# Automatics Image Caption Generator

Shivam Mittal
2015csb1032@iitrpr.ac.in
Department of Computer Science,
IIT Ropar

Shubham Dham
2015csb1117@iitrpr.ac.in
Department of Computer Science,
IIT Ropar

## 1 Abstract

Automatic image caption generator is a challenging task which is gaining popularity in the recent years. In this paper we present our generative model for caption generation using Recurrent architecture of Long Short Term Memory (LSTM) network. Our approach ushers in the possibility of use of multi-modal fusion of R-CNN and CNN for feature extraction of image, that are used to generate image captions for images using translation generation with LSTM. We also introduce a class level caption generation that focusses on a single class of all the images present in the popular image captioning datasets. Our model was able to reach considerable scores for famous image captioning metrics with best blue score **50.7**, CIDEr score **25.7**, ROUGE_L score **41.3** and METEOR socre **17.6**.

## 2 Introduction

A picture is worth a 1000 words, humans have this capability to extract a lot of information given an image. But replicating this ability for machine vision is a challenging task.
Description of the image using captions find great use in real world. It can be used for compact description of the images or videos shared on social media or any other platform. It also finds wide application helping visually impaired people understand the semantic content of an image. With the image captured in cell phones, caption generated can be read out loud to the visually impaired people giving them better understanding of visual environment. It can be also used to tag images by semantic description, and can be used by search engines to search photos.
The problem is formulated as image captioning problem, the task is to make a model that automatically generates captions for an image provided. The caption generated must semantically relate to the content of the image. That is, the caption not only depicts the types, classes of the objects in the image but also illustrates how the objects are related to one another and the kind of activities they are involved in.

Image classification and recognition tasks are very well approached problems and remarkable progress have been done in these tasks [19] . But, captioning a image is a much more difficult task, since apart from recognizing the objects in the image, it also involves understanding the attributes of these objects, interaction and relation of those objects with each other, and then generating a semantically meaningful sentence describing the scene.

Image captioning can also be viewed as a machine translation problem, where the source is the pixels of the image and the target is a sentence in English language. Many machine translation methods have been adapted to this problem such as the scoring approach BLEU [16].
The primary computer vision challenge of the task it two fold. First is to be able to design a model powerful enough to learn the features necessary for describing the objects and how they are related. Hence the Convolution Neural Networks (CNN) find application in this prospect. CNN have evolved with time to best describe the images, specifically in the task of image classification and object recognition. With each layer depicting some information regarding the image, as we go deeper into the layered structure, information at each level starts holding much complex and useful relations and meaning with respect to the image. Hence the deeper layers of CNN becomes an obvious and inevitable source to depict the characteristics of the image. Second task is a language processing challenge, i.e, to be able to describe the visual knowledge of the image in a natural language like English. Hence a language model becomes the second main part of the challenge. Recurrent Neural Networks (RNN) is the most prevalent and productive when it comes to the task of translating a sentence from a source language to the target language.

Most of the works till now have been pipeline based in which the task of learning the visual descriptor and language model are treated as two sub problems and the solutions are stitched together to get a description from an image. On the similar approach, we propose our model which initially finds best representation of the image using Convolution Neural Network (CNN) and Region based Convolution Neural Network (R-CNN), and this feature vector representation is fed into a language generation model - Long Short Term Memory (LSTM) network. The task executed is such as to find the corresponding words of the captions that have maximum probability and the cross entropy loss over the entire caption is minimized.
Our neural network is trained using batch gradient descent. The pretrained GoogleNet Inception-V4 model [2] was used as Convolution Neural Network (CNN) to extract the features from the images. VGG16 model from tensorflow was used, whereas for R-CNN [1], a model trained on MS-COCO dataset was used.

