A Neural Image Caption Generator

Sai Sree Sajja
Department of Computer Science and Engineering
Bapatla Engineering College (Autonomous)
(Affiliated to Acharya Nagarjuna University)
Bapatla, India
saisree0403@gmail.com

Pravallika Yechuri
Department of Computer Science and Engineering
Bapatla Engineering College (Autonomous)
(Affiliated to Acharya Nagarjuna University)
Bapatla, India
pravallikayechuri146@gmail.com

Santhoshini Yeddu
Department of Computer Science and Engineering
Bapatla Engineering College (Autonomous)
(Affiliated to Acharya Nagarjuna University)
Bapatla, India
yeddusanthoshini214@gmail.com

***Abstract*—Humans can understand the information depicted in the images accurately. In the same way creating a generative model with the help of computer vision and natural language processing that can generate natural language sentences describing the given image. In this paper we used different pretrained deep convolutional neural networks and NLP models to understand the content of an image and produce a textual description that accurately reflects the visual content and context depicted in the image and analyzed their performance using Meteor score.**

Keywords— Attention, Caption, CNN, Densenet201, GRU, Image, LSTM, Neural Networks, RNN, VGG16, Xception

1 INTRODUCTION

In the past few years, computer vision in the image processing area has made significant progress, like image classification [1] and object detection [2]. Benefiting from the advances of image classification and object detection it becomes possible to automatically generate captions to understand visual content of an image. The goal of image captioning is to automatically generate descriptions for a given image.

Image caption generation is a task that involves image processing and natural language processing concepts to recognize the context of an image and describe them in a natural language like English or any other language. It generates syntactically and semantically correct sentences. In this paper, we present a deep learning model to describe images and generate captions using computer vision and machine translation. Image caption generators can find applications in Image segmentation as used by Facebook and Google Photos, and even more so, its use can be extended to video frames. They will easily automate the job of a person who has to interpret images.

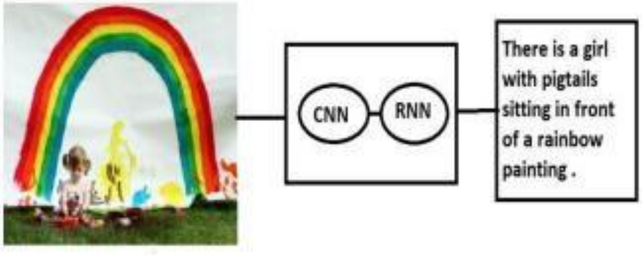

CNN
RNN
There is a girl with pigtails sitting in front of a rainbow painting .

Figure 1.1 Our model is based on a deep learning neural network that consists of a pretrained CNN model followed by a language generating NLP model. It generates complete sentences in natural language as an output.

Pretrained CNN model is a Deep Learning algorithm that will intake in a 2D matrix input image, assign importance (learnable weights and biases) to different aspects/objects in the image, and be intelligent enough to be able to differentiate one from the other.

This model was advantageous in naming the objects in an image but it could not tell us the relationship among them (that's plain image classification).

Present a generative model built on a deep recurrent architecture that unites recent advances in computer vision and machine translation and that can effectively generate meaningful sentences.

Making use of an RNN: They are networks with loops in them, allowing information to persist. LSTMs, GRUs are a particular kinds of RNN, capable of learning long-term dependencies.

Further an attention layer is used along with the RNN model to generate more relevant captions for any given image.

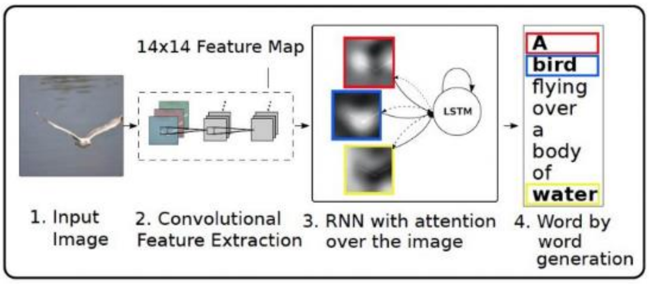

14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

Figure 1.2 Our model with additional attention layer to the pretrained CNN model and the NLP model. It generates more relevant sentences in natural language as an output.

