International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 10 | Oct 2022 www.irjet.net p-ISSN: 2395-0072

Automated Image Captioning – Model Based on CNN – GRU Architecture

Hitesh Dhameja[1], Anmol Rohra[2], Megha Shahri[3], and Asha Bharambe[4]

[1-3] *Student,Department of Information Technology, Vivekanand Education Society Institute of Technology, University of Mumbai*

[4] *Professor,Department of Information Technology, Vivekanand Education Society Institute of Technology, University of Mumbai*

Abstract *- The images when presented to humans can easily identify the objects, relationships among the objects of the given image. Similarly, Image captioning is the model in the AI sector, where the image is sent as input, and then the objects, relationship between the objects are generated as captions. The model is based on encoder decoder architecture which is trained on flickr30k dataset. This paper describes the deep insights of deep learning techniques used for caption generation with convolutional neural networks and recurrent neural networks. The results are then translated into Hindi. The model is prepared with focus to help visually impaired people with voice assistant functionality.*

Key Words: Image captioning, deep neural networks, GRU & NLP, Artificial Intelligence.

1. INTRODUCTION

In accordance with the fast growth of information technology and the Internet, usage of mobile phones and photoshop apps/tools have made more and more images appear on social networking platforms. Hence, greater need for image captioning.

Image captioning is a process of generating image descriptions for a comprehensive understanding of the various components of the given image i.e objects present in the given image, the background of the given image, and the relationships between the objects present. Image caption is inclusive of two most important aspects of the Artificial Intelligence (AI) field, one being computer vision and the other being Natural Language Processing (NLP), and it is a very challenging problem until and unless we have a breakthrough solution to it.

Humans use any language to describe the image given to them. This leads to a better understanding of the whole scenario by generating captions out of the given images and later using those captions to thoroughly describe the image in a much easier manner. Various factors are required in order to get an in-depth understanding of an image such as the spatial and semantic information about the various components present in the image, and the relationships between all the objects of the image. The two major tasks that are required to generate captions from the images are as follows:

1. Getting information about the world i.e collecting the dataset. 2. Generating sentences to describe the given image. Using the Convolutional Neural Network (CNN) [9] - Recurrent Neural Network (RNN) [10] model, Different methods of Computer Vision, Gated Recurrent Unit (GRU), and NLP are used for extracting information from the images and representing them as meaningful sentences.

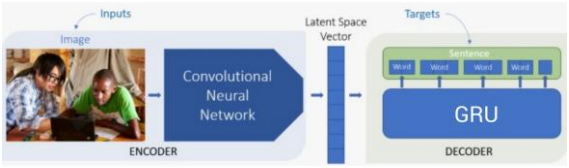


Figure 1. The flow of Image Captioning Model

In paper [11], the CNN-RNN model is incorporated with Long Short Term Memory (LSTM) [12] at the RNN part. But due to the large number of gates the computational power is slightly increased as compared to GRU hence we have performed the task with GRU. This paper's aim is to collect meaningful textual description from any given images, in the form of short yet meaningful captions. In order to make our model reach a vast number of users, these captions are further run into a text-to-speech model. This way, a fully independent and easy experience could be provided to users with short-sightedness. With the rise of social media platforms, these people don't have to feel like they are left out or lonely, especially when it comes to their loved ones sharing pictures on social media. Thus, we have implemented our approach of image captioning model and a detailed analysis of these approaches has been described in this paper

2. PROPOSED MODEL

The main purpose of our project in the ocean of machine learning was to basically make available a proven technique to avail the proper outcome from tons of algorithms and researches proposed. The thorough research was indeed to experiment in solving our problem with machine learning or deep learning techniques. As already a lot of research work is done in this area the primary focus was to develop a production level solution which can be run in almost real-time and can be integrated with any other application to

make best use of it. And hence, it has been made with integration ease in an object oriented approach.

2.1 Dataset

The most common datasets used for image captioning are MSCOCO, Flickr30k. It is because it contains a large number of images, and each image contains five captions associated with it. The human description helps to train the model well.