## 3 Related work

The task of caption generation for images has been widely studied in recent times. From template based caption generation to retrieval methods for caption generation, and most recent state of the art, i.e, end to end fusion of CNN and RNN, caption generation task has been a hot topic of research in recent times.
The beginning pipe lining based approach was put forward by Farhadi et al.[7] in which image was perceived as a triplet of scene elements and were mapped to most relevant words from the dictionary. These words were then used to generate captions for the image. A similar approach was used by Li et al. [14] where sentence were captions were generated out of phrases that fit different sub-regions of the image.
A corpus based approach was put forward by Yang et al. [23] where object detection followed by caption generation using a template was done. Kuznetsova et al. [12] used retrieval approach where they implemented visual recognition to identify distinct elements in the image. Based on the visual features extracted, most similar images were found in the training data set and blend of their captions was used to generate caption for the test image. Retrieval technique [13] [9] as well as direct caption generation [5] from the words extracted from visual recognition of the image did not hold well on unseen data. Comparison with images in the training data set and further caption generation did not perform well on the new data with new combination of objects and scenes as there was no attempt to generate novel captions.
With the limitations of the above methods, attention was diverted towards learning a language model that does the task of converting a source language or features into target language. First Convolution Neural Networks came into application [11], where the images characterization and classification was done with very accurate results. Further advancements came with the popularization of Recurrent Neural Networks [8]. Then the caption generation shifted gears with image characterization done with CNN and RNN used as language model to generate captions out of the features generated by CNN [4] [10]. Fang et al. [6] divided image into sub-regions with each region featurised using CNN. Finally these features were mapped to words from the caption with Multiple Instance Learning (MIL). The paper proved to be state of the art until Vinyals et al. [22] proposed end to end generative model that takes an input image, and is trained to maximize the likelihood $p(S|I)$ of producing a sentence S which is a target sequence of words part of the dictionary. It is a neural and probabilistic framework, they combined deep convolutional network for learning the features of image, and a LSTM based sentence generator to a create a single network which is trained on the loss function given by the sum of negative of the log likelihood of the correct word. The Show and Tell paper pioneered the field of caption generation with setting new bench marks in the field of CNN and RNN application to generate captions with exceedingly accurate results. Rennie et al. [18] remains the current state of the art with new its new approach in application of CNN

and RNN. They devised a Self Critical Sequence Training (SCST) which is Reinforce algorithm that uses test time inferences to optimize the gradient of the rewards experienced while training. They also modified the use of CNN by giving attention to particular region of interest at each time step.
The methods described above and in use are practically divided into two categories with each having its own pros and cons, are discusses below:

- **Generation** : This approach first detects visual features in the image in terms of objects, attributes, actions etc. Then these visual features are fed into a language model like RNN to generate captions of the image in sequential manner [6] [22] [14].

- **Retrieval** : This approach first extracts visual information from the image and then finds images in the training database that closely relate with the input novel image. Then the descriptions for the image are synthesized by simply blending the captions of the image that most relate to the input image. Hence captions are generated ny simply retrieving the information of the training data set. This can be further sub divided into two categories.

  - The images are compared with other images in the training images based only on the visual space or visual content of the images [12] [6].
  - The visual data and captions of the images are mapped into single feature space which is then used to make comparisons among train and test images [10] [9].

The models that formulate the task as retrieval problem have a advantage that the descriptions generated by them would be grammatically correct because the design of the model is such that they transfer the retrieved human generated sentences to the novel images. But the disadvantage of this approach is that massive amount of training data is needed to be able to match a novel image to a seen example, and also the training data needs to be diverse. Whereas, the methods that formulate the problem as a generation task have an advantage that they can produce novel descriptions, they have a possibility to work on unseen type of examples. Theoretically, they can even work when we have limited amount of data. However, the disadvantage of these methods is that there accuracy is dependent upon the visual understanding and the ability to correctly verbalize the visual understanding. However, difficulty lies in both the vision techniques because in practice they do not have very high accuracy, if some important attribute or object is not identified, then it is impossible to generate a correct description. Also, maintaining the language fluency and grammatical correctness of the language generation model is also a challenge in itself.

