Encoder-Decoder Architecture for Image Caption Generation

Harshit Parikh
Dept. of EXTC
SFIT, Borivali
Mumbai, India
harshitparikh12@gmail.com

Harsh Sawant
Dept. of EXTC
SFIT, Borivali
Mumbai, India
thesawantharsh10@gmail.com

Bhautik Parmar
Dept. of EXTC
SFIT, Borivali
Mumbai, India
bhautikparmar98@gmail.com

Rahul Shah
Dept. of EXTC
SFIT, Borivali
Mumbai, India
rshah4297.rs@gmail.com

Santosh Chapaneri
Dept. of EXTC
SFIT, Borivali
Mumbai, India
santoshchapaneri@sfit.ac.in

Deepak Jayaswal
Dept. of EXTC
SFIT, Borivali
Mumbai, India
djjayaswal@sfit.ac.in

***Abstract*—Describing the contents of an image without human intervention is a complex task. Computer Vision and Natural Language Processing are widely used for tackling this problem. It requires an approach with two distinct methods, to understand the contents of the image using computer vision, convert the understanding into semantically correct sentences. Convolutional Neural Network (CNN) is a widely used powerful image feature extraction algorithm for object detection and image classification. Gated Recurrent Unit (GRU) is typically used for effective sentence generation. A combined model of CNN and GRU was proposed to achieve accurate image captions. With the proposed model, an experimentation was done with various datasets and compared the results with existing work. BLEU evaluation metrics was used for benchmarking the results; The proposed model results in a BLEU-4 score (the higher the better) on the MS-COCO 2017 dataset as 53.5.**

***Keywords*—Artificial Intelligence, Computer Vision, Natural Language Processing**

I. Introduction

Humans can easily describe their observable environment. For a given image, it is natural for humans to describe an immense amount of details about the image contents. Giving the ability of humans (ability of describing an image) to computers is a major task for researchers in the field of artificial intelligence. The goal is to detect the different objects captured in the image (object recognition) and identify the relations between various objects. Convolution Neural Network has proved to be a powerful image recognition tool over the years. CNN was used for its impressive feature extraction from the given image. A pre-trained model InceptionResnetv2 model [1] was used for classifying images.

Next, Natural Language Processing (NLP) was used for sentence (caption) generation since language is the basis of human communication. Short-term memory is the main problem for Recurrent Neural Networks (RNNs) [2]. To avoid the problem of the vanishing gradient, Gated Recurrent Unit was used. The output of CNN was used and passed on as an input to the GRU for caption generation. The GRU gives 1536 elements as output, which are then condensed to 1024 units and finally 512 units. This is then passed to 3 GRU layers for the state-of-the-art caption generation.

The proposed model takes an image as the input and passes it to the CNN model, which gives the feature vector of the image; the feature vector is passed to the GRU layer, which generates the caption of the image. In this work, a CNN layer is needed to recognize objects from the image and a RNN layer generates the sentences. To decrease the chances of overfitting when trained on deep convolution layers, He, *et al*. in [3] introduced Residual connections, which were proven to be very important when training deeply. Since residual networks are ideal and work powerful for very deep convolutional layers, the InceptionResNetv2 model was used, which was developed by the researchers at Google in 2016 and was made as an improvement over Inceptionv4. It is a robust and powerful model for image classification, which is a hybrid of both the Inception model made by Google and the ResNet model by Microsoft. The inception blocks were simplified and the number of layers in the network were increased. This model has achieved the highest accuracy across the board, better than its predecessors. This is the finest feature extraction tool for image classification as of today. This model was used for feature extraction and classification. For our particular application, the last layer Softmax was removed and directed the output as an initial state to the GRU layer through a Dense layer. [4]. Using this model, excellent results were obtained as observed by the evaluation metric scores.

