# Image Captioning using Deep Learning Framework

Sai Aditya Thalluri[1]

*Abstract*— **Captioning an image is so easy for humans. Looking at a picture, people can describe it in many ways. But it is not that simple for computers. Image captioning project is to train computers to describes the images in natural language. Its interesting because it concerns what we understand about perception with respect to machines. Using the Deep Learning Models such as Convolutional Neural Network(CNN) which can analyze the patterns very well, we can build a model which can best describe the images. This can be useful in many real-time applications.**

*Keywords*— **image captioning, deep learning, convolution neural network, recurrent neural network, lstm, vggnet, inception, natural language processing.**

## I. INTRODUCTION

Image captioning is one of those applications of image analysis which has gained so much prominence in recent years[1]. When computers are able to describe images, they can understand the situations happening around them and react in proper ways. There won't be limitations in there use to any specific area and be extended to any kind of applications. Even large corporations like Google implemented it in search engines and camera applications.

It is important to know how image captioning is useful to problems in real-world scenarios. Below are the few applications where a solution to this problem can be very useful.

- Self-driving cars: If automated cars can properly caption and understand the situations around them, they will go smooth without any incidents.
- Cameras: We can see cameras everywhere today. If Cameras such as security surveillance can read the images they are capturing and are able to describe them, they can raise events such as security alarms if any inappropriate activities happen.
- Image search: If people don't understand any image or want to find related items based on an image then image captioning is the way to do that. Google already implemented this in their search engines were based on the description of the given images search results will be shown to the user.
- Aid to the Blind: It also helps the blind people by converting the things they can't see into text and the text into voice. Companies such as Nvidia are already doing research to implement this.

In order to implement image captioning, a system which can best analyze the images well is required. Convolutional Neural Network, which is a Deep Learning Model is is an efficient practice in Image analysis. With proper dataset and filters, it will be able to identify the patterns in images accurately. This is what instigates the application of CNN on Image Captioning to get the best description for the images.

## II. BACKGROUND

There is no proper solution to image captioning until the recent development of deep learning. Even advanced research in computer vision could not provide a better solution. Using deep learning, image captioning can be implemented so easily with a proper dataset. This problem was well researched by Andrej Karapathy in his Ph.D. thesis at Stanford, who is also now theDirector of AI at Tesla[9].

Earlier methods for implementation of image captioning uses Computer Vision and Natural language processing[2]. Even they use Neural Networks but not Deep Neural Networks. Convolutional neural networks are able to generalize very well when compared with any other models. So they are the best use when it comes to any of the image analysis these days.

TABLE I
COMPARISON

| Ancient CV technique | Convolutional Neural Networks |
|---|---|
| Can't generalize well | Generalizes datasets well |
| Not much accurate | More Accurate |
| Template matching outdated | Data intensive and learn so well |

So CNN is best useful for the project as they are more accurate and can generalize very well[3,4].

## III. METHODOLOGY

As image captioning involves images and their description, I need to combine two neural network models namely Recurrent Neural Network(RNN) for the image captions and Convolutional Neural Network(CNN) for the actual images.

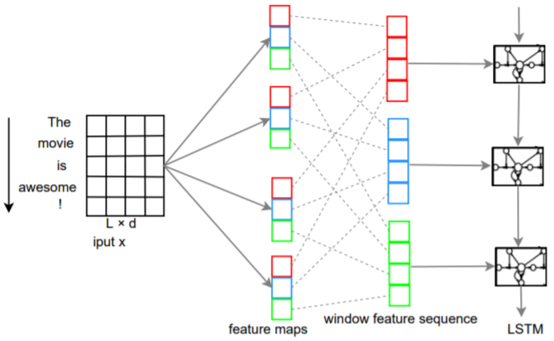
### A. Recurrent Neural Network

Since image captioning involves the processing of the image description, a recurrent neural network model is needed to analyze the image captioning sequences. Feedforward networks only take into account a fixed length of content to predict the next word in sequence. So, they are not suitable problems such as description generation. Whereas recurrent neural network considers all the predecessor words while predicting the next work in sequence. This is what makes them useful problems such as natural language processing.

*This work was not supported by any organization

[1]Sai Aditya is a Graduate student of Computer Science Department, University of New Mexico, Albuquerque, New Mexico, Unites States of America.