The methods of retrieval from multi-modal space have the same advantage as that of retrieval from visual space. They have an added advantage because such multi-modal spaces can also be used if the problem is reversed i.e. we need to generate image for a given description. However there are added disadvantages in this model, a multi-modal similarity metric that can compare image and sentences needs to be developed which is harder than developing a visual space metric, also even more amount of training data is needed to train the common space of image and sentences.

With analysis of both the approaches described above and the papers in the respective field, we present our generative model approach to generate the captions for the images. The novelty of our approach comes with the addition of R-CNN features above the CNN features extracted for the images. The exceedingly well performance of R-CNN [17] in object detection helps perform object detection at best with the use of CNN extracted features to build the relations and connections among the various entities in the images and scenarios. The image representation as obtained above is used as feed input to the language generation model, i.e, LSTM which performs sequential caption generation with each word selected of maximum probability and resultant minimum loss for the entire caption.

# 4 Model

**Overview:** Our approach focuses on significant object detection and subsequent correlation determination among them. With the ultimate goal of generating captions for the images, our approach comprises a two level neural network model, with first level for feature extraction and second level for sentence generation. Feature extraction makes use of Region based Convolution Neural Network (R-CNN) and simple Convolution Neural Network (CNN). The R-CNN to detect object and CNN to detect holistic view of the image, hence the combination providing a fruitful representation of an image to be fed into translational network. Sentence generation uses Long Short Term Memory (LSTM) network, which generated appropriate depiction of the image with required exclusion and update of necessary information.
We use pre-trained networks for RCNN and CNN to extract the features from an image, whose details we will be discussing later.
Then we trained the LSTM on cross entropy loss to basically maximize the probability of generating the correct description. For an instance i in the training data, we have $(x_i, y_i)$ where $x_i$ is the image and $y_i$ is the caption given for the image. We will try to adjust the parameters of our LSTM model $\theta$ such that the log likelihood of the correct caption is maximized:

$$\theta* = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} logp(y_i|x_i;\theta)]$$

The log likelihood of a sentence $y_i$, can written in terms of the individual words in that sentence that are $y_{i1}, y_{i2}, ..., y_{iT}$ where T is the maximum length of a sentence possible.

$$logp(y_i|x_i;\theta) = \sum_{t=1}^{T} logp(y_{it}|x_i, \theta, y_{i1}, y_{i2}, ..., y_{i(t-1)})$$

The probability maximization that we are trying to model is accurately modelled by a recurrent neural network, and we are using Long Short Term Memory (LSTM) net which has state of the art perform in sequence translation tasks. The details of the LSTM will be provided in later sections.

## 4.1 Image Representation and Feature extraction

The image representation is done to extract the most relevant features required to generate captions that depict the objects present in the image as well as the link between them.
The ImageNet challenge [19], in the past few years, saw the introduction of various neural network models that process images to give pretty accurate results on object localization, object detection, scene detection, scene parsing, etc. The ending hidden layers nodes for the respective models carry very abstract, complex and meaningful information for the images. Hence the nodes of these layers are the obvious and enough representations for the images that provide ample information.
The CNN model GoogleNet inception v4 [20] pretrained on Imagenet [19], taken from [2] was used to extract features from all the images. The last hidden layers nodes values were used as the representation of the images. The three channels of the image were flattened out to create a feature vector of **512 * 3 = 1536** as a feature vector for each image.

### 4.1.1 Faster R-CNN

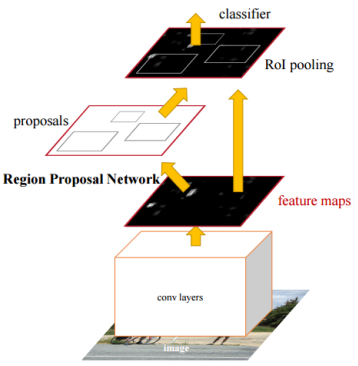


Figure 1: R-CNN

The idea of using a object detector was to increase the focus of the feature representation on the objects being detected so that the caption

being generated doesn't make a mistake and describes the object with reasonable accuracy. The idea was from the fact that if the image being fed to the CNN is the whole image, then there might be significant background information which might be less important than the object information. Although we cannot discard the background information because it might consist some information useful for generating the overall caption. So, we create a multi-modal feature representation for our images as described below.