MS-COCO (Microsoft Common Objects in Context): The dataset is rich in a variety of images containing objects and scenes of different categories and hence it can be used for real time object detection, object segmentation, key point detection and image captioning. The dataset is labeled and is available publicly. The recent version which has 328K images was released in the year 2017 which has around 118K training images, 5K validation and 41K testing images and additional 123K unannotated dataset[5]. It also serves as a supplement to transfer learning and is widely used for training, testing and polishing the models. The dataset has around 328K images.

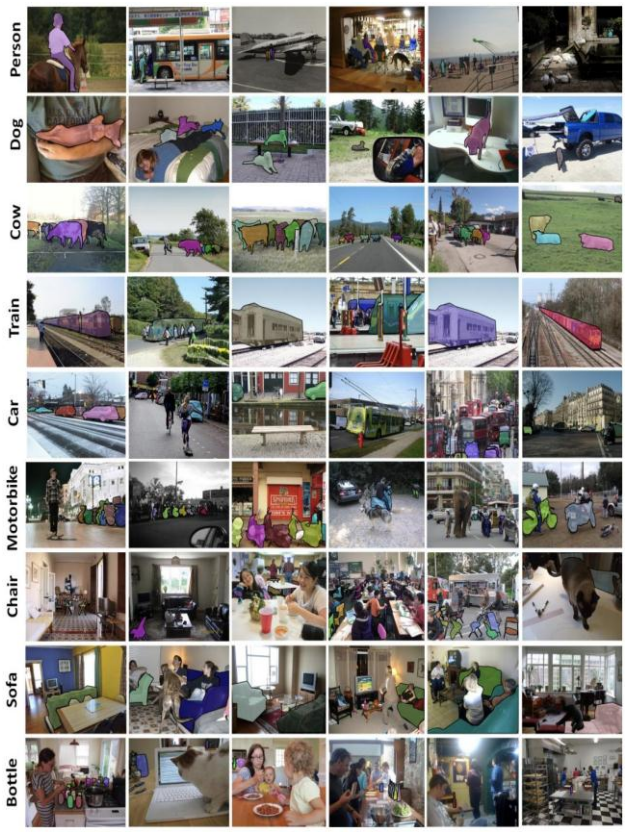


Figure 2. MS COCO Dataset Images [2]

Flickr30K: The Flickr30K dataset is used widely for sentence-based image description, which has a total of 158k captions with 244k coreference chains, where different captions for the same image with same entities is linked and associated with 276k manually annotated bounding boxes.

Figure 3. Flickr30k Dataset Images [3]

Like for example in the Fig.3 below, as the glasses are bound by a pink box so the word 'glasses' in description is also in pink. Similarly, in the rightmost image in Fig.3 the 'bride' is enclosed in a blue box and hence the word 'bride' is in blue color in all the captions. This helps the model to learn how the objects that appear visually in the images correspond to how they are called in captions. For this paper, Flickr30k has been used containing 30,000 images with 80:20 ratio for training and testing.

2.2 Preprocessing

The most important phase is this phase because if we neglect this phase the results could be quite devastating from the presumed ones. Our model requires the preprocessing for two parts. The one for the encoder phase and the other one for the decoder phase.

Let's first understand preprocessing for Convolution Neural Network i.e. encoder phase. The encoder model is based on Resnet50 architecture and hence the input image size must be 224 x 224. It can accept any image format files. The keras application module helps in preprocessing the images with the help of transfer learning on the Resnet50 model. The image features are thus extracted and saved in a file. While extracting image features, the second last layer a.k.a

classification layer has been removed because of transfer learning. These features are reshaped to an array of values between 0 and 1 of the size 2048 with numpy module.

```
[ ]  ## some code for code checking

     print(images_features['10002456.jpg'])
     print(len(images_features['10002456.jpg']))
     print(len(images_features))

     [0.18348733 0.26369268 0.7454376  ... 1.7963673  1.408203   0.01206312]
     2048
     31783
```

Figure 4. Image Feature Extraction

For the data processing at the decoder part (Gated Recurrent Unit - GRU) the following steps are applied:

- The captions from the dataset are made cleaner by removing punctuations, spaces, numerical or special characters.
- A '<start>' and '<end>' token is added to start and end of the captions so that during the training phase the decoder understands the starting and ending point of a particular caption.
- Caption tokenization to create a list of unique words by differentiating them on the basis of blank spaces present between them. As we have used flickr30k dataset which gives about 150k captions hence the vocabulary size is threshold to 18000 words.
- The above generated words are converted to 0's and 1's sequence. Later word map indexing followed by padding these sequences to form the largest sequence possible. In our case the length of the largest sequence is 74.