Long Short Time Memory(LSTM) is one of the recurrent neural networks. Below image shows the architecture of LSTM[8].

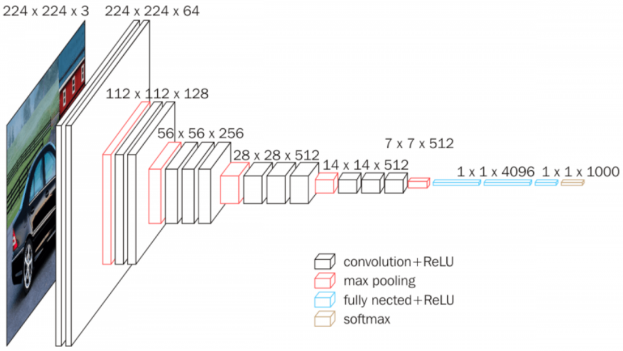


All blocks of the same color in the feature map layer connect to the corresponding colored window feature. The output of the whole model will be hidden in the last unit of LSTM

## B. Convolutional Neural Network

Convolutional Neural Networks(CNNs) are powerful models for image recognition problems. They provide the most optimized results when trained with huge data. In this project, I used two CNN models VGG-16 and InceptionV3.

*1) VGG-16:* This model was proposed by K. Simonyan and A. Zisserman from the University of Oxford's Visual Geometry Group (VGG)[6]. The model achieves 92.7 percent test accuracy which is top-5 in ImageNet. It is composed of 16 weighted layers among which 13 of them are convolutional and remaining three are fully connected layers.

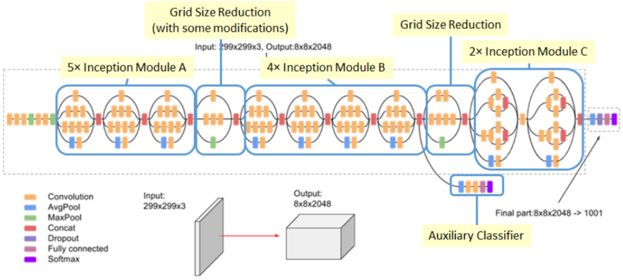


The stride and pad of convolutional layers are fixed to a size of 1 pixel. They have been divided into 5 groups with filter sizes starting from 64 to till 512. Each group also contains a max-pooling layer. Convolutional and fully connected layers are weighted with ReLu function. Whereas the output layer is weighted with a softmax activation function. Below table shows the size and strides of receptive fields in each layer of VGG-16[5].

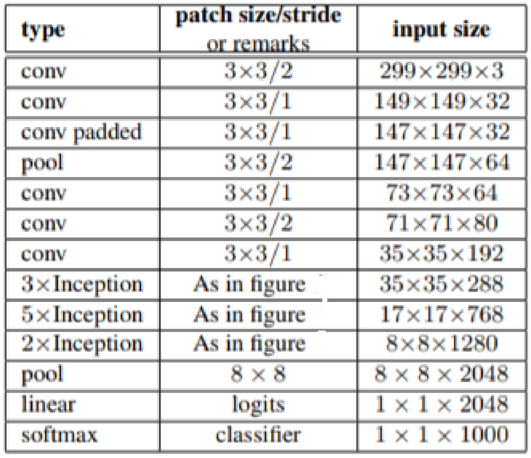
| layer | c1_1 | c1_2 | p1 | c2_1 | c2_2 | p2 | c3_1 | c3_2 | c3_3 |
|---|---|---|---|---|---|---|---|---|---|
| size | 3 | 5 | 6 | 10 | 14 | 16 | 24 | 32 | 40 |
| stride | 1 | 1 | 2 | 2 | 2 | 4 | 4 | 4 | 4 |
| layer | p3 | c4_1 | c4_2 | c4_3 | p4 | c5_1 | c5_2 | c5_3 | p5 |
| size | 44 | 60 | 76 | 92 | 100 | 132 | 164 | 196 | 212 |
| stride | 8 | 8 | 8 | 8 | 16 | 16 | 16 | 16 | 32 |

Since VGG follows a linear connection between all the layers, it is painful and takes more time for training compared to other models.

*2) InceptionV3:* Inception model from TensforFlow is developed from Google Brain Team which is a trained model and can be used for transfer learning. It has a 48 layer deep network and can classify images into 1000 categories. Below image shows the architecture of inception V3 model[7].