II. Related Work

A great deal of research has been carried out to carry out the task of generating captions to images. Bernardi, *et al*. [5] in their survey categorized it into two different approaches. First is the use of pre-defined templates for descriptions as given by Yang *et al*. [6]. The drawback of the completion of templates is that it fails to generate novel and unique captions. The second approach is the use of Natural Language Processing specifically RNN and LSTM as done by Vinyals *et al*. [7] who proposed an end-to-end model called NIC (Neural Image Caption) comprising CNN for image feature extraction and RNN for caption generation. This generates more novel and accurate captions. Several researchers used the combination of CNN and RNN. This approach eliminates the drawbacks of

978-1-7281-2339-4/20/$31.00 ©2020 IEEE

the templates based captioning. Natural Language Processing using standard RNN faced some problems. Short-term memory of RNNs was not viable for generating captions as sentence formation depends on the previous knowledge of words used. The subsequent flow of words depends upon the past input words. Thus RNNs became redundant for generation of captions. Another drawback faced by RNNs was the Vanishing Gradient Problem.

Hossain *et al*. [8] used the method of a CNN and RNN based combined network. It uses CNN for extracting a region-based image feature vector. The output generates captions for every region, which gives a more detailed description of an image but lacks in certain areas. As the regions are dense, there may be overlapping of different regions, and it is challenging to recognise the target and position with this approach. Using LSTM and GRU dealt these problems as LSTM and GRUs use "gates" for controlling the gradient descent and allow the model to learn correctly. LSTM is widely used for its memory retaining capabilities. It solved the vanishing gradient problem and gave more robust descriptions. The subsequent advances in the Natural Language Processing approach were the use of GRU. GRU has two gates as compared to three of the LSTM. This greatly reduces the amount of computation required while still maintaining the overall performance of the model. Evaluation of both LSTM and GRU was done to compare the results in our case.

III. Implementation

A. Datasets

1) **MS COCO (Microsoft Common Objects in Context)**: This dataset [9] has a wide variety of images with more than 90 different object types. This dataset is updated annually and the number of images is also increased with each subsequent updates. The latest dataset version consists of over 1,18,000 images for training, 5,000 for validation and 41,000 test images.

2) **FLICKR8K**: This crowdsourced dataset [10] consists of over 8,000 images, where each image is paired with five different captions. The dataset is separated in to three parts, roughly 6,000 for training, 1,000 for validation and 1,000 for testing. This dataset is slightly outdated compared to other datasets and has less number of images to suit a wider range of applications.

3) **FLICKR30K**: This dataset [11] contains over 30,000 images and is an updated version of Flickr8k. This dataset contains more wide variety of images and also has the same five unique captions per image pattern as with Flickr8k. It is also separated in 3 parts; 28,000 for training, 1,000 for validation and 1,000 for testing. This dataset is mainly used for image captioning and related applications.

TABLE I: Datasets used in this work

Dataset	Number of Images		
Name	Train	Validation	Test
MS COCO	1,18,287	5,000	40,670
Flickr30k	28,000	1,000	1,000
Flickr8k	6,000	1,000	1,000

For this application, MS COCO dataset was used which has also become the standard for image captioning. Conversion of all the sentences to lower case was done and non-alphanumeric characters were discarded.

B. Model

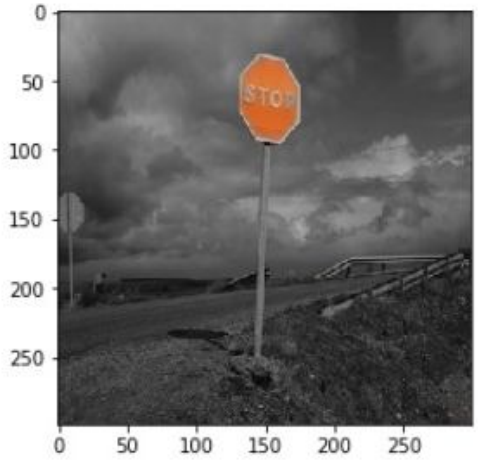


Fig. 1: Sample image

Consider a sample image shown in Fig, 1 with the reference caption: "stop sign on road". The encoder is a CNN network that comprises a pre-trained InceptionResnetv2 model, which is 467 layers deep and the dropout value for this model is 0.8. First, the input image was resized to 299×299 pixels. Next, the image was split into respective RGB planes to create a three-channel image stack and the resulting image is fed to the network. The output of the last layer before the Softmax function was extracted. Functions were made to process all the images within the dataset using the pre-trained image-model and saving the transfer-values in a pickle file so that they can be reloaded quickly. This provides u s a vector with 1536 elements that summarizes the image contents, almost like how a "thought-vector" summarizes the contents of an input-text. This vector was used as the initial state of the Gated Recurrent Units (GRU). However, the interior state-size of the GRU is merely 512 elements, so an intermediate fully-connected (dense) layer was required to map the vector with 1536 elements down to a vector with 1024 elements and finally down to 512 elements.