2.3 Architecture

The most widely accepted architecture is Encoder-Decoder architecture. In our case the encoder part is Convolutional Neural Network (CNN) and the decoder part is Grated Recurrent Network (GRU) which is a type of Recurrent Neural Network (RNN). The generic model representation of the architecture is shown below.

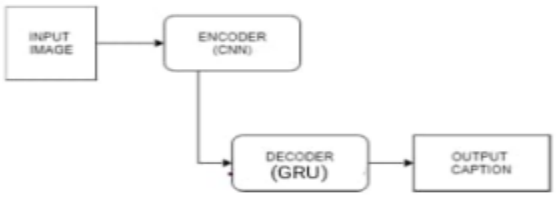


Figure 5. Model Architecture

In the above block diagram, the image features are extracted in the CNN and act as input to the GRU which will process the features with word vocabulary and predict the output caption.

1) Encoder

The encoder in our model is Convolution Neural Network (CNN). The image features are extracted in this step with the help of preprocessing and transfer learning through Resnet50 architecture. The second last layer i.e. softmax layer has been removed as classification is not required. The output is a vector of (512, 2048) shape.

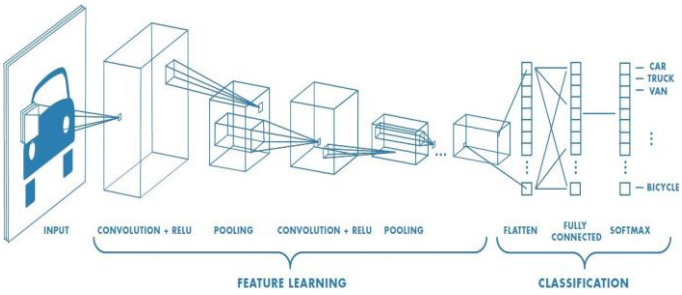


Figure 6. CNN Preprocessing

2) Decoder

The most actively used decoder architecture is LSTM (Long Short Term Memory). On the other hand we have implemented it with GRU (Gated Recurrent Unit) because there isn't much difference between LSTM and GRU but the later one doesn't have the cell state but makes use of hidden state to transfer the information.

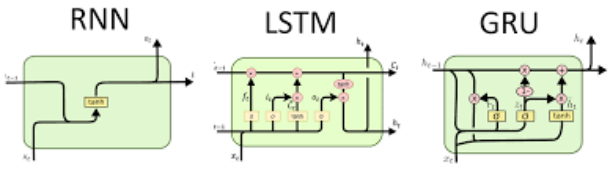


Figure 7. Variations in RNN, LSTM, GRU [7]

3) Training Phase

The training phase begins with splitting the dataset into training and testing phase. Since each image has 5 captions so if we randomly shuffle them then some train data will leak into the Validation data. Hence we will use the k fold group fold method to make sure each caption of the same image stays in the same set. The process is as discussed below.

- Mix up the data set randomly.
- Divide it into n parts.
- For each distinct part:
  - Take that part as test data set
  - While consider the remaining parts as training data set
  - Considering the above training and testing parts, train and evaluate the model.

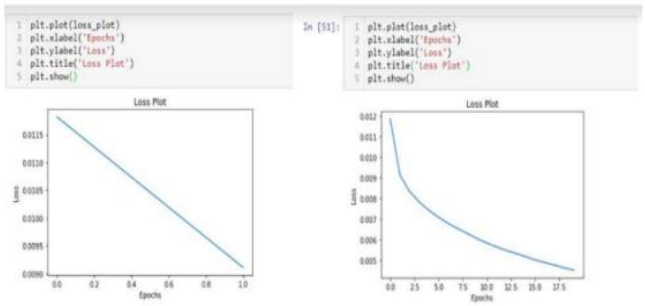


Figure 8. Difference in Loss Evaluation by increasing the epochs [4]

As the dataset is quite large enough we can't pass it into a single batch. The batch basically split the entire dataset into equal parts of batch size so the training can be done in those batch parts. We have used the batch size of 64. Adam optimizer was used with learning rate (lr) of 3e-4 and the loss function used is SparseCategorical Crossentropy passing logistics as true.