2 RELATED WORK

In this section we provide relevant background on previous work on image caption generation and attention. Recently, several methods have been proposed for generating image descriptions. Many of these methods are based on recurrent neural networks and inspired by the successful use of sequence-to-sequence training with neural networks for machine translation. The encoder-decoder framework of machine translation is well suited, because it is analogous to "translating" an image to a sentence.

In a paper proposed [1] by O. Vinyals, A. Toshev, S. Bengio and D. Erhan (2015) used CNN and RNN for image captioning with different datasets and analysed the results.

In another paper [2] proposed by P. Voditel, A. Gurjar, A. Pandey, A. Jain, N. Sharma and N. Dubey (2023) used a specific pretrained model called VGG16 which is trained on ImageNet dataset which has more than 14 million images and a LSTM NLP model to generate captions for images

In other paper [3] proposed by A. Aker and R. Gaizauskas (2010) used N-grams to generate image descriptions using dependency relational patterns. They used a simple model based on N-Gram graphs which does not require any end-to-end training on paired image captions is proposed. Starting with a set of image keywords considered as nodes, the generator is designed to form a directed graph by connecting these nodes through overlapping n-grams as found in a given text corpus. The model then infers the caption by maximising the most probable n-gram sequences from the constructed graph

In the paper [4] proposed by Haoran Wang, Yue Zhang, Xiaosheng Yu (2020) used the attention mechanism to perform the image caption generation. An attention mechanism can be used to improve the contextual aspect of natural language sequences. The use of attention to describe image content is consistent with human understanding.

In our work we created of an image captioning models that employs an NLP model with a soft attention decoder to predict the future sentences by selectively focusing over a specific parts of an image. We used different pretrained models VGG16, Xception, Densenet201 and different NLP models LSTM, GRU to analysis the performance of each of its combinations and identified which gives better captions for any given image. We did analysis with the help of Meteor score.

3 MODELS

Deep learning uses an artificial neural network that is composed of several levels arranged in a hierarchy. The model is based on deep networks where the flow of information starts from the initial level, where the model learns something simple and then the output of it is passed to layer two of the network and input is combined into something that is a bit more complex and passes it on to the third level. This process continues as each level in the network produces something more complex from the input it received from the ascendant level.

A. Convolutional Neural Networks (CNN)

Convolutional Neural networks are specialized deep neural networks that can process the data that has input shape like a 2D matrix. Images can be easily represented as a 2D matrix. CNN is crucial in working with images. It takes an image as input, assigns importance (weights and biases) to various aspects/objects in the image, and differentiates one from the other. The below Fig. 3 demonstrates the architecture of CNN. The CNN makes use of filters which help in feature learning same as a human brain identifying objects in time and space. It has convolutional layer applies a set of learnable filters to the input image for extracting features hierarchically and Pooling layers reduce spatial dimensions. This architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and the reusability of weights.

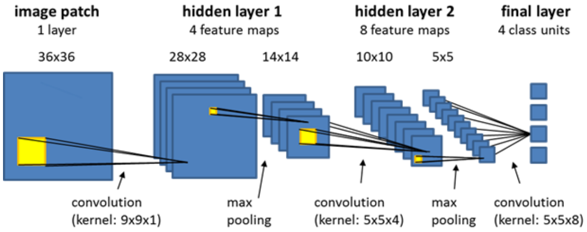


Figure 3.1 Architecture of Convolutional Neural Networks

The following CNN pretrained models are used VGG16, Xception and Densenet201. These are trained on the ImageNet dataset. So these can extract the features easily for any given images.

B. Long Short-Term Memory (LSTM)

Long Short-Term Memory networks are a special kind of RNN, capable of learning long-term dependencies. Remembering information for long periods is practically their default behavior, and this behavior is controlled with the help of "gates". While RNNs process single data points, LSTMs can process entire sequences. Not only that, but they can also learn which point in the data holds importance, and which canbe thrown away.

Hence, the only relevant information is passed on to the next layer. Their ability to remember information for extended periods of time which makes them widely used in various research areas.