In the pipeline of fast-rcnn, the slowest part is generating regions from Select Search( 2s) or edgeboxes( 0.2s). Faster-RCNN [17] replaces Selective Search with CNN itself for generating the region proposals(called RPN-region proposal network) which gives out tentative regions at almost negligible amount of time. This is done by using the convolutional layers from detection network (therefore no overhead) and introducing two convolutional layers on top of this (in parallel to FC layers of detection network) to generate regions at various spatial location. Since the conv layers are shared it does not add to the computation time and the only additional time involved is the two additional conv layers which have relatively small number of filters. So for RPN a small network with kernel size of 3X3 is run through the final conv feature map and a smaller 256 dimension feature is obtained at every spatial location. This is then fed to the two sibling layers just as in previous detection network for the two tasks of classification and localization.

Faster R-CNN was used to detect objects in the image. R-CNN with its very good accuracy in object detection helps detect objects. Faster R-CNN with Inception and Resnet as their internal CNN [1] trained on MS-COCO data set was used. The bounding box obtained in the output of the R-CNN were used to obtain the objects. The objects were then extracted such that, in the whole image everything was turned black except the object region. Hence each output image from the R-CNN highlighted each object with all the remaining area turned black.

This image was then fed into the Inception v4 CNN model described above along with the simple image, so we get features for two image each of size 1536, we concatenate the image features from both the images, hence the output representation of each image was [**2 * 1536 = 3072** vector.

### 4.2 Sentence Generation

The sequential data generation using Recurrent Neural Network (RNN) faces the issue of exploding and vanishing gradients. The issue is resolved with the use of Long Short Term Memory (LSTM) Network which has additional gates to forget and update the information in each cell.

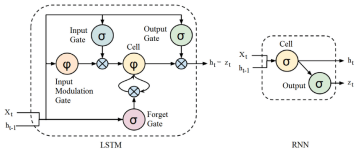


Figure 2: LSTM cell

The LSTM cell generates sequential data based upon previous inputs and the present input. These inputs are presented in timesteps where each time stamp correspond to generation of single word with maximum probability. The further words for each sentence are generated sequentially with regard to the previous generations. The heart of the LSTM network is a single cell with three gates to which take either 0 or 1 value. Based upon which the decision is made whether to update the previous values, forget the input and to give the new output. The equation for the gates are as follows:

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1})$$

$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1})$$

$$o_t = \sigma(W_{ix}x_t + W_{om}m_{t-1})$$

$$c_t = f_t \odot c_{t-1} + i_t \odot h(W_{cx}x_t + W_{cm}m_{t-1})$$

$$m_t = o_t \odot c_t$$

$$p_{t+1} = Softmax(m_t)$$

where $\odot$ represents multiplication with gate value, i.e., 0 or 1 and W are trained weights. With the helps of these gates LSTM overcomes the problem of vanishing and exploding gradients. The non linearities are added with two functions - $\sigma$ and hyperbolic tangent (h). $m_t$ is what is used to feed to probability $p_t$ to find probability of each word over others.

### 4.3 Training

The representation of the images generated with CNN and R-CNN were used as the feature vector, to be used for training the LSTM as well as testing data. The feature vector is fed into the LSTM network to learn the weights.

The captions for all the images are initially pre-processesed to create the vocabulary and word-to-index values for each word in the dictionary. Each caption in the set is padded to the maximum length of the caption with a dummy value, and with additional markers at the beginning and ending of the caption to detect the start and ending of a caption. This length is equal to the number of timesteps, i.e., number of times a caption is parsed sequentially in order to generate its words.