The training phase is associated with a callback function for early stop ping. If the accuracy is not increasing with the initial learning rate then it will be decreased by a factor of 0.2 until the threshold lr= 1e-5 doesn't reach.

The larger the epochs considered it is assumed that the more the accuracy will be. The below diagram explains the same. In our model we have trained for 36 epochs. After every epoch the trained weights are saved for back up in 'pickle' format. The entire training phase was done on GPU of google colab for faster processing.

Model: "model"

Layer (type)	Output Shape	Param #	Connected to
decoder_input (InputLayer)	[(None, None)]	0	[]
decoder_embedding (Embedding)	(None, None, 512)	9161728	['decoder_input[0][0]']
decoder_gru1 (LSTM)	(None, None, 512)	2099200	['decoder_embedding[0][0]']
encoder_input (InputLayer)	[(None, 2048)]	0	[]
decoder_gru2 (LSTM)	(None, None, 512)	2099200	['decoder_gru1[0][0]']
dense (Dense)	(None, 512)	1049088	['encoder_input[0][0]']
decoder_gru3 (LSTM)	(None, None, 512)	2099200	['decoder_gru2[0][0]']
encoder_dense (RepeatVector)	(None, 73, 512)	0	['dense[0][0]']
dropout (Dropout)	(None, None, 512)	0	['decoder_gru3[0][0]']
concatenate (Concatenate)	(None, 73, 1024)	0	['encoder_dense[0][0]', 'dropout[0][0]']
lstm (LSTM)	(None, 73, 1024)	8392704	['concatenate[0][0]']
dropout_1 (Dropout)	(None, 73, 1024)	0	['lstm[0][0]']
lstm_1 (LSTM)	(None, 73, 512)	3147776	['dropout_1[0][0]']
lstm_2 (LSTM)	(None, 73, 512)	2099200	['lstm_1[0][0]']
dropout_2 (Dropout)	(None, 73, 512)	0	['lstm_2[0][0]']
decoder_output (Dense)	(None, 73, 17894)	9179622	['dropout_2[0][0]']

Total params: 39,327,718
Trainable params: 39,327,718
Non-trainable params: 0

Figure-9: Different Layers of Proposed Model

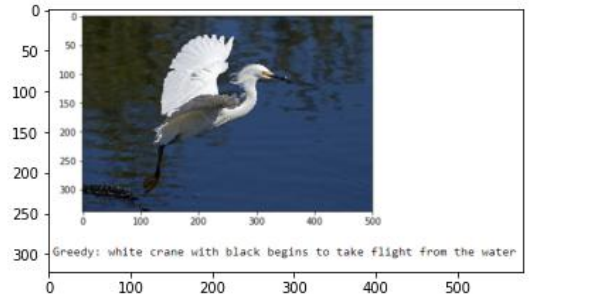
3. RESULTS

The model is trained for 36 epochs and it gives an accuracy of 0.8891 which is 88%. The training loss which indicates incorrect predictions for the model is 0.5948. The model is tested on testing images as well as some real-world images.

Each output is presented in English and converted into Hindi language. There's an audio associated with each that speaks out the result which is done using gTTS (Google Text to Speech) library as shown in the first output. The BLEU (Bilingual Evaluation Understudy) which is a metric to compare the output with the predicted one [8].For calculating the Bleu score we have used the NLTK library of python. For our model we have achieved the Bleu score of 0.5625.