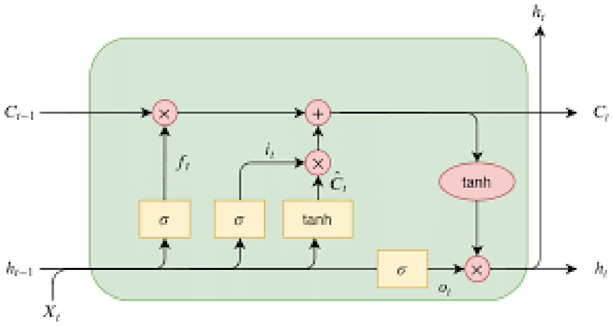


Figure 3.2 LSTM Structure

The Figure 3.2 shows the structure of LSTM. The 3 main gates involved are: input gate, output gate and forget gate. These gates decide whether to forget the current cell value, read a value into the cell, or output the cell value. The hidden states play an important role since the previous hidden states are passed to the next step of the sequence. The hidden state acts as the neural network's memory, as it is storing the data that the neural network has seen before. Thus, it allows the neural network to function like a human brain trying to form sentences.

C. Gated Recurrent Unit (GRU)

Gated Recurrent Unit is a special kind of RNN designed for sequential data by allowing information to be selectively remembered or forgotten over time. However, GRU has a simpler architecture than LSTM, with fewer parameters, which can make it easier to train and more computationally efficient.

The main difference between GRU and LSTM is the way they handle the memory cell state. In LSTM, the memory cell state is maintained separately from the hidden state and is updated using three gates: the input gate, output gate, and forget gate. In GRU, the memory cell state is replaced with a "candidate activation vector," which is updated using two gates: the reset gate and update gate

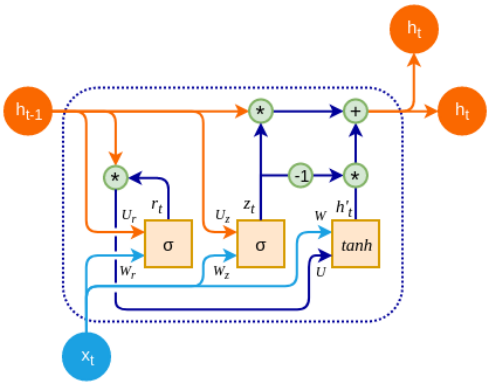


Figure 3.3 GRU Structure

The GRU architecture consists of the following components:

a) Input layer: The input layer takes in sequential data, such as a sequence of words or a time series of values, and feeds it into the GRU.

b) Hidden layer: The hidden layer is where the recurrent computation occurs. At each time step, the hidden state is updated based on the current input and the previous hidden state. The hidden state is a vector of numbers that represents the network's "memory" of the previous inputs.

c) Reset gate: The reset gate determines how much of the previous hidden state to forget. It takes as input the previous hidden state and the current input, and produces a vector of numbers between 0 and 1 that controls the degree to which the previous hidden state is "reset" at the current time step.

d) Update gate: The update gate determines how much of the candidate activation vector to incorporate into the new hidden state. It takes as input the previous hidden state and the current input, and produces a vector of numbers between 0 and 1 that controls the degree to which the candidate activation vector is incorporated into the new hidden state.

e) Candidate activation vector: The candidate activation vector is a modified version of the previous hidden state that is "reset" by the reset gate and combined with the current input. It is computed using a tanh activation function that squashes its output between -1 and 1.

f) Output layer: The output layer takes the final hidden state as input and produces the network's output. This could be a single number, a sequence of numbers, or a probability distribution over classes, depending on the task to be performed.

D. Attention Mechanism

Attention mechanisms, inspired by human visual attention, have been widely used in neural network architectures to improve the performance of various tasks, including natural language processing and computer vision. The core idea behind attention mechanisms is to allow models to focus on relevant parts of the input data (e.g., words in a sentence, regions in an image) while performing a task.

In the context of sequence-to-sequence tasks such as machine translation or image captioning, where an input sequence (source) is mapped to an output sequence (target), attention mechanisms help the model to dynamically weigh the importance of different parts of the input sequence when generating each element of the output sequence. This allows the model to selectively attend to relevant information during the decoding process, rather than relying solely on fixed-length representations or context vectors.

The above process is done as follows, The encoder will extract the features from the image. That features are fed to the attention layer Uattn and the Wattn takes the word from hidden state. The Vattn performs addition of Uattn and Wattn outputs.

