2019 Global Conference for Advancement in Technology (GCAT)
Bangalore, India. Oct 18-20, 2019

Deep Learning based Automatic Image Caption Generation

Varsha Kesavan
Electronics and Telecommunications
Fr. Conceicao Rodrigues Institute of Technology, Mumbai University
Sector-9A Vashi,Navi Mumbai, India
varz.kesavan@gmail.com

Vaidehi Muley
Electronics and Telecommunications
Fr. Conceicao Rodrigues Institute of Technology, Mumbai University
Sector-9A Vashi,Navi Mumbai,,India
vaidehimuley5@gmail.com

Megha Kolhekar
Electronics and Telecommunications
Fr. Conceicao Rodrigues Institute of Technology, Mumbai University
Sector-9A Vashi,Navi Mumbai, India
megha.kolhekar@fcrit.ac.in

***Abstract*— The paper aims at generating automated captions by learning the contents of the image. At present images are annotated with human intervention and it becomes nearly impossible task for huge commercial databases. The image database is given as input to a deep neural network (Convolutional Neural Network (CNN)) encoder for generating "thought vector" which extracts the features and nuances out of our image and RNN (Recurrent Neural Network) decoder is used to translate the features and objects given by our image to obtain sequential, meaningful description of the image. In this paper, we systematically analyze different deep neural network-based image caption generation approaches and pretrained models to conclude on the most efficient model with fine-tuning. The analyzed models contain both with and without 'attention' concept to optimize the caption generating ability of the model. All the models are trained on the same dataset for concrete comparison.**

Keywords*— *automated captions, deep neural network, CNN, RNN, feature extraction, attention.

I. INTRODUCTION

A large amount of information is stored in an image. Everyday huge image data is generated on social media and observatories. Deep learning can be used to automatically annotate these images, thus replacing the manual annotations done. This will greatly reduce the human error as well as the efforts by removing the need for human intervention The generation of captions from images has various practical benefits, ranging from aiding the visually impaired, to enabling the automatic, cost-saving labelling of the millions of images uploaded to the Internet every day, recommendations in editing applications, beneficial in virtual assistants, for indexing of images, for visually challenged people, for social media, and several other natural language processing applications. The field brings together state-of-the-art models in Natural Language Processing and Computer Vision, two of the major fields in Artificial Intelligence. One of the challenges is availability of large number of images with their associated text ever-expanding internet. However, most of this data is noisy and hence it cannot be directly used in image captioning model. For training an image caption generation model, a huge dataset with properly available annotated image is required. In this paper, we plan to demonstrate a system that generates contextual description about objects in images. Given an image, break it down to extract the different objects, actions, attributes and generate a meaningful sentence (caption/description) for the image.

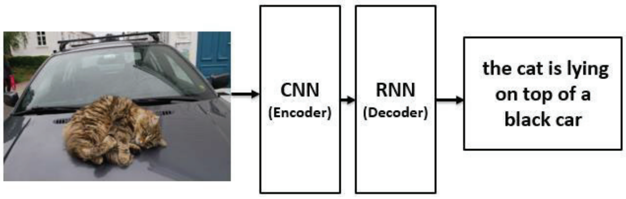

Fig. 1. Image captioning task

Generating captions automatically from images is a complex task as it entails the model to extract features from the images and then form a meaningful sentence from the available features. Basically, the feature extraction is done by training a CNN (Convolutional Neural Network) with huge number of images and the correct weights are identified by multiple forward and backward iterations. With the help of RNN (Recurrent Neural Network) and the extracted features, a sentence is generated. Figure 2 shows the block diagram.

