for image captioning using deep neural network. In *Cognitive Informatics and Soft Computing: Proceeding of CISC 2020*. Springer, 481–491.

[16] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7008–7024.

[17] Himanshu Sharma, Manmohan Agrahari, Sujeet Kumar Singh, Mohd Firoj, and Ravi Kumar Mishra. 2020. Image captioning: a comprehensive survey. In *2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control* (*PARC*). IEEE, 325–328.

[18] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014).

[19] YajushPratap Singh, Sayed Abu LaisEzaz Ahmed, Prabhishek Singh, Neeraj Kumar, and Manoj Diwakar. 2021. Image captioning using artificial intelligence. In *Journal of Physics: Conference Series*, Vol. 1854. IOP Publishing, 012048.

[20] Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Silvia Cascianelli, Giuseppe Fiameni, and Rita Cucchiara. 2022. From show to tell: A survey on deep learningbased image captioning. *IEEE transactions on pattern analysis and machine intelligence* 45, 1 (2022), 539–559.

[21] Ishaan Taneja and Sunil Maggu. 2023. Generating Captions for Images Using Neural Networks. (2023).

[22] OriolVinyals, 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.

[23] OriolVinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2016. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence* 39, 4 (2016), 652–663.

[24] Yiyu Wang, JungangXu, and Yingfei Sun. 2022. End-to-end transformer based model for image captioning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 2585–2594.

[25] Haiyang Wei, Zhixin Li, Feicheng Huang, Canlong Zhang, Huifang Ma, and Zhongzhi Shi. 2021. Integrating scene semantic knowledge into image captioning. *ACM Transactions on Multimedia Computing, Communications, and Applications* (*TOMM*) 17, 2 (2021), 1–22.

[26] Jiahong Wu, He Zheng, Bo Zhao, Yixin Li, Baoming Yan, Rui Liang, Wenjia Wang, Shipei Zhou, Guosen Lin, Yanwei Fu et al., 2017. AI challenger: A large-scale dataset for going deeper in image understanding. arXiv preprint arXiv:1711.06475 (2017).

[27] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, RuslanSalakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*. PMLR, 2048–2057.

[28] Jufeng Yang, Yan Sun, Jie Liang, Bo Ren, and Shang-Hong Lai. 2019. Image captioning by incorporating affective concepts learned from both visual and textual components. *Neurocomputing* 328 (2019), 56–68.

[29] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. 2018. Exploring visual relationship for image captioning. In *Proceedings of the European conference on computer vision* (*ECCV*). 684–699.

[30] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and JieboLuo. 2016. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4651–4659.