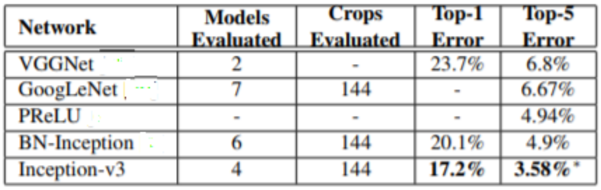
Below table shows the size and strides of each layers of inceptionV3[7].



| type | patch size/stride or remarks | input size |
|---|---|---|
| conv | 3×3/2 | 299×299×3 |
| conv | 3×3/1 | 149×149×32 |
| conv padded | 3×3/1 | 147×147×32 |
| pool | 3×3/2 | 147×147×64 |
| conv | 3×3/1 | 73×73×64 |
| conv | 3×3/2 | 71×71×80 |
| conv | 3×3/1 | 35×35×192 |
| 3×Inception | As in figure | 35×35×288 |
| 5×Inception | As in figure | 17×17×768 |
| 2×Inception | As in figure | 8×8×1280 |
| pool | 8 × 8 | 8 × 8 × 2048 |
| linear | logits | 1 × 1 × 2048 |
| softmax | classifier | 1 × 1 × 1000 |

Due to its architecture, inceptionV3 model detects the image features accurately. It is among the top-5 with an error rate of 3.58 percent.

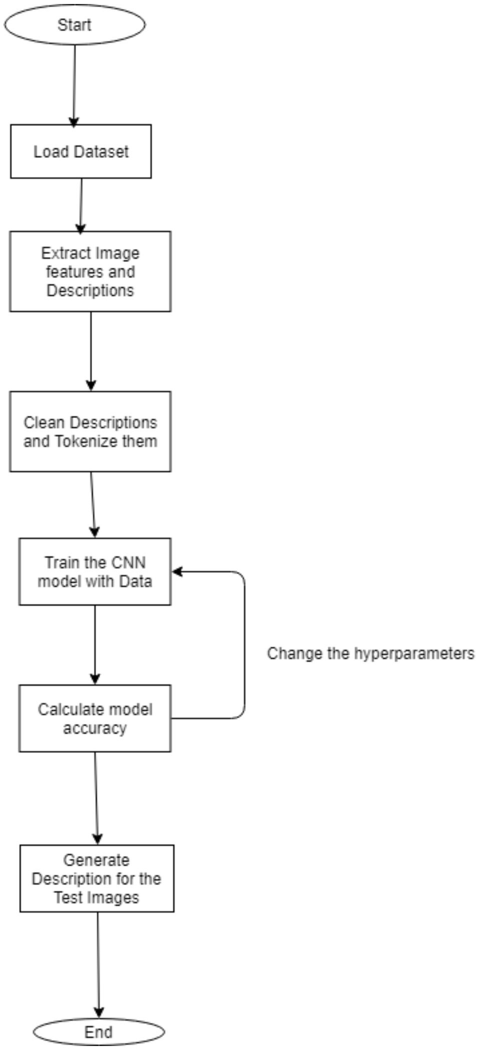
*3) Comparison between ImageNet models:* When compared to other models such as VGG, inceptionV3 performs better with the lowest error rate by training on huge datasets. Below table shows the comparison between most used CNN models[7].



| Network | Models Evaluated | Crops Evaluated | Top-1 Error | Top-5 Error |
|---|---|---|---|---|
| VGGNet | 2 | - | 23.7% | 6.8% |
| GoogLeNet | 7 | 144 | - | 6.67% |
| PReLU | - | - | - | 4.94% |
| BN-Inception | 6 | 144 | 20.1% | 4.9% |
| Inception-v3 | 4 | 144 | **17.2%** | **3.58%**$^*$ |

### C. Flowchart

As the VGG model takes more time, I have shifted my project to Inception V3 by applying transfer learning and it gives more accuracy compared to the former. To train on both the images and description, I have added the LSTM recurrent neural network layer to the Inception model. Below flow chart shows the steps in the implementation of image captioning using InceptionV3 and LSTM.