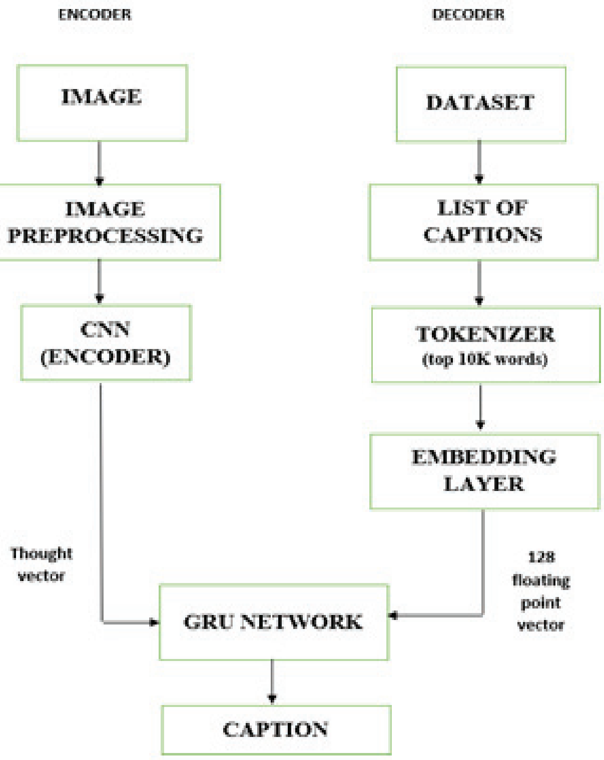

Fig. 2. Block Diagram without attention

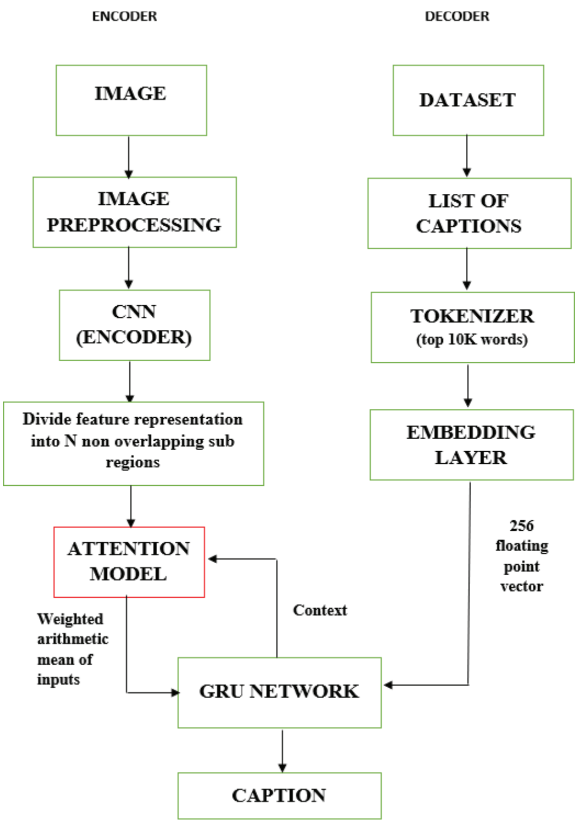

Fig. 3. Block Diagram with attention model

II. RELATED WORK

To start with automatic image caption generation, image annotation was studied from Image Annotation via deep neural network [1] which proposes a novel framework of multimodal deep learning where the convolutional neural networks (CNN) with unlabeled data is utilized to pre-train the multimodal deep neural network to learn intermediate representations and provide a good initialization for the network then use backpropagation to optimize the distance metric functions on individual modality. This was followed by Automatic image annotation using DL representation [2] in which the last layer of CaffeNet of the CNN based model is replaced with a projection layer to perform regression and the resulting network is trained for mapping images to semantically meaningful word embedding vectors. Advantage of this modelling is: firstly, it does not require dozens of handcrafted features and secondly, the approach is simpler to formulate than any other generative or discriminative models. A single network is created for generating captions of images in Show and Tell: A Neural Image Caption Generator [3]. In this network, deep convolutional network is used for image classification and sentence generation is done by a powerful Recurrent Neural Network which is trained with the visual input so that RNN can keep track of the objects explained by the text. A different approach to caption generation is incorporated in Show, Attend and Tell: Neural Image Caption Generation with Visual Attention [4] where, a form of attention, "hard" attention mechanism and "soft" attention mechanism are described. A deterministic "soft" attention mechanism is employed by standard back-propagation methods and a stochastic "hard" attention mechanism by maximizing an approximate variational lower bound.

III. METHODOLOGY

In this paper we propose a transfer learning approach to generate automated captions for any given image. In this model the encoder used is pre-trained VGG16 model. This model makes use of a recurrent neural network which encodes the variable length input into a fixed dimensional vector and uses this representation to "decode" it to the desired output sentence. The vector containing the output of the fully connected layer in VGG16 is connected to GRU units is called "thought" vector.