A replacement dataset of the transfer-values was created. This is because it takes an extended amount of time to process a picture within the InceptionResnetv2 model. All parameters of the pre-trained model were not changed, so whenever it processed an image, it gave precisely the same result. The transfer-values were needed to coach the image-captioning model for several epochs, thus calculations of the transfer-values was done once and saved them in a pickle file for further computation. This finally gives 1536 output units, which is then followed by a Dense layer of 1024 units and finally through a 512 units Dense layer, which is passed to the three GRU layers.

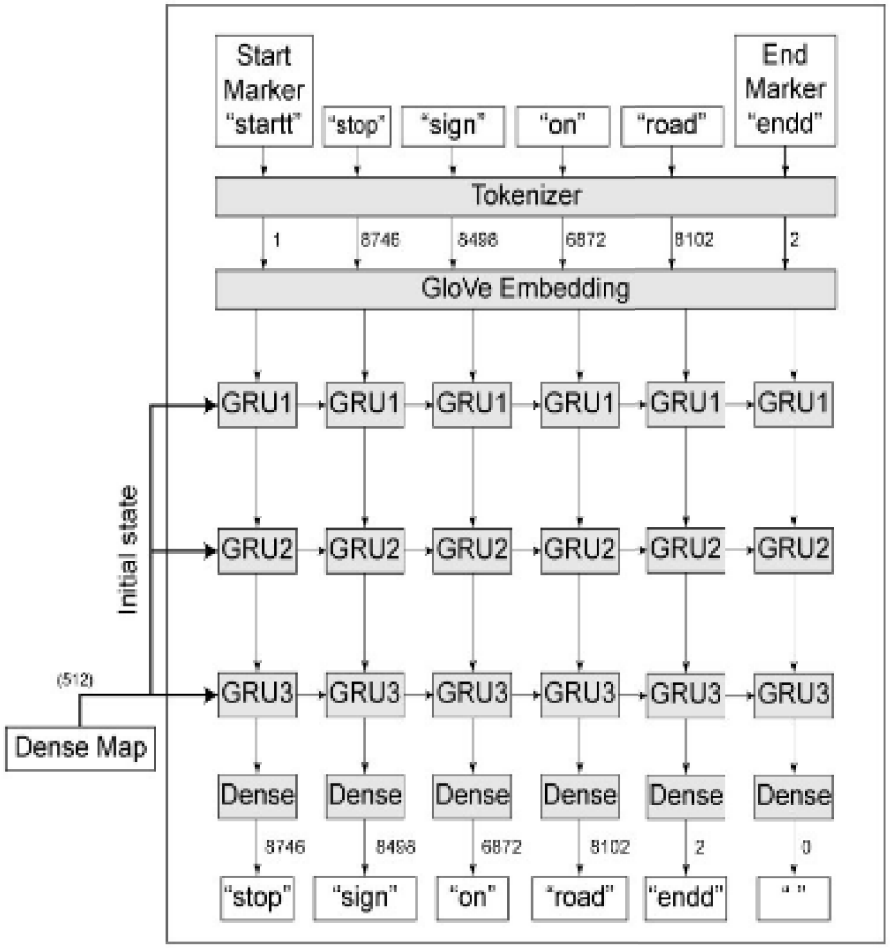


Fig. 2: Decoder Network of our model [12]

GRU uses two gates called update gate and reset gate. When x_t is plugged as an input into the network unit, it is multiplied by its own weight W^z. An equivalent goes for h_{t-1} which keeps the knowledge and data for the previous $(t-1)$ units and it is multiplied by its own weight U^z. The update gate helps the model to work out what proportion of past information (from previous time steps) must be passed along to the longer term. The sigmoid activation function squashes the obtained values between 0 and 1. The results for z_t and r_t are added together and a tanh activation function is applied which squashes the values between -1 and 1.