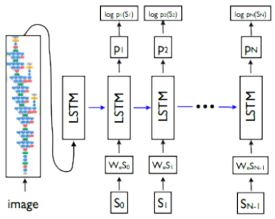


Figure 3: unrolled LSTM cell

To understand the training process, let us unroll the LSTM network, i.e, same cell repeated equal to timesteps times, with all cells having same number of parameters. Hence each cell in the unrolled network at any timestep $t$ has input as o of previous cell $t-1$, i.e., $m_t$ and a word of the caption.

$$x_{-1} = CNN(I) + R - CNN(I)$$

$$x_t = W_e S_t, t \in \{0...N-1\}$$

$$p_{t+1} = LSTM(x_t), t \in \{0...N-1\}$$

where I is the input Image. Each word of the caption is one-hot encoded, with size equal to the size of the vocabulary. $S_0$ denotes a marker which depicts beginning of a sentence and $S_n$ denotes a marker which depicts ending of a sentence.

Cross entropy loss is used over the probabilities generated for each word in the vocabulary at each step of the timestep.

$$Loss(o_i, y_i, t) = -\sum_{k=1}^{K} y_{i,t,k} * logo_{i,t,k}$$

This is the loss given for timestep t, where $o_{i,t}$ is the predicted output at time-step t from our model, and $y_{i,t}$ is the one-hot encoding for the actual encoding. k varies from 1 to K where K is the number of words in the vocabulary.

This loss is minimized during the training process and caption generated is such that the probability of the correct word being generated is maximum and loss over entire sentence is minimum.

## 5 Experiments

### 5.1 Data Set Pre-Processing

The caption generation for the project is focused on a single class of all the images present in the dataset. The Flickr30K dataset has generic images with not any particular class or similarity. Hence we pre-processed data to obtain images on **dogs** only, i.e., all the images that had dogs or anything related to them.

The words dog/dogs was searched in the caption file and all the captions containing the mentioned word were separated with the corresponding image. Hence our final data set had around **2200** images and all these

images were in some way related to dogs. Initially all the image in the Flickr30K data set had 5 captions, but after processing, this generality did not hold and the number of captions for each image varied from 1 to 5 because for a single image some of the captions had the keyword we searched for, but some of the captions did not. The average caption per image came down to 4.2 from 5.
The dataset was then split into training, testing and validation in the ratio (80:10:10).

### 5.2 Evaluation protocol

The best way to evaluate the output of a model is human subjective evaluation. Human raters can be asked to give subjective scores to the generated caption based on its usefulness and accuracy in describing the image. But, since human evaluation is very expensive, we need need automatic measures that correlate with the human rated subjective score.
The popular automatic measure used for this task are BLEU [16] which is a fraction of n-grams in common between hypothesis and set of reference, ROUGE [15] using overlap of n-grams between system and reference summary, Meteor [3] using unigram precision and recall, CIDEr [21] which uses a method based on triplet to collect human annotations in order to measure consensus. Except CIDEr, the other measures were originally developed for the evaluation of text summarization systems or Machine translation systems, but CIDEr was developed exclusively for the task of image captioning. The score computed by each of the measures is an indication of the similarity between the automatic system output and the human reference text (ground truth captions in our case). However, there is a limitation of these measures, as in the MS COCO 2015 challenge, some models as evaluated by these automatic measures seemed to outperform the human upper bound. However upon human evaluation, even the best system did not perform better than the human upper bound. This shows that the correlation between the automatic measure and the human evaluation is not very high.
Some methods have more limitations than other, for example the METEOR metric gives a more reliable score than the BLEU score. Even though these measures suffer from some limitations, however these are the standard evaluation metrics being used for this task.