Start
Load Dataset
Extract Image features and Descriptions
Clean Descriptions and Tokenize them
Train the CNN model with Data
Change the hyperparameters
Calculate model accuracy
Generate Description for the Test Images
End

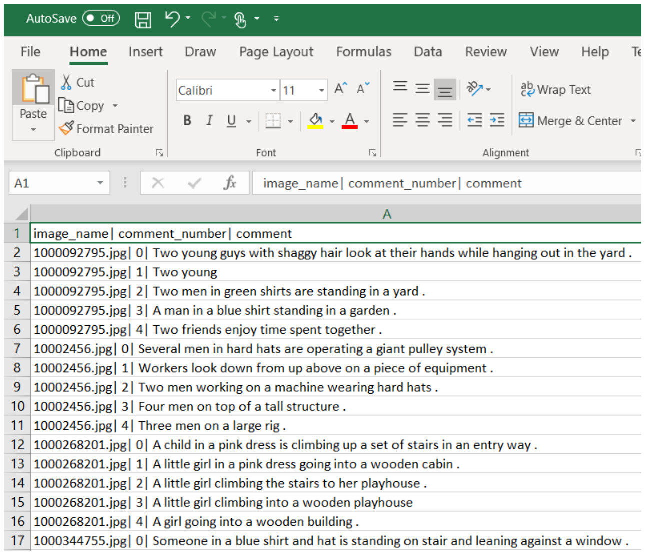
Firstly we load the images along with their descriptions from the dataset. Extract the image features using Inception model and tokenize the description using the natural language tool kit. Train the model with training and validation datasets. For tuning the hyperparameters, we will run the model through multiple epochs until a better accuracy is achieved. As it is unsupervised learning, the batch rate in each epoch has to be as large as possible. The high batch size makes the model to go through all the model multiple times to better understand all the features. After training, we will test the model on test images and draw performance inferences from the generated captions.

## IV. DATASET

Since this is a project on image analysis, a diverse dataset which covers various objects that we usually see every day is needed. As we are using CNN, the dataset should be large enough so that the model will have enough data to train on. Otherwise, it could lead to problems such as overfitting. So the dataset we choose must satisfy all these requirements.

Kaggle is one such online community when we can find excellent datasets posted by many data scientists and machine learners. We have found a Flickr dataset which will meet all the requirements[10].

The dataset basically consists of a folder *flickr8kimages* and separate CSV files containing image captions for training. validation and testing. The folder contains approximately 8k images of Flickr, which is an image and video hosting service. Among them *6k* are for training, *1k* for validation and the remaining *1k* for testing. And the CSV has file contains around *20k* captions for the images, for which there are multiple captions for the same image which helps the model to train even more accurately. Below snapshot gives a clear picture of the CSV file.



image_name| comment_number| comment

|  | A |
|---|---|
| 1 | image_name\| comment_number\| comment |
| 2 | 1000092795.jpg\| 0\| Two young guys with shaggy hair look at their hands while hanging out in the yard . |
| 3 | 1000092795.jpg\| 1\| Two young |
| 4 | 1000092795.jpg\| 2\| Two men in green shirts are standing in a yard . |
| 5 | 1000092795.jpg\| 3\| A man in a blue shirt standing in a garden . |
| 6 | 1000092795.jpg\| 4\| Two friends enjoy time spent together . |
| 7 | 10002456.jpg\| 0\| Several men in hard hats are operating a giant pulley system . |
| 8 | 10002456.jpg\| 1\| Workers look down from up above on a piece of equipment . |
| 9 | 10002456.jpg\| 2\| Two men working on a machine wearing hard hats . |
| 10 | 10002456.jpg\| 3\| Four men on top of a tall structure . |
| 11 | 10002456.jpg\| 4\| Three men on a large rig . |
| 12 | 1000268201.jpg\| 0\| A child in a pink dress is climbing up a set of stairs in an entry way . |
| 13 | 1000268201.jpg\| 1\| A little girl in a pink dress going into a wooden cabin . |
| 14 | 1000268201.jpg\| 2\| A little girl climbing the stairs to her playhouse . |
| 15 | 1000268201.jpg\| 3\| A little girl climbing into a wooden playhouse |
| 16 | 1000268201.jpg\| 4\| A girl going into a wooden building . |
| 17 | 1000344755.jpg\| 0\| Someone in a blue shirt and hat is standing on stair and leaning against a window . |

As we can see in the captions file above, each cell is divided into three parts namely image name, comment number, and comment. So each image is given a unique name. And each will have multiple captions numbered using field comment number. So we have a dataset large enough to train on our model.