$$z_t = \sigma(W^z x_t + U^{(z)} h_{t-1} + b_z) \tag{1}$$

$$r_t = \sigma(W^{(r)} x_t + U^{(r)} h_{t-1} + b_r) \tag{2}$$

$$h_t = tanh(W x_t + r_t \odot U h_{t-1}) \tag{3}$$

The decoder of the proposed model shown in Fig. 2 uses the above-mentioned initial state alongside a start-marker "startt", to differentiate it from the word "start" which may create a few complexities, to produce output words. With the first iteration, it will output the word "stop". Then, the decoder will get this word as input and it will generate the word "sign" out, and so on. Finally, it generates the text "stop sign on road endd" where "endd" marks the top of the text.

For word embedding GloVe is used which was introduced by J. Pennington, *et al* [13] as it outperforms the previous work done by other authors in this field. GloVe word embedding are trained on 6 billion tokens, 400k vocabulary and different dimensionality starting from 50D to 300D vectors. The 300 dimensionality vector that results in mapping the index or the tokenized value of each word to a 300 dimensional vector. In our model, the decoder consists of three layers of GRU having a state size of 512. To initialize the interior states of GRU units, the feature vector of an image was used that had a size of 1536 obtained through the last layer of CNN. Since the dimensions of the internal units of GRU are 512, a fully connected dense layer is employed to map the vector from 1536 to 1024 and then to 512. Here, a tanh activation function was used to limit the output values between -1 and 1. Hence these three layers of GRU return the sequences because eventually sequence of tokens will be generated and converted into the text.

Fifteen thousand most recurrent words were chosen from the captions of the training dataset which consists of around 1,18,000 images with each having 5 captions except some images with less than or more than 5 captions. For each word, there is a 300 dimensional vector associated to ensure the correlation between each word.

The loss function used is the sparse categorical cross entropy; it is preferred over the categorical cross entropy. The latter is used when the output targets are one-hot encoded

such as [1,0,0] or [0,1,0] or [0,0,1] and the former is used when the output targets are integers such as 1 or 2 or 3. The output gained is the tokenized value of each word which is an integer. The optimizer used is the Adam optimizer which gives a loss of 0.58 after saving the best model when trained for 100 epochs. RMSprop optimizer gave equivalent results, but the BLEU score was 51.5 and some captions were not up to the mark. Hence, the ADAM optimizer was used which gives better results. Other optimizers that were tried on were Nadam, SGD, Adagrad, Adadelta and Adamax, but none were better than the Adam or RMSprop optimizer.

IV. Results

A. Evaluation Metrics

Our model generates captions in the form of sentences. However, language is a subjective topic and the sentence formation based on the description diversifies from person to person. Hence, the correct computation of the accuracy of the model becomes difficult. Some of the evaluation metrics used by our model were as follows:

*1) **BLEU (Bilingual Evaluation Understudy Score)**:* BLEU is an algorithm that checks the quality of text generated at the output by the model. It is used for checking the model-generated output and comparing the text to one or more reference captions. The output of BLEU is always between 0 and 1. Value closer to 1 shows that the text is more analogous to the human-generated caption.

In general, the formula for precision(P) can be given as:

$$P_1 = \frac{\sum_{unigram\ \epsilon\ \hat{y}} count_{\text{clip}}(unigram)}{\sum_{unigram\ \epsilon\ \hat{y}} count(unigram)} \tag{4}$$

However, using individual words as the unit of comparison is not optimal. Instead, BLEU computes the modified precision metric using n-grams. The length which has the "highest correlation with monolingual human judgements was found to be four (i.e. P range from (1 to 4)). The precision for n-gram can be given as:

$$P_n = \frac{\sum_{ngram\ \epsilon\ \hat{y}} count_{\text{clip}}(ngram)}{\sum_{ngram\ \epsilon\ \hat{y}} count(ngram)} \tag{5}$$

Since the length of the generated caption matters, the length is essential for computing BLEU score. Effect of length can be determined by the Brevity Penalty factor (BP), which compares the length of generated captions with the closest length in the reference sentences. The concept of Brevity Penalty is given in more detail by Kishore *et al.* [14]

$$BP = \begin{cases} 1, & \text{if } c > r \\ e^{(1-r/c)}, & \text{if } c \leq r \end{cases} \tag{6}$$

where, c = length of generated caption and r = reference effective corpus length.