Table 1: Blue Score without Dropout

| Blue | Score without RCNN | | Score with RCNN | |
|---|---|---|---|---|
| | nodes:64 | nodes:254 | nodes:64 | nodes:254 |
| Blue_1 | 42.6 | NA | 43.4 | NA |
| Blue_2 | 25.2 | NA | 25.6 | NA |
| Blue_3 | 14.8 | NA | 15.2 | NA |
| Blue_4 | 9.0 | NA | 9.5 | NA |

Table 2: Blue Score with Dropout

| Blue | Score without RCNN | | Score without RCNN | |
|---|---|---|---|---|
| | nodes:64 | nodes:254 | nodes:64 | nodes:254 |
| Blue_1 | 50.7 | 48.8 | 49.4 | 46.6 |
| Blue_2 | 33.4 | 31.1 | 32.5 | 28.2 |
| Blue_3 | 21.2 | 19.3 | 19.9 | 15.7 |
| Blue_4 | 13.1 | 11.7 | 12.3 | 8.7 |

The experiments were done so as to find out the effect of using R-CNN features in the image caption generation task. The hypothetical idea one generates based upon the efficiency of R-CNN in object detection is that the respective features from R-CNN should help detect individual objects with improved accuracy. On top of this object detection from the bounding boxes generated, the CNN features should help define connections between various objects detected along with holistic scenario of the image as well. Based upon this judgement experiments were carried out. Hence the training was done with input features of CNN and R-CNN both combined as well as only using CNN features also, so as to compare the results of the two modalities. The multi-modal approach of CNN and R-CNN combined was the main core idea which was to be analysed. Further the test results were obtained for both the approaches. The testing was done on the feature extracted on previously separated 200 images from the training images.

Further experiments were carried to notice the effect of number of nodes in the LSTM cell and dropout on the scores obtained for both the approaches. This was also done for all the metrics.

Table 3: Other metric Score without Dropout

| Blue | Score without RCNN | | Score with RCNN | |
|---|---|---|---|---|
| | nodes:64 | nodes:256 | nodes:64 | nodes:256 |
| CIDEr | 11.9 | NA | 15.0 | NA |
| ROUGE_L | 34.1 | NA | 34.6 | NA |
| METEOR | 14.3 | NA | 14.5 | NA |

Table 4: Other metric Score with Dropout

| Blue | Score without RCNN | | Score with RCNN | |
|---|---|---|---|---|
| | nodes:64 | nodes:256 | nodes:64 | nodes:256 |
| CIDEr | 25.7 | 22.2 | 22.6 | 16 |
| ROUGE_L | 41.3 | 38.6 | 40 | 37.2 |
| METEOR | 17.6 | 16.9 | 16.4 | 16.2 |

## 6 Results


Figure 4: Example of captions produced by our model. The left two captions are produced by the model with RCNN and the right two captions are produced by model without RCNN

The final results obtained for our approach is summarized in the **Table 1** and **Table 2**. The accuracies for the caption generated on test set from the features extracted using CNN and R-CNN combined as well as only CNN were scrutinized. The Blue score for the results obtained using R-CNN features is lower than the one generated without using R-CNN. This is evident for all the four classes of the Blue score. Also the Scores from the other metric that is METEOR, CIDEr and ROUGE_L are also less for the one with R-CNN. Hence the captions generated without R-CNN are more close to human generated captions. These resultant scores obtained are independent of the presence of dropout or the number of nodes in the LSTM cell. This observation observed from results of our experiments clearly states that addition of R-CNN extracted features makes no improvement in the task, rather somewhere degrades the scores to a minor extent. The reason for this can be that R-CNN extracted objects do not provide extra information to the LSTM model, i.e, the CNN in itself is able to detect objects very accurately. Hence the addition of the R-CNN features results in repetition of the features which rather than improving the performance results in the overhead which mars the performance of