Code for this project is referenced from git and an online article on how to apply transfer learning for image captioning[11].

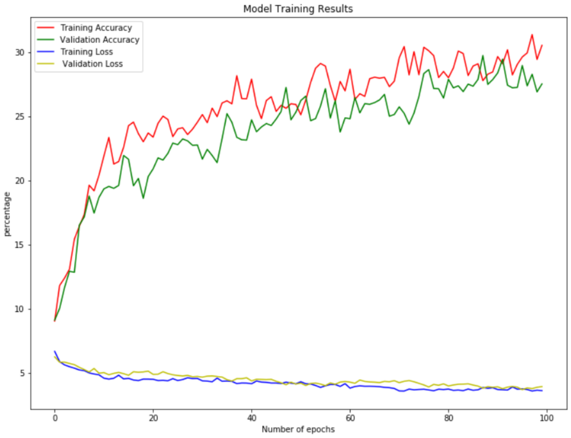
## V. RESULTS AND DISCUSSION

Model is trained on 6000 train images, 1000 validation images along with their captions and tested on 1000 images. In this section, I will discuss training and testing results separately. Training results are based on loss and accuracy. For the testing, I used Confusion matrix, BLEU score, and CHRF score to determine test accuracy.

### A. *Training Results*

After training the InceptionV3 model for 100 epochs with the train and validation datasets, we get four results in each epoch namely *loss, acc, val-loss,val-acc*. *loss, acc* represent the loss and accuracy values of the model for each epoch. Whereas *val-loss, val-acc* represent the loss and accuracy values of the validation set in each epoch of training. For optimal model training, loss should be decreasing and accuracy should be increasing.

To better understand the results, I will plot all the results on a single graph. It will be easy to determine whether the model is *under-fitting*, *over-fitting* or *perfect fit*.



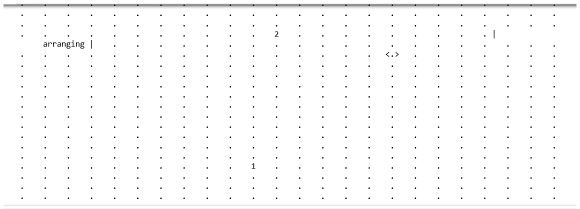
As we can see, both the training loss and validation loss are decreasing and validation loss is little high than training loss in most of the cases. Training accuracy and validation accuracy are increasing and validation accuracy is always less than training accuracy.

Based on the plotted results, the chosen dataset perfectly fits the model.

### B. *Testing Results*

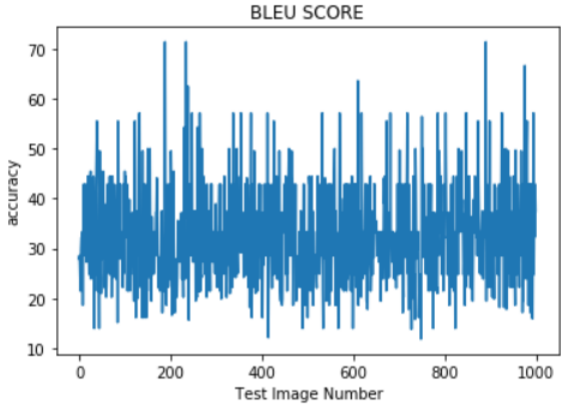
Testing is performed using 1000 images to generate captions for each of them using the trained model. For determining the accuracy, I compared actual image captions with the predicted captions. Three measures Confusion Matrix, BLEU Score and CHRF score are used to do this. Since this project is unsupervised learning and involves natural language processing, BLEU Score and CHRF Score are apt than ROC to determine the accuracy.

*1) Confusion Matrix:* Confusion matrix in the natural language tool kit is a plot between two lists of words. For each of the test image, we have an actual caption and predicted caption. By converting two of them into lists, I plotted confusion matrix for all the captions. It shows true positives and true negatives i.e., how many words in the caption of each image are correctly predicted by the model. Below image shows the confusion matrix between the actual captions of images and predicted captions.