Hence,

$$BLEU = BP * \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{7}$$

*2) **METEOR**:* METEOR Score by S. Banerjee, *et al.* [15] is an evaluation metric, which is better than the BLEU evaluation metrics since it has unique features which are not found in BLEU. It uses the combination of unigram-precision, unigram-recall to give the results. It also considers the stemming, metonym and synonym matching along with standard exact words in the caption.

To obtain Meteor score, we calculate the precision and recall, which are given as

$$P = \frac{m}{w_{\text{t}}} \tag{8}$$

$$R = \frac{m}{w_{\text{r}}} \tag{9}$$

where, m is the number of unigrams in the generated caption that are also found in the reference captions, w_{t} is the number of unigrams in the candidate translation and w_{r} is the number of unigrams in the reference captions.

Then, the Eq. 8 and Eq. 9 are combined using the harmonic mean to get the Eq. 10

$$F_{\text{mean}} = \frac{10PR}{R + 9P} \tag{10}$$

This measure considers unigram which takes a single word into consideration but not n-gram segments that appear in both the reference and the generated captions.

So, there is a need to compute longer n-gram matches and this can be done by computing penalty.

$$p = 0.5 \times \left(\frac{c}{u_{\text{m}}}\right)^3 \tag{11}$$

where, c is the number of chunks of characters and u_{m} is the number of unigrams that have been mapped.

Finally, now the calculation of Meteor score is done using Eq. 10 and Eq. 11 as follows:

$$M = F_{\text{mean}} \times (1 - p) \tag{12}$$

For both the evaluation metrics, higher the score better the caption is.

B. Experimental Results

The training of the model was done on a variety of images from the MS COCO dataset. The model was tested by test images from the same dataset as well as some images from other sources. Although the results were very good, as shown in Fig. 3, there were a few exceptions where mediocre captions were observed. In general, standard scenes and objects are easily identified and are accurately captioned, while uncommon or rare objects and scenes are misinterpreted by the model. This problem is mitigated by the use of an extensive dataset with vivid object types. We observed that our model benefits from the number of images it gets trained on and the variety of objects in it. One of the limitation of our model is that any of the datasets: Flickr8k, Flickr30, MS COCO doesn't have classes or object for each image, hence we cannot train the CNN layer and we have to use it as it is for our model.

Our model gives BLEU-1 score as 73.8, BLEU-2 score as 65.1, BLEU-3 score as 59.2, BLEU-4 score as 53.5 and

METEOR score as 37.5 on the MS-COCO 2017 Dataset when using three GRU layers.

A comparison was done of this result with that of the result when 3 LSTM layers are used and the results were as follows:

TABLE II: Evaluation Metrics which describes the performance of the model.

Parameters	Evaluation Metrics				
	BLEU-4	BLEU-3	BLEU-2	BLEU-1	METEOR
3 GRU layers	53.5	60.2	67.1	74.8	37.5
3 LSTM layers	50.4	56.8	62.8	70.4	36.4

Some of the previous work is compared to our model as shown in the table III below.

TABLE III: Comparison of scores of previous related work.

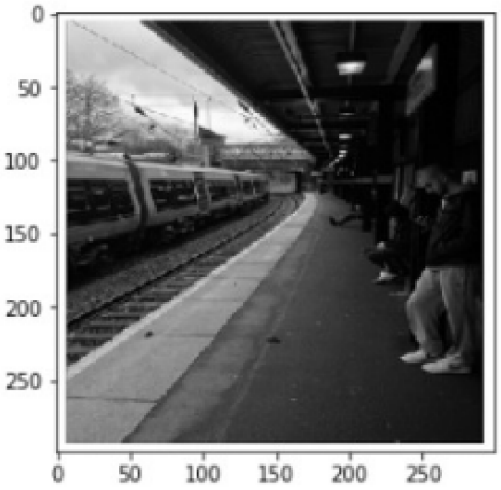
Sr. no.	Model	BLEU-4	METEOR
1	O. Vinyals *et al.* [7]	27.7	23.7
2	M. Hossain *et al.* [8]	33.2	22.6
3	M. Tanti *et al.* [18]	17.0	15.8
4	K.simonym *et al.* [19]	24.6	20.41
5	Moses Soh *et al.* [20]	24.4	21.7
6	Haoran Wang *et al.* [21]	24.3	23.9
7	Karapathy *et al.* [22]	23.0	19.5
8	Xihui Liu *et al.* [23]	35.8	27.4
9	Our Model	53.5	37.5