The Vattn gives the scores and by applying softmax function to its output we get Attention weights.That features and attention weights are multiplied giving context vector.The context vector and words from decoder are concatenated and fed to GRU followed by fully connected layers.Finally it generates a word in each time step.

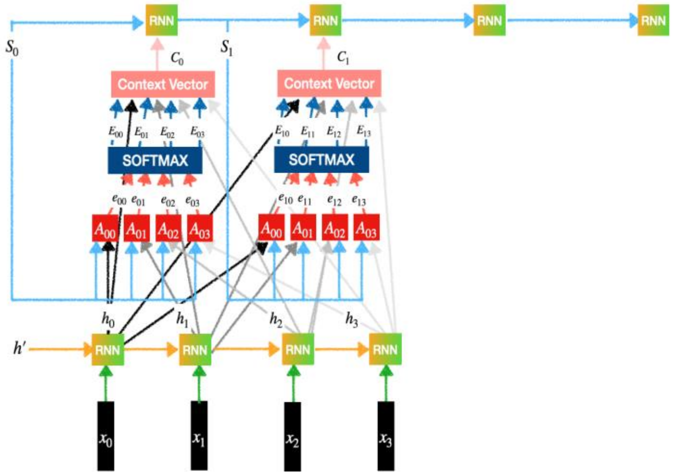

RNN
RNN
RNN
RNN
S_0
C_0
S_1
C_1
Context Vector
Context Vector
E_{00} E_{01} E_{02} E_{03}
E_{10} E_{11} E_{12} E_{13}
SOFTMAX
SOFTMAX
e_{00} e_{01} e_{02} e_{03}
e_{10} e_{11} e_{12} e_{13}
A_{00} A_{01} A_{02} A_{03}
A_{00} A_{01} A_{02} A_{03}
h_0 h_1 h_2 h_3
h'
RNN RNN RNN RNN
x_0 x_1 x_2 x_3

Figure 3.4 Attention Structure

Image Caption Model with Attention consists of four logical components:

a) Encoder: As the pre-trained CNN model encodes the image, the Encoder which consists of a Linear layer that takes the pre-encoded image features and passes them on to the Decoder

b) Sequence Decoder: This is a recurrent network built with GRUs. The captions are passed in as the input after first going through an Embedding layer.

c) Attention: As the Decoder generates each word of the output sequence, the Attention module helps it to focus on the most relevant part of the image for generating that word.

d) Sentence Generator: this module consists of a couple of Linear layers. It takes the output from the Decoder and produces a probability for each word from the vocabulary, for each position in the predicted sequence.

The encoder creates a hidden state at each step. The attention component generates a context vector that capture the relevant information from the encoder’s hidden states. The context vector is fed into decoder along with the current hidden state, to predict the next token.

4 Experiments

We performed an extensive set of experiments to assess the effectiveness of models using few metrics, model architectures on a data source, in order to compare to prior art.

4.1 Dataset

We have used the Flickr 8K dataset as the corpus. The dataset consists of 8000 images and for every image, there are 5 captions. The 5 captions for a single image help in understanding all the various possible scenarios. The dataset has a predefined training dataset Flickr_8k.trainImages.txt (6,000images), development dataset Flickr_8k.devImages.txt (1,000 images), and test dataset Flickr_8k.testImages.txt (1,000 images).

Generating a caption for a given image is a challenging problem in the deep learning domain. It is small in size. So, the model can be trained easily on low-end laptops/desktops. Data is properly labelled. For each image 5 captions are provided. The dataset is available for free. It is a new bookmark collection for sentence-base image description and search. It doesn’t touch original photos that we upload.

The architecture proposed is an encoder-decoder attention-based architecture. The encoder is one of the Xception, Densenet201, VGG16 models which is pretrained on ImageNet dataset. The decoder is one of the LSTM, GRU models. Images are present in Flickr8k_dataset folder and each image is associated with 5 captions.

4.2 Performance Metrics

To assess the performance of model prediction, we can use various metrics.

The performance metrics gives score for each of the generated caption by comparing with the caption given by the user. It helps us to identify which model predicted the caption more relevant to user given caption. It only helps to measure the performance of the model

a) BLEU (Bilingual evaluation understudy): a well-known machine translation statistic is used to measure the similarity of one sentence with reference to multiple sentences. It was proposed by Papineni et al. It returns a value where a higher value represents more similarity.