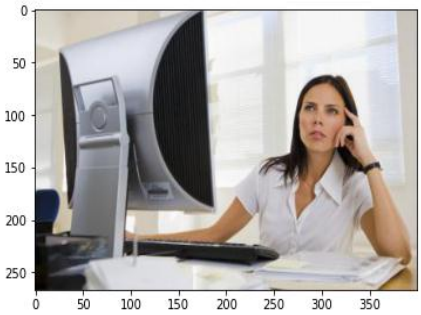
(a)

(b)



Predicted caption: white bird is flying through the air
Converted to Hindi: सफेद पक्षी हवा में उड़ रहा है

(c)



Predicted caption: woman in white shirt is sitting in front of table
Converted to Hindi: सफेद शर्ट में महिला टेबल के सामने बैठी है

(d)

Figure 10. Testing Model on Various Images

The output is expected to be short and precise which can be seen from the Fig.10 (a). Although the other kids seen in the image could also be mentioned. In the Fig.10 (b) & (c) shows correct identification of animal's .The grammar could be better like the bird flying "in" the air rather than "through" the air. The model recognizes the gender correctly and differentiates humans from animals too. The output of Fig.10 (d) could have been more accurate like "woman in white shirt is sitting in front of the computer on the table". Our model hence recognises humans and animals with gender specification with certain accuracy which can thereby be enhanced by increasing the training epochs.

Some outputs are displayed having moderately good accuracy although the model can be trained more to achieve better results. A website has been developed for users to upload an image and find the corresponding caption useful for different purposes especially for visually impaired people. Also, an API has been created so that this application can be reused.

3. CONCLUSIONS

This paper aimed to develop a model for automated image captioning using deep learning. To reach the goal of helping visually impaired people, this prototype is integrated with a text-to-speech model. In this paper, we have used the standard CNN-RNN model with an extension of GRU and natural language processing (NLP). Thus, we have implemented our approach of image captioning model and a detailed analysis of these approaches has been described here. Experimental results and analysis strongly demonstrate the effectiveness of the proposed model. In the future, we will continue with our work in other aspects of the model.

REFERENCES

[1] M. M. Tsung-Yi Lin, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll´ar and L. Zitnick, 2014, "Microsoft COCO: Common Objects in Context," Computer Vision – ECCV 2014, 8693.

[2] The COCO Dataset Website, Available online: http://cocodataset.org/#home.

[3] Bryan A. Plummer, Liwei Wang, Christopher M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik, 2017, "Flickr30K Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to Sentence Models, IJCV", 123(1), pp. 74-93.

[4] Adela Puscasiu, Alexandra Fanca, Dan-Ioan Gota, Honoriu Valean, 2020, "Automated image captioning", 2020 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR).

[5] Lin, Tsung-Yi, et al., 2014, "Microsoft coco: Common objects in context." European conference on computer vision. Springer, Cham.

[6] Md. Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga, 2018,"A Comprehensive Survey of Deep Learning for Image Captioning. ACM Comput. Surv", computing methodologies→Machine learning: Neural networks.

[7] DProgrammer Lopez, 2019, RNN, LSTM GRU, http://dprogrammer.org/rnn lstm-gru.

[8] Jason Brownlee, 2020,"A Gentle Introduction to Calculating the BLEU Score for Text in Python", Machine Learning Mastery, https://machinelearningmastery.com/calculate-bleu-score-for-text-python/.

[9] Yamashita, R., Nishio, M., Do, R.K.G. *et al, 2018,"* Convolutional neural networks: an overview and application in radiology", *Insights Imaging,* 9, pp. 611–629, https://doi.org/10.1007/s13244-018-0639-9.

[10] Kanagachidambaresan, G.R., Ruwali, A., Banerjee, D., Prakash, K.B, 2021, "Recurrent Neural Network", In: Prakash, K.B., Kanagachidambaresan, G.R. (Eds) Programming with TensorFlow, EAI/Springer

Innovations in Communication and Computing. Springer, Cham, https://doi.org/10.1007/978-3-030-57077-4_7.

[11] Sehgal, S., Sharma, J., & Chaudhary, N., *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, IEEE. https://ieeexplore.ieee.org/document/9197977.

[12] Van Houdt, G., Mosquera, C. & Nápoles, G, 2020, "A review on the long short-term memory model. *Artif Intell Rev* ", 53**,** pp. 5929–5955, https://doi.org/10.1007/s10462-020-09838-1