We can observe that there are no improvements as such in the BLEU score when using the LSTM layer instead of a GRU layer. Here, the model predicts captions on the test images that were not provided as input to the image. Some results of the generated captions are below shown in Fig. 3

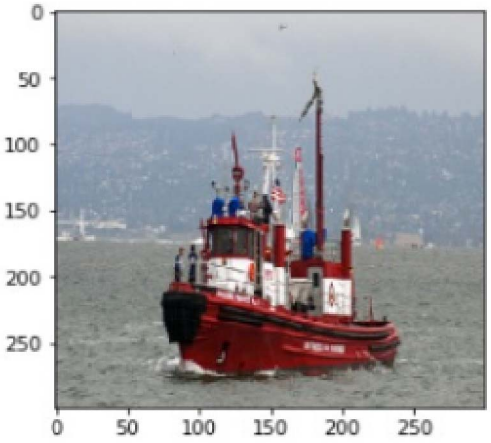
However, there were some instances where the captions were irrelevant as shown in Fig. 4.

V. Conclusion

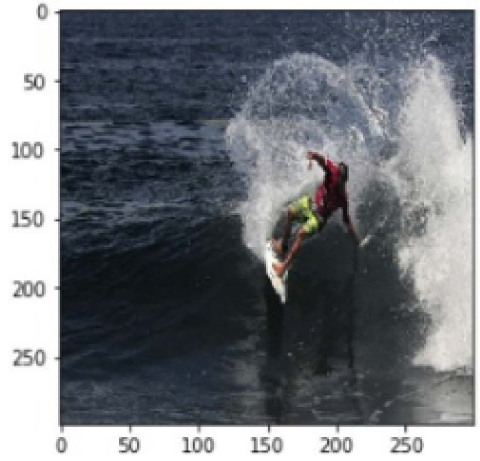
A neural network model was presented that can understand an image, process it and give the description in the English language. It is based on a Convolutional Neural Network, which is used as an encoder to process the images and then relevant captions are generated using the Gated Recurrent Unit. This unit has less number of gates and hence requires less time to train and also gives suitable descriptions for an image. The model is trained to maximise the likelihood of the sentence given the image. The ranking metrics used for evaluation purpose were the BLEU-4 score, which was 53.5 and METEOR score which was found to be 37.5 when trained on MS-COCO Dataset. Our model benefits from the use of a large dataset, as reflected from the results above. The results can be further improved by using extensive datasets with several object types. For further work, some of the other evaluation metrics can be used such as ROUGE [16] and CIDEr [17] for evaluation along with the BLEU and the METEOR score. This can be used to implement a blind stick that can be used as a guidance for the visually impaired person by enabling text to speech conversion. We can also implement real time captioning for videos that can be used for CCTV Camera surveillance. Using captioning for videos, we can also provide real time information for visually impaired people, since a particular image can only give the caption at one specific time.



(a) A black and white photo of a train on a track.



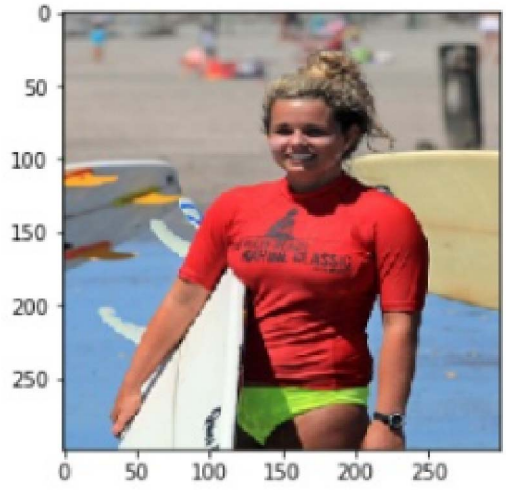
(b) A large boat in a large body of water.