A. Encoder

In the encoder of the system, pre-trained models VGG 16, RESNET and Inception are used to compare between the results obtained from each of them. The encoder is used to extract thought vector of the image which describes the contents of the image. The size of the output of last fully connected layer is (none,4096). The embedding dimension of decoder is 512, thus the thought vector is passed through a dense map function to downsize the thought vector from 4096 to 512. This thought vector is given as initial state to the GRU units in each layer. The dense map function follows the tanh function. Tanh function is rescaling of logistic sigmoid function so that the output range is from -1 to 1. The logistic sigmoid function is given as:

$$g(x) = \frac{e^x}{1 + e^x}$$

Tanh function is given as:

$$\tanh(x) = 2g(2x) - 1$$

Pretrained VGG model: The softmax layer of the VGG model is striped to avoid classification of the image and instead the information about the entire image is obtained. The hidden layers of VGG model consists of 2 convolutional layers followed by max pooling layer to reduce the size by half. This architecture is repeated 3 times followed by flatten layer to get a one-dimensional output which is fed to fully connected layers. The output of second fully connected layer is taken as the initial state of GRU layers in the decoder after it is downsized by the dense map layer. The summary of VGG model is shown in Figure 5.

B. Decoder

The decoder consists of Tokenizer, embedding layer, GRU layers and dense layer. Each of the captions are first prepended and appended by start and end marker respectively. Tokenizer layer is used to convert the first predefined number of unique words into integer tokens. Once the tokens are assigned, the embedding layer converts integer-tokens into vectors of 128 floating-point number since the RNN network works on vectors and not integers. The sequence vectors are padded to ensure that all the sequence vectors are of same length which is equal to the length of the maximum sequence. The GRU unit comprises of three gates: forget gate, output gate, input gate. The gates are defined as:

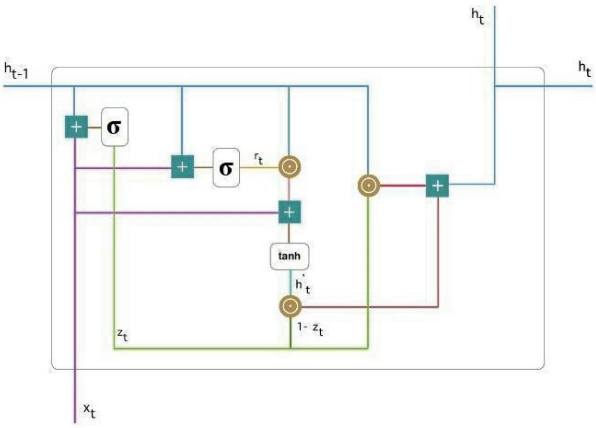

Fig. 4. GRU architectures

Source: http://colah/posts/2015-08-Understanding-GRUs/

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h_t$$
$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1})$$
$$r_t = \sigma (W^{(r)}x_t + U^{(r)}h_{t-1})$$
$$h_t = tanh(Wx_t + r_t \odot U\, h_{t-1})$$

The output of the GRU network is further given to the dense layer to convert the sequences into integer tokens which generates output after passing through token_to_word instance of tokenizer object. The steps per epoch is calculated as:

$$\text{steps per epoch} = \frac{\text{totalnumberofcaptionsinthetrainingdataset}}{\text{Batch size}}$$

Loss is computed using sparse softmax cross entropy with logits which measures the probability error in discrete classification tasks in which the classes are mutually exclusive (each entry is in exactly one class). The optimizer used is RMSprop instead of Adam optimizer for better results. The activation used in the last dense layer is linear activation.

Training: The model is trained using MS-COCO dataset which contains 5 captions for each image in the training dataset. The size of the training dataset is 118287 images. The RNN model is trained by shifting the training caption dataset by one unit to train the model to correlate the different words with each other for better prediction.

B. Image Captioning With Attention Model

In the previous block diagram (Figure 2), as explained, the approach was to encode image into vector representation using CNN encoder and decode into word vectors signifying captions using RNN. The problem with this approach is that while generating a single caption, GRU every time looks at the entire image vector representation and this seems inefficient as different words in a caption are generated by looking at specific parts of the image.