This method works by counting the number of n-grams in one sentence with the n-grams in the reference sentence. A unigram or 1-gram represents a token, whereas a bi-gram indicates each pair of a word. For calculating the BLEU score of multiple sentences, Corpus BLEU is employed, in which a reference list is indicated using a list of documents, and a candidate document is a list where the document is a list of tokens.

b) METEOR (Metric for Evaluation for Translation with Explicit Ordering): Unlike BLEU, this metric calculates F-score based on mapping unigrams. Meteor score is a metric used to evaluate machine translation by comparing it to human translations.it takes both accuracy and fluency of the translation, as well as the order in which words appear.it ranges from 0 to 1.

5 RESULTS

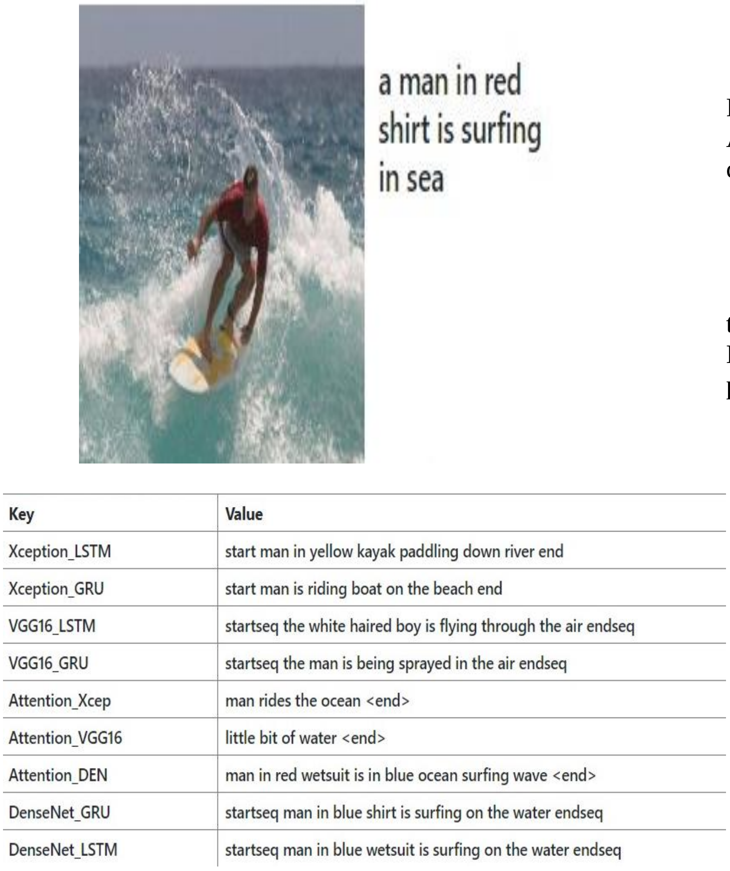
This section will show the results of the image caption generator with different methods and their comparison with the actual caption and with other method captions.

This comparison can be done in two ways. Either with human evaluation or with performance metrics.

Various images have been subjected to testing, and the following result is obtained.

5.1 Comparison with human evaluation

Human can evaluate the image by understanding context and content of the image by observing visual features. The following will show the captions generated by the different models.