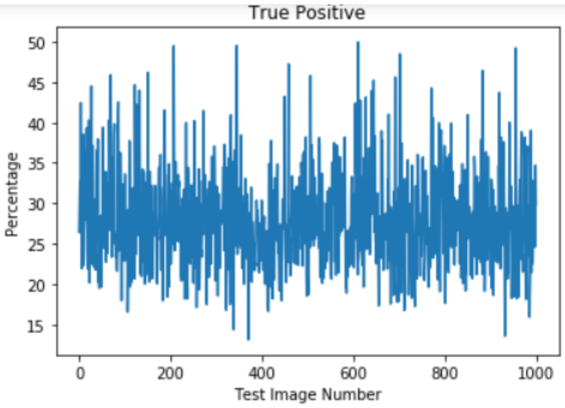


As we can see that the number of true positives i.e., correctly generated captions are less due to less training accuracy so far. True positives will increase with training the model and its accuracy.

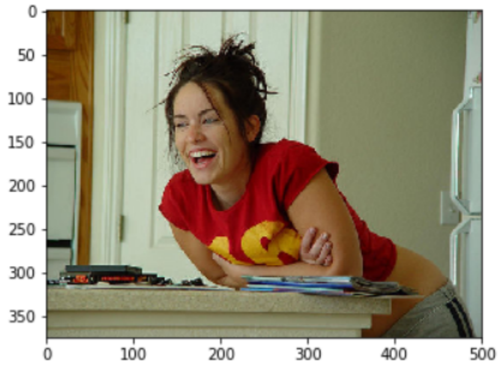
*2) BLEU SCORE:* Bilingual Evaluation Understudy is ascorefor comparing the predicted translation of the text to one or more reference translations. For each of 1000 image captions predicted by the model, bleu score compares it the actual image caption and gives the matching percentage between them. It is a part of the natural language tool kit and is useful in problems involving natural language processing. Below image shows the blue score for the 1000 test image captions predicted by the model.



*3) CHRF SCORE:* CHRF score determines the True positive percentage between the predicted image description and its original descriptions. It is similar to the result of confusion matrix. It will compare each of the predicted captions of an image with its actual caption and determines the true positive percentage based on the number of correctly predicted words in the caption. Below image shows the chrf score for the 1000 test image captions predicted by the model.



*4) Test Image:* I have randomly selected and images from the test images and generated caption for it using the trained model described till now. Below image shows the test image and its caption predicted by the model. The words *startseq* and *endseq* are the start and end points of the caption.



```
startseq girl in into wooden wooden endseq
```

## VI. CONCLUSION

Therefore, a image captioning model is successfully built using *InceptionV3* and *LSTM* neural networks on *flickr8kimages* and test results have been plotted. As mentioned earlier, this project needs a lot of training to get as more accurate as possible. The model has to go through all kinds of images in multiple iterations to analyze the objects in images and map their actions. As future work, this can be extended to videos where computers can caption continuous frames in videos which can be useful in many more real-time applications.

## REFERENCES

[1] Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, Tao Mei, ”Boosting Image Captioning With Attributes” in *The IEEE International Conference on Computer Vision (ICCV)*,2017, pp. 4894-4902.

[2] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, Jiebo Luo, ”Image Captioning With Semantic Attention” in *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*,2016, pp. 4651-4659.

[3] Hoo-Chang Shin, Holger R. Roth, Mingchen Gao, ”Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning” in *IEEE Transactions on Medical Imaging*,2016 , Vol 5 . 1285 - 1298.

[4] Chen, Long, et al. ”Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[5] Yu, Wei, et al. ”Visualizing and comparing AlexNet and VGG using deconvolutional layers.” Proceedings of the 33 rd International Conference on Machine Learning. 2016.

[6] Simonyan, Karen, and Andrew Zisserman. ”Very deep convolutional networks for large-scale image recognition.” arXiv preprint arXiv:1409.1556 (2014).

[7] Szegedy, Christian, et al. ”Rethinking the inception architecture for computer vision.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[8] Zhou, Chunting, et al. ”A C-LSTM neural network for text classification.” arXiv preprint arXiv:1511.08630 (2015).

[9] Karpathy, Andrej, and Li Fei-Fei. ”Deep visual-semantic alignments for generating image descriptions.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[10] https://www.kaggle.com/hsankesara/flickr-image-dataset/version/1

[11] https://towardsdatascience.com/image-captioning-with-keras-teaching-computers-to-describe-pictures-c88a46a311b8