As a solution to this problem, [4] introduced the concept of 'attention' into image captioning. Figure 3 shows the block diagram of Image captioning with Attention. The idea was to view the image from the perceptive of captioner to decide what is important and what is not important when it comes to captioning an image or in other words, trying to decide what details are worth paying 'attention' to. The paper discussed two types of attention: 1. Hard attention 2. Soft attention.

Attention mechanism starts by first creating 'N' different non overlapping sub regions of the feature vector representation which is the output of CNN encoder. Attention unit takes all the sub regions (say Y_i, i=1,2,… N) and context (C) as input and outputs weighted arithmetic means of these regions. Context C is the collection of recent outputs of RNN. The inputs Y_i and C are applied to weights that form the learnable parameters of this unit. 'tanh' activation is applied to it so that very high values tend to have very small differences (close to 1) and very low values also have very small differences (close to -1). Let the output of 'tanh' activation function be 'm_i',

$$m_i = tanh(Y_i W_{Yi} + CW_C)$$

Basically, regions that come as output should be relevant to the context. Instead of 'tanh' function, inner dot product between regions Y_i and context C can also be calculated. More the similarity, higher the value of dot product and output will weigh more relevant regions.

$$m_i = Y_i . C$$

Although 'tanh' will give less choppy and more smoother parts of sub region over the inner dot product approach. These values of m's are passed through softmax function which outputs probabilities 'S'. Inner product of 'S' with 'Y_i' will give relevant regions of the entire image. Further, the output of the attention model is given to GRU network for caption generation just as explained for block diagram Fig.3.1.

In this paper, the CNN models tested with attention model are InceptionV3 and Inception-ResnetV2. Dataset used is MS-COCO. 30000 images were used for training. The model is trained with 3 different values (epoch=2,5,10) to compare results.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312

Fig. 5. VGG model summary

IV. RESULTS AND ANALYSIS

A. *Results Of Image Captioning Without Attention Model*

Using VGG16 model as an encoder with 16 hidden layers and GRU network (using 3 GRU layers) as decoder with number of epochs set to 20 for optimum performance taking 118287 images from MS-COCO dataset as training dataset and vocabulary size of 10000 unique words, Figure 6 shows prediction for an image in validation set. Figure 8 shows the image and the predicted caption for an image from testing data set.

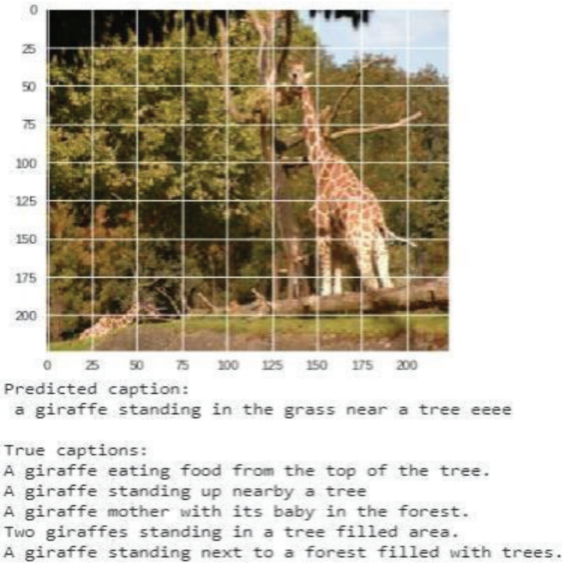

Fig. 6. Prediction for image in validation set

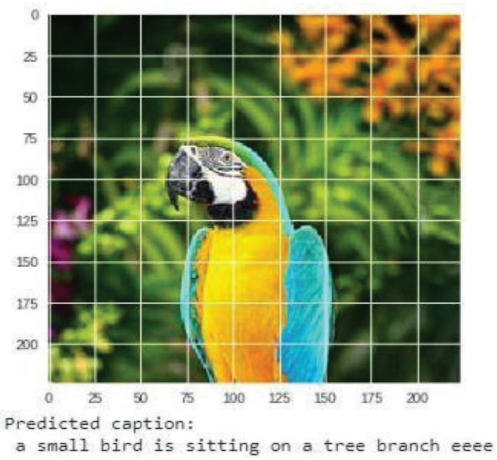

Fig. 7. Prediction for random image

B. *Results of Image Captioning with Attention Model*

Using InceptionV3 model as an encoder and GRU network (using 3 GRU layers) as decoder with number of epochs set to 2 for optimum performance taking 30000 images from MS-COCO dataset as training dataset and vocabulary size of 10000 unique words, Figure 8 to Figure 11 outputs are observed. Average BLEU [5] score observed is 0.7912.