(c) A man riding a wave on top of a surfboard.

Fig. 3: Some of the best captions predicted by our model.

(a) A group of people standing in line on a snowy hill.



(b) A man in red shirt is holding a yellow frisbee.

Fig. 4: Some predicted captions which were not accurate.

REFERENCES

[1] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, "Inception-v4, InceptionResNet and the Impact of Residual Connections on Learning," *arXiv:1602.07261v2 [cs.CV]*, 2016.

[2] X. Chen, C. Zitnick, "Learning a Recurrent Visual Representation for Image Caption Generation," in *arXiv:1411.5654*, 2014.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition" in *IEEE CVPR 2016 proceedings, IEEE, pp 770–778*, 2015.

[4] S. Tsutsui and D. Crandall, "Using Artificial Tokens to Control Languages for Multilingual Image Caption Generation,", *arXiv:1706.06275 [cs.CV]*, 2017.

[5] R. Bernardi, R. Cakici, D. Elliott, A. Erdem, E. Erdem, N. Ikizler-Cinbis, F. Keller, A. Muscat, and B. Plank, "Automatic description generation from images: A survey of models, datasets, and evaluation measures," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017*, pp. 4970–4974, 2017.

[6] Y. Yang, C. Teo, L., Daume and Y. Aloimonos, "Corpus-guided sentence generation of natural images," in *Conference on Empirical Methods in Natural Language Processing*, 2011.

[7] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3156–3164, 2015.

[8] M. Hossain, F.Sohel, M.Shirattuddin and H.Laga, "A Comprehensive Survey of Deep Learning for Image Captioning", in *ACM Computing Surveys*, 2019.

[9] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. Lawrence, "Microsoft COCO: Common Objects in Context," in *European Conference on Computer Vision*, 2014.

[10] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier, "Collecting image annotations using amazon's mechanical turk," in *NAACL HLT Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pp. 139–147, 2010.

[11] B. Plummer, L. Wang, C. Cervantes, J. Caicedo, J. Hockenmaier, and S. Lazebnik, "Flickr30k entities: Collecting region-to-phrase correspondences for richer image-tosentence models," in *Proceedings of the IEEE International Conference on Computer Vision* (ICCV), 2015.

[12] M. Pederson, *TensorFlow-Tutorials*, Feb 3, 2019. [Online]. Available: https://github.com/Hvass-Labs/TensorFlow-Tutorials/blob/master/22_Image_Captioning.ipynb [Accessed: Nov 24, 2019.]

[13] J. Pennington, R. Socher, C. Manning, "GloVe: Global Vectors for Word Representation", in *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

[14] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method forautomatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics.* Association for Computational Linguistics, pp.311–318, 2002.

[15] S. Banerjee and A. Lavie, "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments", in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72, 2005.

[16] Lin, C.Y, "Rouge: A package for automatic evaluation of summaries." in *Text Summarization Branches Out;* Association for Computational Linguistics: Barcelona, Spain, 2004.

[17] R. Vedantam, C. Zitnick and D. Parikh, "Cider: Consensus-based image description evaluation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4566–4575, 2015.

[18] M. Tanti, A. Gatt and K. Camilleri, "What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator?," in *Proc. 10th International Conference on Natural Language Generation (INLG'17)*, 2017.

[19] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556 [cs.CV]*, 2014.

[20] M. Soh, "Learning CNN-LSTM Architectures for Image Caption Generation", 2016.

[21] Haoran Wang, Yue Zhang, and Xiaosheng Yu, "An Overview of Image Cation Generation Methods", in *Computational Intelligence and Neuroscience*, 2020.

[22] Andrej Karapathy, L. Fei-Fei, "Deep Visual-Semantic Alignments for Generating Image Descriptions", in *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 664–676, 2017.

[23] Xihui Liu, Hongsheng Li, Jing Shao, Dapeng Chen, Xiaogang Wang, "Show, Tell and Discriminate: Image Captioning by Self-retrieval with Partially Labeled Data", 2018.