the network.
Also it is evident from the scores obtained for different metrics that dropout has profound effect on the caption of the image generated. The BLue score as well as other scores are less without dropout than when the dropout is implemented in the network. This is observed for both the modalities, i.e, scores obtained for features using R-CNN and without using R-CNN. Thus it is evident that implementation of dropout is helpful as it improved the performance in our case. The reason behind could simply be the effect that dropout normally has, i.e, reduces ovefitting and to some extent implements the effect of ensemble.
Also the effect of number of nodes in the LSTM cell was examined. The results observed shown that increasing the number of nodes resulted in decrease in the performance, independent of other factors. This was also true for all the four metrics. The reason behind could be that lesser number of nodes are enough to learn the content of the images being parsed, i.e, lesser weights are able to learn the information required to generate meaningful captions. Hence on increasing the number of weights, redundant or extra information is learned that is not required for the task at hand. But this behaviour is actually opposite to what is generally observed with the increase in number of nodes. The information initially held as 2 * 1536 features is directly reduced to 64 nodes, which shall give relatively less performance as compared to 254 nodes. Hence reason for this observation is difficult to infer.

## 7 Conclusion and Future Work

We introduced a model to generate captions for a subset of the in-general data sets used for the image captioning task. The subset selected was that of the images related to dogs only. All the images used were directly or indirectly related to dogs this not so large data set was framed out of the available Flikr30K dataset. Further we introduced the use of R-CNN features in the image captioning task. In our approach, in addition to the simple image used to generate CNN features, another image was also used in which only the objects detected by R-CNN were highlighted. This approach is completely new in the field, and our attempt has somehow obtained the considerable results for this particular approach. The scores obtained for all the mentioned metrics are considerably good with best blue score **50.7**, CIDEr score **25.7**, ROUGE_L score **41.3** and METEOR socre **17.6**
The results obtained by using R-CNN are less than the results obtained without using R-CNN, but the difference between them is marginal. This is observed not only in Blue score metric but also for other image captioning metrics like METEOR, ROUGE_L and CIDEr. Hence the conclusion that can be drawn from these results is that the use of R-CNN in addition to CNN does not yield any favorable or improved results. But this cannot be generalized for any general dataset, bacause our model operates only on a sub class of all the images present in the complete dataset. Hence effect of R-CNN for other class of images or any general image can not be decided.
For the further experiments, we would like to remove the limitations that were observed, i.e, the scarcity of data set for the particular class. It is natural to say that in case of large data set available, the results obtained could completely be different, as the true essence of neural networks lie in the areas where huge data sets are available for training and evaluation.

## 8 References


[1] Object detection in tensorflow : `https://github.com/KleinYuan/tf-object-detection`.

[2] Deepdetect model : `https://deepdetect.com/models/tf/inception_v4.pb`.

[3] Satanjeev Banerjee and Alon 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*, volume 29, pages 65–72, 2005.

[4] Xinlei Chen and C Lawrence Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2422–2431, 2015.

[5] Desmond Elliott and Frank Keller. Image description using visual dependency representations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1292–1302, 2013.

[6] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. From captions to visual concepts and back. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1473–1482, 2015.

[7] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every picture tells a story: Generating sentences from images. In *European conference on computer vision*, pages 15–29. Springer, 2010.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[9] Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013.

[10] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[12] Polina Kuznetsova, Vicente Ordonez, Alexander C Berg, Tamara L Berg, and Yejin Choi. Collective generation of natural image descriptions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 359–368. Association for Computational Linguistics, 2012.

[13] Angeliki Lazaridou, Nghia The Pham, and Marco Baroni. Combining language and vision with a multimodal skip-gram model. *arXiv preprint arXiv:1501.02598*, 2015.

[14] Siming Li, Girish Kulkarni, Tamara L Berg, Alexander C Berg, and Yejin Choi. Composing simple image descriptions using web-scale n-grams. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 220–228. Association for Computational Linguistics, 2011.

[15] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain, 2004.

[16] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[17] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[18] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. *arXiv preprint arXiv:1612.00563*, 2016.

[19] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3): 211–252, 2015.

[20] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, pages 4278–4284, 2017.

[21] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[22] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[23] Yezhou Yang, Ching Lik Teo, Hal Daumé III, and Yiannis Aloimonos. Corpus-guided sentence generation of natural images. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 444–454. Association for Computational Linguistics, 2011.