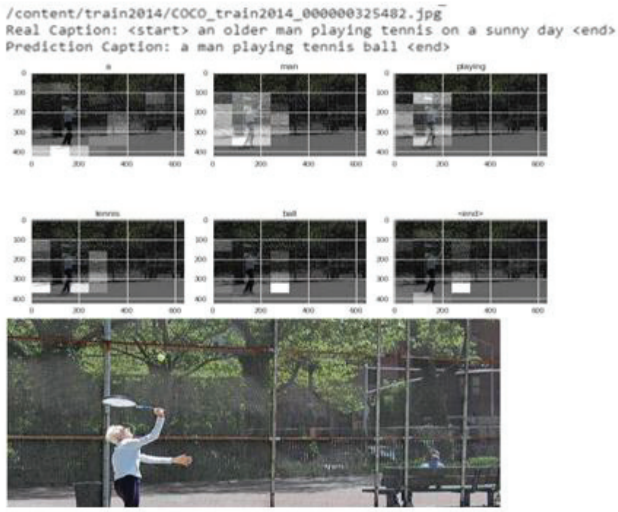

Fig. 8. Prediction for image in validation set

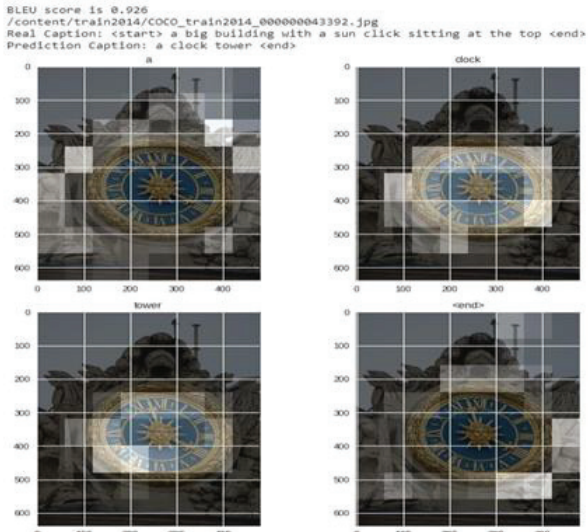

Fig. 9. Prediction for image with BLEU[5] score 0.926

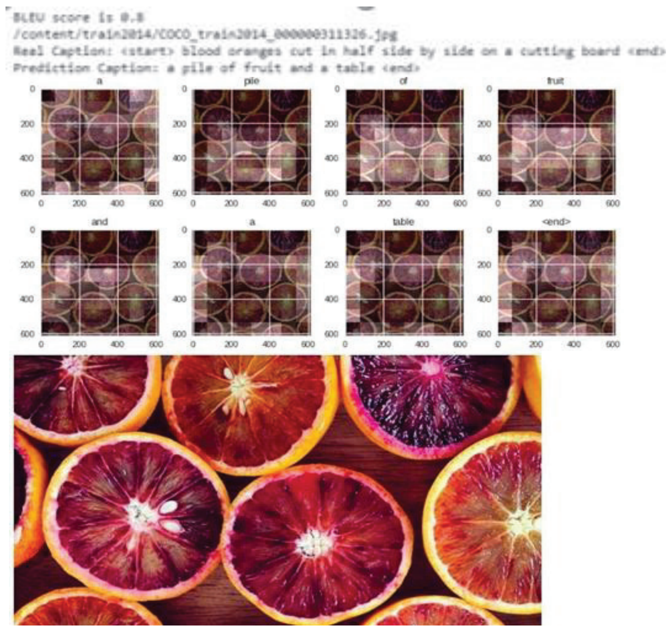

Fig. 10. Prediction for image with BLEU score 0.8

Fig. 11: Prediction for image with BLEU score 0.3

Using Inception-ResNetV2 model as an encoder and GRU network (using 3 GRU layers) as decoder with number of epochs set to 2 for optimum performance taking 30000 images from MS-COCO dataset as training dataset and vocabulary size of 10000 unique words, Figure 8 to Figure 11 outputs are observed. Average BLEU score observed is 0.765.