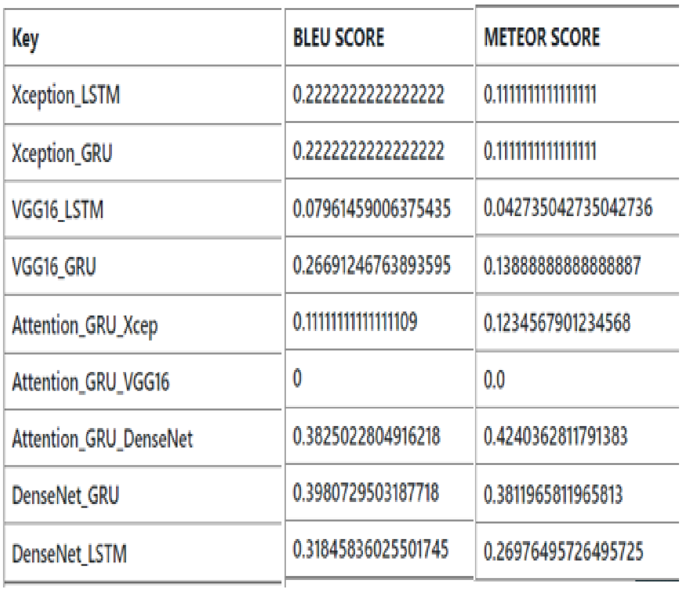
a man in red shirt is surfing in sea

Key	Value
Xception_LSTM	start man in yellow kayak paddling down river end
Xception_GRU	start man is riding boat on the beach end
VGG16_LSTM	startseq the white haired boy is flying through the air endseq
VGG16_GRU	startseq the man is being sprayed in the air endseq
Attention_Xcep	man rides the ocean <end>
Attention_VGG16	little bit of water <end>
Attention_DEN	man in red wetsuit is in blue ocean surfing wave <end>
DenseNet_GRU	startseq man in blue shirt is surfing on the water endseq
DenseNet_LSTM	startseq man in blue wetsuit is surfing on the water endseq

Figure 5.1 Captions by different methods

By human evaluation we can say that Attention with densenet, GRU with densenet and LSTM with densenet gave more relevant caption for the given image compared to other methods.

5.2 Comparison with Performance Metrics



Key	BLEU SCORE	METEOR SCORE
Xception_LSTM	0.2222222222222222	0.1111111111111111
Xception_GRU	0.2222222222222222	0.1111111111111111
VGG16_LSTM	0.07961459006375435	0.042735042735042736
VGG16_GRU	0.26691246763893595	0.13888888888888887
Attention_GRU_Xcep	0.11111111111111109	0.1234567901234568
Attention_GRU_VGG16	0	0.0
Attention_GRU_DenseNet	0.3825022804916218	0.4240362811791383
DenseNet_GRU	0.3980729503187718	0.3811965811965813
DenseNet_LSTM	0.31845836025501745	0.26976495726495725

Figure 5.2 Metrics using BLEU and METEOR scores

From the above scores comparison we can say the Attention with Densenet gave better performance compared to other methods.

6 CONCLUSION

From our experiments and their evaluation with the help of performance metrics we conclude that.In NLP Models Attention + GRU gives best caption, and in pretrained models DenseNet give better performance.

So the method with Attention+GRU+DenseNet gives more relevant caption compared to other methods in most of the cases. We hope that the results of this paper will encourage future work in using visual attention. We also expect that the modularity of the encoder-decoder approach combined with attention to have useful applications in other domains.

ACKNOWLEDGMENTS

The authors would like to thank professor Dr.N.Sudhakar, Prof for guiding throughout the work and the authors would also like to thank Show, Attend and Tell research paper writers for providing base to our work

REFERENCES

[1] Ahmet Aker, R. G. (2010). Generating Image Descriptions Using Dependency Relational Patterns. *48th Annual Meeting of the Association for Computational Linguistics.*

[2] Cao, P. Y. (2019). Image captioning with bidirectional semantic attention-based guiding of long short-term memory. *Neural Processing Letters, 50(1)*, 103-119.

[3] Girshick, R. e. (2015). Region-based

Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Transactions on Pattern Analysis & Machine Intelligence.*

[4] Haoran Wang, Y. Z. (2020). An Overview of Image Caption Generation Methods. *Computational Intelligence and Neuroscience, 2020*, 13.

[5] Huang, G. &. (2019). c-RNN: a fine-grained language model for image captioning. *Neural Processing Letters, 48(2)*, 683-691.

[6] Krizhevsky, A. I. (2012). ImageNet classification with deep convolutional neural networks. *International Conference on Neural Information Processing Systems Curran Associates Inc.*

[7] Preeti Voditel, A. G. (2023). Image Captioning-A Deep Learning Approach Using CNN and LSTM Network. *3rd International Conference on Pervasive Computing and Social Networking (ICPCSN).*

[8] Ren, X. W.-J. (2017). Deep reinforcement learning- based image captioning with embedding reward. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[9] Rennie, E. M. (2017). Self- critical sequence training for image. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[10] Vinyals, A. T. (2015). Show and tell: A neural image caption generator. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[11] Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and Tell: A Neural Image Caption Generator. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[12] Yang, L. &. (2019). Adaptive syncretic attention for constrained image captioning. *Neural Processing Letters, 50(1)*, 549-564.