Image 1:

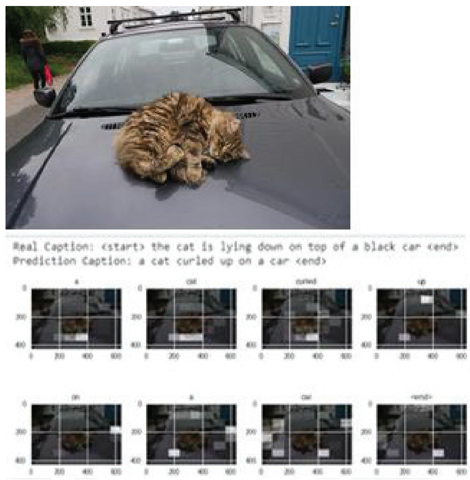

Image 2:

Fig. 12. Prediction for images 1,2 from validation set using Inception ResNetV2

Fig. 13. Prediction for a random image

(Source: i.ytimg.com/vi/4Hpzw9ImGSI/maxresdefault.jpg)

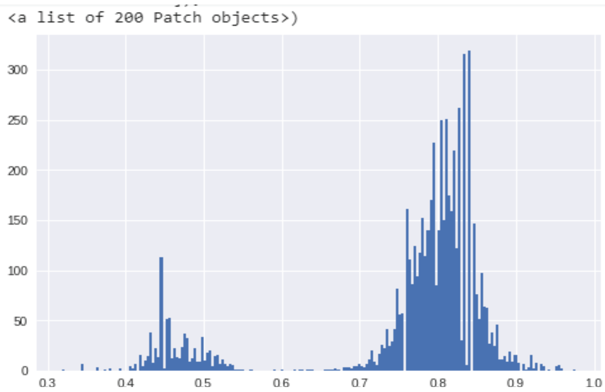

Fig. 14. Histogram of unigram BLEU score for images in validation set in Inception V3 model

Thus, majority of images in validation dataset have BLEU score above 0.7 for unigram BLEU. The table 1 summaries the observation and analysis of BLEU scores.

TABLE I

BLEU Score	Nature of predicted sentence
Above 0.9	1. At least one word match between true caption and predicted caption 2. Number of words in predicted caption is less 3. Overall relevance: moderate
0.8 - 0.9	1. One word match 2. Number of words in predicted caption is relatively more 3. Overall relevance: High
0.3-0.5	1. No word match 2. Number of words in predicted sentence is the highest. 3. Single word is repeated multiple times 4. Overall relevance: Low

V. CONCLUSION

We have presented an end-to-end neural network system that can automatically view an image and generate a reasonable description in plain English. It is based on a convolution neural network that encodes an image into a compact representation, followed by a recurrent neural network that generates a corresponding sentence. The model is trained to maximize the likelihood of the sentence given the image. We also saw the effect of the encoder-decoder approach combined with attention and made analysis.

The following conclusions were drawn:

1. The number of epochs required for the same dataset varies for different models.
2. As the network becomes deeper, the number of epochs for “Image captioning” problem becomes less.
3. There is a trade-off between time required for execution and the number of hidden layers.
4. The measurement of average value of different metrics on the same dataset with different models and different number of epochs was done.
5. The precision with which different metrics evaluate performance of system as compared to the human generated captions was judged.

REFERENCE

[1] Sun Chengjian, Songhao Zhu, Zhe Shi, “Image Annotation Via Deep Neural Network”, Published in: 2015 14th IAPR International Conference on Machine Vision Applications (MVA).

[2] Venkatesh N. Murthy, Subhransu Maji, R Manmatha, “Automatic Image Annotation using Deep learning representations”, ICMR '15 Proceedings of the 5th ACM on International Conference on Multimedia Retrieva.

[3] Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan, “Show and Tell: A Neural Image Caption Generator”, published 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

[4] Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, Yoshua Benjio, “Show, attend and tell: neural image caption generation with visual attention”, ICML’15 Proceedings of the 32nd International Conference on Machine Learning – Volume 37, Pages 2048-2057.

[5] Kishore Papineni, Salim Roukos, Todd Ward, Wei-Jing Zhu, “BLEU: a Method for Automatic Evaluation of Machine Translation”, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 311-318.

[6] www.deeplearning.ai

[7] www.tensorflow.org